

Graph Engine Service

API Reference

Issue	01
Date	2025-08-22



Copyright © Huawei Technologies Co., Ltd. 2025. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <https://www.huawei.com>

Email: support@huawei.com

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

- 1 Before You Start..... 1**
- 2 Constraints and Limitations on Using GES..... 4**
 - 2.1 Using Service Plane APIs..... 4
 - 2.2 Naming OBS Objects..... 5
- 3 API Overview..... 6**
 - 3.1 Management Plane APIs..... 6
 - 3.2 Service Plane API..... 10
 - 3.2.1 Memory Edition..... 10
 - 3.2.2 Database Edition..... 19
- 4 API Type or Version..... 24**
- 5 Calling APIs..... 25**
 - 5.1 Making an API Request..... 25
 - 5.1.1 Making a Management Plane API Request..... 25
 - 5.1.2 Making a Service Plane API Request..... 28
 - 5.2 Authentication..... 30
 - 5.2.1 Authentication of Management Plane APIs..... 30
 - 5.2.2 Authentication of Service Plane APIs..... 31
 - 5.3 Response..... 32
- 6 Management Plane APIs (V2)..... 34**
 - 6.1 System Management..... 34
 - 6.1.1 Querying Quotas (1.0.0)..... 34
 - 6.2 Graph Management..... 37
 - 6.2.1 Listing Graphs (2.1.18)..... 37
 - 6.2.2 Querying Graph Details (1.0.0)..... 46
 - 6.2.3 Creating a Graph (2.2.2)..... 55
 - 6.2.4 Stopping a Graph (1.0.0)..... 62
 - 6.2.5 Starting a Graph (1.0.0)..... 65
 - 6.2.6 Deleting a Graph (1.0.0)..... 67
 - 6.2.7 Incrementally Importing Data to a Graph (2.1.14)..... 70
 - 6.2.8 Exporting a Graph (1.0.5)..... 75
 - 6.2.9 Clearing a Graph (2.1.2)..... 78

6.2.10 Upgrading a Graph (1.0.5).....	81
6.2.11 Binding an EIP (1.0.6).....	84
6.2.12 Unbinding an EIP (1.0.6).....	86
6.2.13 Resizing a Graph (2.2.21).....	89
6.2.14 Forcibly Restarting a Graph (2.2.21).....	92
6.2.15 Expanding a Graph (2.2.23).....	94
6.2.16 Changing the Security Group (2.4.3).....	97
6.3 Backup Management.....	100
6.3.1 Viewing the List of All Backups (1.0.0).....	100
6.3.2 Viewing the Backup List of a Graph (1.0.0).....	104
6.3.3 Adding a Backup (1.0.0).....	108
6.3.4 Deleting a Backup (1.0.0).....	110
6.3.5 Exporting a Backup (2.3.16).....	113
6.3.6 Importing a Backup (2.3.16).....	115
6.3.7 Obtaining a Download Link for Backup Files (2.4.3).....	118
6.4 Metadata Management.....	120
6.4.1 Constraints.....	120
6.4.2 Querying the Metadata List (1.0.2).....	123
6.4.3 Querying Metadata (1.0.2).....	126
6.4.4 Adding Metadata (2.1.18).....	130
6.4.5 Deleting Metadata (1.0.2).....	134
6.4.6 Importing Metadata from OBS (1.0.0).....	136
6.5 Task Center.....	138
6.5.1 Querying the Job Status on the Management Plane (1.0.0).....	138
6.5.2 Querying Job Details in the Job Center (1.1.8).....	144
6.6 Plugin Management.....	150
6.6.1 Querying Scene Analysis Plugin Information.....	150
6.6.2 Subscribing to a Scene Analysis Plugin.....	154
6.6.3 Unsubscribing from a Scene Analysis Plugin.....	157
7 Service Plane APIs.....	161
7.1 Memory Edition.....	161
7.1.1 Vertex Operation APIs.....	161
7.1.1.1 Querying Vertices That Meet Filter Criteria (1.0.0).....	161
7.1.1.2 Querying Vertex Details (1.0.0).....	165
7.1.1.3 Adding a Vertex (1.0.6).....	168
7.1.1.4 Deleting a Vertex (1.0.6).....	170
7.1.1.5 Updating Vertex Properties (1.1.6).....	172
7.1.1.6 Batch Querying Vertex Data (1.1.9).....	175
7.1.1.7 Batch Adding Vertices (2.1.16).....	178
7.1.1.8 Batch Deleting Vertices (2.1.9).....	181
7.1.1.9 Batch Updating Vertex Properties (2.1.10).....	183
7.1.1.10 Adding a Vertex Label (1.1.6).....	187

7.1.1.11 Deleting a Vertex Label (1.1.6).....	189
7.1.1.12 Exporting Filtered Vertices (2.2.7).....	191
7.1.1.13 Deleting Filtered Vertices (2.2.7).....	194
7.1.2 Edge Operation APIs.....	198
7.1.2.1 Querying Edges That Meet Filter Criteria (1.0.0).....	198
7.1.2.2 Querying Edge Details (1.0.0).....	202
7.1.2.3 Adding an Edge (2.1.2).....	205
7.1.2.4 Deleting an Edge (1.0.6).....	209
7.1.2.5 Updating Edge Properties (1.1.6).....	212
7.1.2.6 Batch Querying Edge Data (1.1.6).....	215
7.1.2.7 Batch Adding Edges (2.1.16).....	218
7.1.2.8 Batch Deleting Edges (2.1.9).....	223
7.1.2.9 Batch Updating Edge Properties (2.1.10).....	227
7.1.2.10 Exporting Filtered Edges (2.2.7).....	230
7.1.2.11 Deleting Filtered Edges (2.2.7).....	233
7.1.3 Metadata Operation APIs.....	236
7.1.3.1 Adding a Label (1.1.6).....	236
7.1.3.2 Updating a Label (1.1.7).....	240
7.1.3.3 Querying Graph Metadata Details (1.0.0).....	243
7.1.3.4 Deleting a Label (2.2.18).....	247
7.1.3.5 Batch Adding Labels (2.2.21).....	249
7.1.3.6 Querying a Schema.....	253
7.1.3.7 Generating a Schema.....	256
7.1.3.8 Generating Data Assets.....	258
7.1.3.9 Obtaining Data Assets.....	260
7.1.4 Index Operation APIs.....	262
7.1.4.1 Creating an Index (1.1.6).....	262
7.1.4.2 Deleting an Index (1.1.6).....	265
7.1.4.3 Querying Indexes (1.1.6).....	267
7.1.5 Gremlin Operation APIs.....	269
7.1.5.1 Executing Gremlin Queries (1.0.0).....	269
7.1.6 Algorithm APIs.....	272
7.1.6.1 Running Algorithms (1.0.0).....	272
7.1.6.2 Algorithm API Parameter References.....	274
7.1.6.2.1 Common Algorithm Parameters.....	274
7.1.6.2.2 PageRank (1.0.0).....	279
7.1.6.2.3 PersonalRank (1.0.0).....	280
7.1.6.2.4 K-core (1.0.0).....	282
7.1.6.2.5 K-hop (1.0.0).....	282
7.1.6.2.6 Common Neighbors (1.0.0).....	283
7.1.6.2.7 Common Neighbors of Vertex Sets (2.2.13).....	284
7.1.6.2.8 Link Prediction (1.0.0).....	286

7.1.6.2.9 Shortest Path (2.1.5).....	286
7.1.6.2.10 All Shortest Paths (1.0.12).....	288
7.1.6.2.11 Filtered Shortest Path (2.2.4).....	289
7.1.6.2.12 SSSP (1.0.0).....	292
7.1.6.2.13 Shortest Path of Vertex Sets (2.1.5).....	293
7.1.6.2.14 n-Paths (1.1.2).....	294
7.1.6.2.15 Filtered n-Paths (2.2.22).....	295
7.1.6.2.16 Filtered All Pairs Shortest Paths (2.2.17).....	298
7.1.6.2.17 All Shortest Paths of Vertex Sets (2.2.15).....	300
7.1.6.2.18 Filtered All Shortest Paths (2.2.17).....	301
7.1.6.2.19 Connected Component (1.0.0).....	302
7.1.6.2.20 Label Propagation (2.1.8).....	303
7.1.6.2.21 Louvain (2.2.1).....	304
7.1.6.2.22 Node2vec (1.0.5).....	306
7.1.6.2.23 Real-time Recommendation (2.2.21).....	307
7.1.6.2.24 Degree Correlation (1.0.0).....	309
7.1.6.2.25 Triangle Count (1.0.0).....	310
7.1.6.2.26 Cluster Coefficient (1.0.0).....	310
7.1.6.2.27 Closeness Centrality (1.0.0).....	311
7.1.6.2.28 Betweenness Centrality (2.2.4).....	311
7.1.6.2.29 Edge Betweenness Centrality (2.2.4).....	313
7.1.6.2.30 Origin-Destination Betweenness Centrality (2.2.4).....	314
7.1.6.2.31 Circle Detection with a Single Vertex (2.2.4).....	315
7.1.6.2.32 Filtered Circle Detection (2.2.15).....	317
7.1.6.2.33 Subgraph Matching (2.2.16).....	320
7.1.6.2.34 Topicrank (2.2.20).....	321
7.1.7 Temporal Graph APIs.....	322
7.1.7.1 Community Evolution (temporal_graph).....	323
7.1.7.2 Temporal BFS (temporal_bfs).....	326
7.1.7.3 Temporal Paths.....	330
7.1.8 Path APIs.....	334
7.1.8.1 Querying Path Details (1.1.6).....	334
7.1.9 Graph Statistics APIs.....	337
7.1.9.1 Querying General Information About a Graph (1.0.0).....	337
7.1.9.2 Querying the Graph Version (2.0.0).....	341
7.1.10 Graph Operation APIs.....	343
7.1.10.1 Importing a Graph (2.1.14).....	343
7.1.10.2 Exporting a Graph (1.0.5).....	348
7.1.10.3 Clearing a Graph (2.1.1).....	351
7.1.11 Subgraph Operation APIs.....	353
7.1.11.1 Querying a Subgraph (2.1.13).....	353
7.1.11.2 Executing an Algorithm on a Subgraph.....	356

7.1.12 Job Management APIs.....	359
7.1.12.1 Querying Job Status on the Service Plane (1.0.0).....	359
7.1.12.2 Canceling a Job (1.0.0).....	363
7.1.12.3 Exporting Job Execution Results to Files (2.2.1).....	365
7.1.12.4 Listing Jobs (2.2.13).....	370
7.1.13 Custom Operations APIs.....	374
7.1.13.1 Performing Custom Operations.....	374
7.1.14 Filtered Query (2.2.13).....	376
7.1.15 Filtered Query V2 (2.3.6).....	388
7.1.16 Cypher Queries (2.2.16).....	398
7.1.16.1 Executing Cypher Queries.....	398
7.1.16.2 Cypher Prerequisites.....	404
7.1.16.3 Basic Operations and Compatibility.....	405
7.1.16.4 Supported Expressions, Functions, and Procedures.....	410
7.1.17 Domain-Specific Language (DSL) Query APIs (2.3.14).....	420
7.1.17.1 Executing DSL Algorithms.....	420
7.1.17.2 DSL Syntax.....	422
7.1.18 Graph Query Language (GQL) APIs (2.4.15).....	435
7.1.18.1 Executing GQL Queries.....	436
7.1.18.2 GQL Syntax.....	441
7.1.18.3 GQL Compatibility.....	448
7.1.19 Updating Specified Properties of Vertices and Edges by Importing a File (2.2.13).....	453
7.1.20 Deleting Vertices and Edges by Files (2.2.15).....	458
7.1.21 O&M Monitoring APIs.....	461
7.1.21.1 Viewing Monitoring Metrics.....	461
7.1.21.2 Viewing Real-Time Requests.....	468
7.2 Database Edition.....	471
7.2.1 Specification Description.....	471
7.2.2 Vertex Operation APIs.....	472
7.2.2.1 Querying Vertex Details.....	472
7.2.2.2 Batch Querying Vertices.....	475
7.2.2.3 Batch Adding Vertices.....	478
7.2.2.4 Batch Deleting Vertices.....	481
7.2.2.5 Batch Updating Vertex Properties.....	483
7.2.3 Edge Operation APIs.....	486
7.2.3.1 Querying Edge Details.....	486
7.2.3.2 Batch Querying Edges.....	489
7.2.3.3 Batch Adding Edges.....	492
7.2.3.4 Batch Deleting Edges.....	496
7.2.3.5 Batch Updating Edge Properties.....	500
7.2.4 Metadata Operation APIs.....	503
7.2.4.1 Adding a Label.....	503

7.2.4.2 Updating a Label.....	507
7.2.4.3 Querying Labels.....	510
7.2.4.4 Querying Graph Metadata Details.....	513
7.2.4.5 Generating Data Assets.....	516
7.2.4.6 Obtaining Data Assets.....	518
7.2.5 Index Operation APIs.....	520
7.2.5.1 Creating an Index.....	520
7.2.5.2 Deleting an Index.....	524
7.2.5.3 Querying Indexes.....	526
7.2.5.4 Batch Creating Indexes.....	528
7.2.6 HyG Graph Management APIs.....	532
7.2.6.1 Creating a HyG Graph.....	533
7.2.6.2 Synchronizing HyG Graph Data.....	535
7.2.6.3 Querying General Information About a HyG Graph.....	538
7.2.6.4 Deleting a HyG Graph.....	541
7.2.6.5 Listing HyG Graphs.....	543
7.2.6.6 Importing an HyG Graph.....	546
7.2.7 HyG Algorithm APIs.....	550
7.2.7.1 Running Algorithms.....	550
7.2.7.2 Algorithm API Parameter Reference.....	552
7.2.7.2.1 Common Algorithm Parameters.....	552
7.2.7.2.2 PageRank.....	554
7.2.7.2.3 PersonalRank.....	556
7.2.7.2.4 K-core.....	559
7.2.7.2.5 K-Hop.....	562
7.2.7.2.6 Common Neighbors of Vertex Sets.....	565
7.2.7.2.7 Shortest Path.....	568
7.2.7.2.8 All Shortest Paths.....	571
7.2.7.2.9 All Pairs Shortest Paths.....	574
7.2.7.2.10 SSSP.....	577
7.2.7.2.11 Shortest Path of Vertex Sets.....	580
7.2.7.2.12 All Shortest Paths of Vertex Sets.....	583
7.2.7.2.13 Connected Component.....	586
7.2.7.2.14 Triangle Count.....	588
7.2.7.2.15 Closeness Centrality.....	591
7.2.7.2.16 Betweenness Centrality.....	594
7.2.7.2.17 Edge Betweenness Centrality.....	597
7.2.7.2.18 Origin-Destination Betweenness Centrality.....	600
7.2.7.2.19 Circle Detection with a Single Vertex.....	603
7.2.7.2.20 TopicRank.....	606
7.2.7.2.21 Louvain.....	612
7.2.7.2.22 Bigclam.....	615

7.2.7.2.23 Cesna.....	618
7.2.7.2.24 Infomap.....	621
7.2.7.2.25 Label Propagation.....	624
7.2.7.2.26 Subgraph Matching.....	627
7.2.7.2.27 Link Prediction.....	630
7.2.7.2.28 n Paths.....	633
7.2.7.2.29 Cluster Coefficient.....	636
7.2.7.2.30 Maximum Common Connected Induced Subgraph (MCCIS).....	639
7.2.7.3 Algorithm Results in TXT Format.....	642
7.2.7.4 Executing the DSL Algorithm.....	658
7.2.7.5 DSL Syntax.....	660
7.2.7.5.1 Graph Operation APIs.....	660
7.2.7.5.2 API for Running Custom Algorithms (Currently, the Pregel Programming Model Is Supported).....	661
7.2.7.5.3 Pregel Programming API.....	661
7.2.7.5.4 Programming Example of Creating Custom Graph Analysis Algorithms.....	662
7.2.8 HyG Job Management APIs.....	664
7.2.8.1 Dumping HyG Algorithm Results.....	664
7.2.8.2 Canceling a HyG Job.....	667
7.2.9 Native Algorithm APIs.....	669
7.2.9.1 Running Algorithms.....	669
7.2.9.2 Algorithm API Parameter References.....	671
7.2.9.2.1 Common Algorithm Parameters.....	671
7.2.9.2.2 Shortest Path.....	674
7.2.9.2.3 Shortest Path of Vertex Set.....	675
7.2.9.2.4 Common Neighbors of Vertex Sets.....	677
7.2.10 Graph Statistics APIs.....	679
7.2.10.1 Querying General Information About a Graph.....	679
7.2.10.2 Querying the Graph Version.....	682
7.2.11 Graph Operation APIs.....	684
7.2.11.1 Importing a Graph.....	684
7.2.11.2 Clearing a Graph.....	688
7.2.11.3 Exporting a Graph.....	690
7.2.11.4 Creating a Graph.....	694
7.2.11.5 Deleting a Graph.....	696
7.2.11.6 Graph List.....	698
7.2.12 Job Management APIs.....	700
7.2.12.1 Querying the Job List.....	700
7.2.12.2 Querying the Job Status.....	703
7.2.13 Cypher Operation APIs.....	706
7.2.13.1 Executing Cypher Queries.....	706
7.2.13.2 Basic Operations and Compatibility.....	712
7.2.13.3 Supported Expressions, Functions, and Procedures.....	717

7.2.14 GQL APIs.....	726
7.2.14.1 Executing GQL Queries.....	726
7.2.14.2 GQL Syntax.....	732
7.2.14.3 GQL Compatibility.....	739
7.2.15 Interactive Transaction APIs.....	744
7.2.15.1 Creating a Transaction.....	744
7.2.15.2 Executing Transaction Cypher Statements.....	746
7.2.15.3 Committing a Transaction.....	752
7.2.15.4 Rolling Back a Transaction.....	754
7.2.16 O&M Monitoring APIs.....	756
7.2.16.1 Viewing Monitoring Metrics.....	756
7.2.16.2 Viewing Real-Time Requests.....	763
8 Application Examples.....	766
8.1 Analyzing Graphs Using HyG Algorithms.....	766
9 Permissions Policies and Supported Actions.....	769
10 GES Metrics on Cloud Eye.....	776
11 Out-of-Date APIs.....	783
11.1 Management Plane APIs (V1).....	783
11.1.1 System Management APIs.....	783
11.1.1.1 Viewing Quotas (1.0.0).....	783
11.1.2 Graph Management APIs.....	786
11.1.2.1 Querying the Graph List (2.1.18).....	786
11.1.2.2 Querying Graph Details (1.0.0).....	793
11.1.2.3 Creating a Graph (2.2.2).....	800
11.1.2.4 Stopping a Graph (1.0.0).....	807
11.1.2.5 Starting a Graph (1.0.0).....	810
11.1.2.6 Deleting a Graph (1.0.0).....	812
11.1.2.7 Incrementally Importing Data to Graphs (2.1.14).....	814
11.1.2.8 Exporting a Graph (1.0.5).....	820
11.1.2.9 Clearing a Graph (2.1.2).....	823
11.1.2.10 Upgrading a Graph (1.0.5).....	825
11.1.2.11 Binding an EIP (1.0.6).....	828
11.1.2.12 Unbinding an EIP (1.0.6).....	831
11.1.2.13 Resizing a Graph (2.2.21).....	833
11.1.2.14 Forcibly Restarting a Graph (2.2.21).....	836
11.1.2.15 Expanding a Graph (2.2.23).....	838
11.1.3 Backup Management APIs.....	841
11.1.3.1 Viewing the List of All Backups (1.0.0).....	841
11.1.3.2 Viewing the Backup List of a Graph (1.0.0).....	845
11.1.3.3 Adding a Backup (1.0.0).....	849
11.1.3.4 Deleting a Backup (1.0.0).....	851

11.1.4 Metadata Management APIs.....853

11.1.4.1 Constraints.....853

11.1.4.2 Listing Metadata Files (1.0.2).....855

11.1.4.3 Querying Metadata (1.0.2).....858

11.1.4.4 Adding Metadata (2.1.18).....861

11.1.4.5 Deleting Metadata (1.0.2).....865

11.1.4.6 Importing Metadata from OBS (1.0.0).....867

11.1.5 Task Center APIs.....869

11.1.5.1 Querying Job Status on the Management Plane (1.0.0).....869

11.1.5.2 Querying Job Details in the Job Center (1.1.8).....874

11.2 Service Plane APIs.....880

11.2.1 Vertex Operation APIs.....880

11.2.1.1 Batch Adding Vertices (2.1.9).....880

11.2.2 Edge Operation APIs.....883

11.2.2.1 Adding an Edge (1.0.6).....883

11.2.2.2 Batch Adding Edges (2.1.9).....885

11.2.3 Algorithm APIs.....889

11.2.3.1 Shortest Paths (1.0.0).....889

11.2.3.2 Shortest Path of Vertex Sets (1.0.0).....890

11.2.3.3 Label Propagation (1.0.0).....891

11.2.3.4 Louvain (1.0.0).....891

12 Appendix.....893

12.1 Status Codes.....893

12.2 Error Codes.....897

12.2.1 Error Codes for Management Plane APIs.....897

12.2.2 Error Codes for Service Plane APIs.....905

12.3 Obtaining a Project ID.....914

12.4 Obtaining the Account Name and Account ID.....915

1 Before You Start

Welcome to *Graph Engine Service API Reference*. Graph Engine Service (GES) is the industry's first commercially available, independently developed distributed native graph engine. It provides services for querying and analyzing graph-structured data based on relationships. It is extensively applied in scenarios with abundant relational data, such as social applications, corporate relationship analysis, risk control, recommendations, public sentiment monitoring, and fraud prevention.

You can use the APIs provided in this document to perform various operations on GES resources.

- **Management plane APIs**
Management plane APIs offer functions for graph management, including creation, stopping, startup, restoration, upgrade, import, export, and data clearance, EIP binding and unbinding, backup creation, query, deletion, and metadata management. You need to call these APIs when performing such operations.
- **Service plane APIs**
Service plane APIs deliver service-related functions for graphs, including addition, deletion, query, and modification of vertices, edges, and metadata, execution of Gremlin queries, and algorithm runs. You need to call these APIs when performing such operations.

For a comprehensive list of all APIs supported by GES, refer to [Management Plane APIs](#) and [Memory Edition](#).

Prior to calling GES APIs, make sure you have a thorough understanding of GES concepts. Detailed information can be found in the [Service Overview](#).

Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoints of all services, see [Regions and Endpoints](#).

Concepts

- Account

An account is created when you successfully sign up for Huawei Cloud. An account has full access permissions for all the resources and cloud services. It can be used to reset user passwords and grant users permissions. The account is a payment entity. For security purposes, do not directly use the account to perform routine management. Instead, create IAM users and grant them permissions for routine management.

- User

A user is created in IAM to use cloud services. Each user has its own identity credentials (password and access keys).

The account name, username, and password will be required for API authentication.

- Region

A region is a geographic area where cloud services are deployed. Availability zones (AZs) in the same region can communicate with each other over an intranet, while AZs in different regions are isolated from each other. By creating cloud resources in various regions, you can design applications to better meet the needs of specific customers or to comply with legal or other regional requirements.

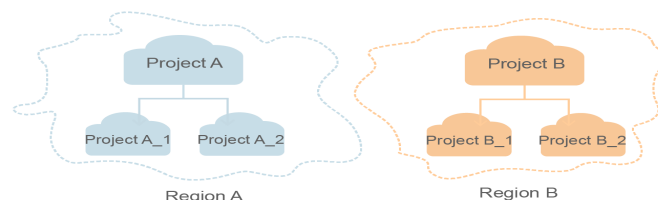
- AZ

An availability zone (AZ) consists of one or more physical data centers. Within an AZ, compute, networking, storage, and other resources are logically partitioned into multiple clusters. Multiple AZs within a region are connected via high-speed fiber optics to meet user needs for building cross-AZ high-availability systems.

- Project

Projects group and isolate resources (including compute, storage, and network resources) across physical regions. A default project is provided for each Huawei Cloud region, and subprojects can be created under each default project. Users can be granted permissions to access all resources in a specific project. If you desire more granular permission controls, you can create sub-projects under the default regional project, purchase resources within these sub-projects, and authorize them accordingly. This ensures that users can only access resources from particular sub-projects, thereby enhancing precision in resource authorization management.

Figure 1-1 Project isolating model



- Enterprise project

Enterprise projects group and manage resources across regions. Resources in enterprise projects are logically isolated from each other. An enterprise project

can include resources from multiple regions, and resources can be added to or removed from enterprise projects.

For details about enterprise projects and how to obtain enterprise project IDs, see [Enterprise Management User Guide](#).

2 Constraints and Limitations on Using GES

2.1 Using Service Plane APIs

You can access the service plane APIs by anyone of the following methods:

- Through the ECS. The VPC for creating the ECS must be the same as that selected during graph creation. If the same security group is selected, you can directly access the APIs. If the security groups are not the same, enable the access permission for the ECS in the security group where the graph is created. That is, enable ports **80** and **443** for inbound traffic and all ports for outbound traffic. The ingress and egress ports support HTTP and HTTPS access respectively. In this scenario, **SERVER_URL** of the APIs is the private access address in the graph details on the GES console or the value of the **private_ip** field in the response body of the management plane API for querying graph details.
- Through the ECS. The VPC for creating the ECS is not the same as that selected during graph creation. You need to create a VPC peering connection between the VPC to which the ECS belongs and the VPC in which the graph is created. For details, see [Creating a VPC Peering Connection](#). In addition, you need to enable the access permission for the ECS in the security group where the graph is created. That is, enable ports **80** and **443** for inbound traffic and all ports for outbound traffic. In this scenario, **SERVER_URL** of the APIs is the private access address in the graph details on the GES console or the value of the **private_ip** field in the response body of the management plane API for querying graph details.
- Through the public network. You need to create an EIP and enable the access permission for the client in the security group where the graph is created, that is, enable ports **80** and **443** for inbound traffic and all ports for outbound traffic. In this scenario, **SERVER_URL** of the APIs is the public access address in the graph details on the GES console or the value of the **publicIp** field in the response body of the management plane API for querying graph details (also the EIP you bind or create).

2.2 Naming OBS Objects

The OBS object names supported by GES can contain the following characters:

Letters and digits	0-9, a-z, A-Z
Special characters	! - _ . * ' ()

The following characters are not supported:

Special characters	\{^}%`]">[~<# &@:,\$=+? and spaces
ASCII control characters	Range: • 00–1F in hexadecimal form (0–31 in decimal form) • 7F (127 in decimal form)

3 API Overview

3.1 Management Plane APIs

GES management plane APIs comprise APIs for system management, graph management, backup management, metadata management, and task center.

Table 3-1 System management APIs

Parameter	Version	URL	Function
Querying Quotas	1.0.0	GET /v2/{project_id}/graphs/quotas	Query the quota of graphs, edges, or backups. When creating or backing up a graph, you can call this API to view the quota.

Table 3-2 Graph management APIs

Parameter	Version	URL	Function
Listing Graphs	2.1.18	GET /v2/{project_id}/graphs	List all created graphs.
Querying Graph Details	1.0.0	GET /v2/{project_id}/graphs/{graph_id}	Query details about a graph, including the graph's private and public access addresses, version number, and imported vertex and edge data sets.
Creating a Graph	2.2.2	POST /v2/{project_id}/graphs	Create a graph after you define the metadata and vertex and edge data sets of the graph.

Parameter	Version	URL	Function
Stopping a Graph	1.0.0	POST /v2/{project_id}/graphs/{graph_id}/stop	You can stop a graph at any time because service continuity is not required. The graph will not be charged after it is stopped.
Starting a Graph	1.0.0	POST /v2/{project_id}/graphs/{graph_id}/start	If you want to use a stopped graph, you can restore the data to the state it was before the shutdown or to a backup time point.
Deleting a Graph	1.0.0	DELETE /v2/{project_id}/graphs/{graph_id}	Delete a graph when you do not need it. The graph will not be charged after it is deleted.
Incrementally Importing Data into a Graph	2.1.14	POST /v2/{project_id}/graphs/{graph_id}/import-graph	You need to incrementally import graph data in batches.
Exporting a Graph	1.0.5	POST /v2/{project_id}/graphs/{graph_id}/export-graph	You need to export all the data in a graph as a text file.
Clearing a Graph	2.1.2	POST /v2/{project_id}/graphs/{graph_id}/clear-graph	You need to clear all the data in a graph, including the vertex and edge data.
Upgrading a Graph	1.0.5	POST /v2/{project_id}/graphs/{graph_id}/upgrade	If bugs are detected in a graph of the early version or new functions need to be added, you need to upgrade the graph to the new version.
Binding an EIP	1.0.6	POST /v2/{project_id}/graphs/{graph_id}/bind-eip	To access a graph over the public network, you need to bind an EIP.
Unbinding an EIP	1.0.6	POST /v2/{project_id}/graphs/{graph_id}/unbind-eip	You can unbind an EIP from a graph if you no longer need to access the graph over the public network.
Resizing a Graph	2.2.21	POST /v2/{project_id}/graphs/{graph_id}/resize	Resize a graph.

Parameter	Version	URL	Function
Forcibly Restarting a Graph	2.2.21	POST /v2/{project_id}/graphs/{graph_id}/restart	Forcibly start a graph.
Expanding a Graph	2.2.23	POST /v2/{project_id}/graphs/{graph_id}/expand	Dynamically resize multiple secondary nodes to enable concurrent handling of increased read requests.

Table 3-3 Backup management APIs

Parameter	Version	URL	Function
Viewing the List of All Backups	1.0.0	GET /v2/{project_id}/graphs/backups	View all backup details of all graphs.
Viewing the Backup List of a Graph	1.0.0	GET /v2/{project_id}/graphs/{graph_id}/backups	View details about all backups of a graph, including the backup start time and end time.
Creating a Backup	1.0.0	POST /v2/{project_id}/graphs/{graph_id}/backups	Backup is used to improve data reliability. It can also be used as a snapshot of a graph for you to restore data when necessary.
Deleting a Backup	1.0.0	DELETE /v2/{project_id}/graphs/{graph_id}/backups/{backup_id}	Delete backups of a graph.
Exporting a Backup	2.3.16	POST /v2/{project_id}/graphs/{graph_id}/backups/export	Export a GES graph instance backup to OBS.
Importing a Backup	2.3.16	POST /v2/{project_id}/graphs/{graph_id}/backups/import	Import a GES graph instance backup from OBS.

Table 3-4 Metadata management APIs

Parameter	Version	URL	Function
Listing Metadata	1.0.2	GET /v2/{project_id}/graphs/metadatas	Query details about all metadata files, including the status and OBS storage path.
Querying Metadata	1.0.2	GET /v2/{project_id}/graphs/metadatas/{metadata_id}	Query details about a graph metadata file.
Creating a Metadata File	2.1.18	POST /v2/{project_id}/graphs/metadatas	Adding metadata is a preparation operation before creating a graph. You must create metadata before creating the graph.
Deleting a Metadata File	1.0.2	DELETE /v2/{project_id}/graphs/metadatas/{metadata_id}	Delete a metadata file.
Importing Metadata from OBS	1.0.0	POST /v2/{project_id}/graphs/metadata/upload-from-obs	Import metadata from OBS.

Table 3-5 Task center APIs

Parameter	Version	URL	Function
Querying Job Status	1.0.0	GET /v2/{project_id}/graphs/{graph_id}/jobs/{job_id}/status	Graph deleting, stopping, starting, restoring, incrementally importing, clearing, and upgrading are asynchronous jobs initialized by calling these APIs. These APIs return the job IDs. You can view the execution status of each asynchronous job through the corresponding API.

Parameter	Version	URL	Function
Querying Job Details in Task Center	1.1.8	GET /v2/{project_id}/graphs/jobs	View all asynchronous jobs.

Table 3-6 Plugin management APIs

API	Version	URL	Function
Querying Scene Analysis Plugin Information	2.3.11	GET /v2/{project_id}/graphs/scenes	Query information about the application analysis capability in a scene, including information about the applications, parameters, and function details.
Subscribing to a Scene Analysis Plugin	2.3.11	POST /v2/{project_id}/graphs/{graph_id}/scenes/register	Subscribe to a scene analysis plugin so that you can use the function through the service plane APIs.
Unsubscribing from a Scene Analysis Plugin	2.3.11	POST /v2/{project_id}/graphs/{graph_id}/scenes/unregister	Unsubscribe from a scene analysis plugin. After the subscription is canceled, you cannot use the function through application service plane APIs.

3.2 Service Plane API

3.2.1 Memory Edition

Memory edition service plane APIs cover vertex operations, edge operations, metadata operations, index operations, Gremlin operations, algorithms, paths, graph statistics, graph and subgraph operations, job management, and Cypher operations.

Table 3-7 Vertex operation APIs

API	Version	URL	Function
Querying Vertices That Meet Filter Criteria	1.0.0	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=query	Query vertices based on a filtering criterion. For example, if the vertex metadata contains the age property, the filter criterion can be 'age > 18'.
Querying Vertex Details	1.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/detail?vertexIds={vertex_ids}	Query details about a specified vertex of a specified set of vertices, including the label information.
Adding a Vertex	1.0.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices	Add a vertex.
Deleting a Vertex	1.0.6	DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}	Delete a vertex.
Updating Vertex Properties	1.1.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/properties/action?action_id={actionId}	Modify a vertex's properties, including adding, changing, and deleting properties.
Batch Querying Vertex Data	1.1.9	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-query	Query vertex details in batches.
Batch Adding Vertices	2.1.16	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-add	Add vertices in batches.
Batch Deleting Vertices	2.1.9	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-delete	Delete vertices in batches based on the vertex IDs.
Batch Updating Vertex Properties	2.1.10	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/properties/action?action_id={actionId}	Update vertex properties in batches.
Adding a Vertex Label	1.1.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/labels	Add a vertex label.

API	Version	URL	Function
Deleting a Vertex Label	1.1.6	DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/labels/{label_name}	Delete a vertex label.
Exporting Filtered Vertices	2.2.7	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=export	Export vertex sets that meet the filter criteria.
Deleting Filtered Vertices	2.2.7	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=delete	Delete vertex sets that meet the filter criteria.

Table 3-8 Edge operation APIs

API	Version	URL	Function
Querying Vertices That Meet Filter Criteria	1.0.0	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=query	Filter the edges that meet the filter criteria of edge properties.
Querying Edge Details	1.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?source={sourceVertex}&target={targetVertex}&index={index}	Query detailed information about an edge based on its source vertex, target vertex, and index, and return the labels and properties on the edge.
Adding an Edge	1.0.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges	Add an edge.
Deleting an Edge	1.0.6	DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/edges?source={sourceVertex}&target={targetVertex}&index={index}	Delete an edge.

API	Version	URL	Function
Updating Edge Properties	1.1.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?action_id={actionId}&source={sourceVertex}&target={targetVertex}&index={index}	Modify an edge's properties, including adding, changing, and deleting properties.
Batch Querying Edge Data	1.1.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-query	Query edge details in batches.
Batch Adding Edges	2.1.16	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add	Add edges in batches.
Batch Deleting Edges	2.1.9	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-delete	Delete edges in batches based on the source vertices, target vertices, and indexes of the edges.
Batch Updating Edge Properties	2.1.10	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?action_id={actionId}	Update edge properties in batches.
Exporting Filtered Edges	2.2.7	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=export	Export the edge set that meets the filter criteria.
Deleting Filtered Edges	2.2.7	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=delete	Delete the edge set that meets the filter criteria.

Table 3-9 Metadata operation APIs

API	Version	URL	Function
Adding a Label	1.1.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels	Add a label.

API	Version	URL	Function
Updating a Label	1.1.7	POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{label_name}/properties	Update a label.
Querying Graph Metadata Details	1.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema	Query metadata details.
Deleting a Label	2.2.18	DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{labelName}	Delete a label as well as the vertices and edges associated with the label.
Batch Adding Labels	2.2.21	POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/action?action_id=batch-add	Add labels in batches.

Table 3-10 Index operation APIs

API	Version	URL	Function
Creating an Index	1.1.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/indices	Create an index.
Deleting an Index	1.1.6	DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/indices/{indexName}	Delete an index.
Querying Indexes	1.1.6	GET /ges/v1.0/{project_id}/graphs/{graph_name}/indices	Query indexes.

Table 3-11 Gremlin operation APIs

API	Version	URL	Function
Executing Gremlin Queries	1.0.0	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-gremlin-query	Execute Gremlin queries.

Table 3-12 Algorithm APIs

Operation	Version	URL	Function
Running Algorithms	1.0.0	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm	Run algorithms.

Table 3-13 Temporal graph APIs

API	Version	URL	Function
Community Evolution (temporal graph)	2.3.9	POST /ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?action_id=execute-analysis	The community evolution algorithm generates a temporal graph that shows structure changes of a community over time.
Temporal BFS (temporal_bfs)	2.3.9	POST /ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?action_id=execute-analysis	This algorithm searches for associated vertices based on temporal message passing and temporal BFS algorithms, and outputs the visit time of each vertex and the distance from the vertex to the source start vertex.
Temporal Paths	2.3.9	POST /ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?action_id=execute-analysis	Execute the temporal paths algorithm based on input parameters. Only one temporal path that meets the conditions is returned between two vertices.

Table 3-14 Path APIs

Parameter	Version	URL	Function
Querying Path Details	1.1.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/paths/action?action_id=query-detail	Query the path details.

Table 3-15 Graph statistics APIs

Operation	Version	URL	Function
Querying General Information About a Graph	1.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/summary	Query general information about a graph.
Querying the Graph Version	2.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/version	Query the graph version.

Table 3-16 Graph operation APIs

Operation	Version	URL	Function
Importing a Graph	2.1.14	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-graph	Import data.
Exporting a Graph	1.0.5	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph	Export data.
Clearing a Graph	2.1.2	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=clear-graph	Clear data.

Table 3-17 Subgraph operation APIs

Operation	Version	URL	Function
Querying a Subgraph	2.1.13	POST /ges/v1.0/{project_id}/graphs/{graph_name}/subgraphs/action?action_id=query	Queries the subgraphs formed by the entered vertices and edges between the vertices.
Executing an Algorithm on a Subgraph	2.1.13	POST /ges/v1.0/{project_id}/graphs/{graph_name}/subgraphs/action?action_id=execute-algorithm	Adjust the subgraph creation type based on the input and execute an algorithm on the generated subgraph.

Table 3-18 Job management APIs

Operation	Version	URL	Function
Querying Job Status	1.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?offset= <i>offset</i> &limit= <i>limit</i>	Query job status.
Canceling a Job	1.0.0	DELETE https://{Endpoint}/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}	Cancel a job.
Exporting Job Execution Results to Files	2.2.1	POST /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/action?action_id=export-result	Export the execution result (result) of an asynchronous job (job_id) to a file.

Operation	Version	URL	Function
Listing Jobs	2.2.13	GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status?limit={limit}&offset={offset}	After the ID of an asynchronous job is returned, if the job ID at the service layer is lost and cannot be obtained through the API, a new API is provided to query all asynchronous jobs stored in the engine. The job ID, job status, and original request of each job are returned.

Table 3-19 Custom operations APIs

API	Version	URL	Function
Performing Custom Operations	2.3.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-operation	Query the job status.

Table 3-20 Other service plane APIs

Operation	Version	URL	Function
Filtered-query API	2.2.15	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=filtered-query	Filter the k-hop process layer by layer, and list the k-hop vertices or edges that meet the filter criteria.
Updating Specified Properties of Vertices and Edges by Importing a File	2.2.13	POST /v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-properties	Update specified properties of vertices and edges by importing a file.

Operation	Version	URL	Function
Deleting Vertices and Edges by Reading the Files	2.2.15	POST /v1.0/{project_id}/graphs/{graph_name}/action?action_id=delete-by-file	Delete vertices and edges by reading the files.
Cypher operation APIs	2.2.16	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query	Run Cypher statements to query data in GES and obtain results.

3.2.2 Database Edition

Database edition service plane APIs include vertex operations, edge operations, metadata operations, index operations, HyG dataset management, HyG algorithm, algorithms, graph statistics, graph operations, job management, and Cypher operations.

Table 3-21 Vertex operation APIs

API	Start Version	URL	Function
Querying Vertex Details	1.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/detail?vertexIds={vertex_ids}	Query details about a specified vertex of a specified set of vertices, including the label information.
Batch Querying Vertex Data	1.1.9	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-query	Query vertex details in batches.
Batch Adding Vertices	2.1.16	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-add	Add vertices in batches.
Batch Deleting Vertices	2.1.9	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-delete	Delete vertices in batches based on the vertex IDs.
Batch Updating Vertex Properties	2.1.10	POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/properties/action?action_id={actionId}	Update vertex properties in batches.

Table 3-22 Edge operation APIs

API	Start Version	URL	Function
Querying Edge Details	1.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail? source={sourceVertex}&target={targetVertex}&label={label}&sortKey={sortKey}&sortKeyType={sortKeyType}	Query detailed information about an edge based on its source vertex, target vertex, and index, and return the labels and properties on the edge.
Batch Querying Edge Data	1.1.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-query	Query edge details in batches.
Batch Adding Edges	2.1.16	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add	Add edges in batches.
Batch Deleting Edges	2.1.9	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-delete	Delete edges in batches based on the source vertices, target vertices, and indexes of the edges.
Batch Updating Edge Properties	2.1.10	POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?action_id={actionId}	Update edge properties in batches.

Table 3-23 Metadata operation APIs

API	Start Version	URL	Function
Adding a Label	1.1.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels	Add a label.
Updating a Label	1.1.7	POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{label_name}/properties	Update a label.
Querying Graph Metadata Details	1.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema	Query metadata details.

API	Start Version	URL	Function
Querying Labels	2.2.18	GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema?label={labelName}	Query labels.

Table 3-24 Index operation APIs

API	Start Version	URL	Function
Creating an Index	1.1.6	POST /ges/v1.0/{project_id}/graphs/{graph_name}/indices	Create an index.
Deleting an Index	1.1.6	DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/indices/{indexName}	Delete an index.
Querying Indexes	1.1.6	GET /ges/v1.0/{project_id}/graphs/{graph_name}/indices	Query indexes.

Table 3-25 Native algorithm API

API	Start Version	URL	Function
Running Algorithms	1.0.0	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm	Run native algorithms.

Table 3-26 Graph statistics APIs

API	Start Version	URL	Function
Querying General Information About a Graph	1.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/summary?label_details={labelDetails}	Query general information about a graph.

API	Start Version	URL	Function
Querying the Graph Version	2.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/version	Query the graph version.

Table 3-27 Graph operation APIs

API	Start Version	URL	Function
Importing a Graph	2.1.14	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-graph	Import graph data.
Clearing a Graph	2.1.2	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=clear-graph	Clear graph data.

Table 3-28 Job management APIs

API	Start Version	URL	Function
Querying Job Status	1.0.0	GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?offset= <i>offset</i> &limit= <i>limit</i>	Query the job status.
Querying the Job List	2.2.13	GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status?limit={limit}&offset={offset}	Query all asynchronous jobs saved in the engine and return the ID, status, and original request of each job.

Table 3-29 Cypher operation APIs

API	Start Version	URL	Function
Cypher Operation APIs	2.2.16	POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query	Run Cypher statements to query data in GES and obtain results.

Table 3-30 Interactive transaction APIs

API	URL	Function
Creating a Transaction	POST /ges/v1.0/{project_id}/graphs/{graph_name}/transaction	Create a transaction.
Executing Transaction Cypher Statements	POST /ges/v1.0/{project_id}/graphs/{graph_name}/transaction/{commit}	Execute transaction Cypher statements.
Committing a Transaction	POST /ges/v1.0/{project_id}/graphs/{graph_name}/transaction/{commit}/commit	Commit a transaction.
Rolling Back a Transaction	DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/transaction/{commit}	Roll back a transaction.

Table 3-31 O&M monitoring APIs

API	URL	Function
Viewing Monitoring Metrics	GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?offset= <i>offset</i> &limit= <i>limit</i>	Query the job status.
Viewing Real-Time Requests	GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status?limit={limit}&offset={offset}	Query all asynchronous jobs saved in the engine and return the ID, status, and original request of each job.

4 API Type or Version

The GES API version corresponds to the software version. 1.0.0 is the start version number. Other versions are updated based on the start version and are backward compatible. For details, see the versions of APIs in [Management Plane APIs](#) and [Memory Edition](#).

You are advised to select a proper type as required.

5 Calling APIs

5.1 Making an API Request

5.1.1 Making a Management Plane API Request

This section describes the structure of a REST API request on the management plane of GES, and uses the IAM API for [obtaining a user token](#) as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

- **URI-scheme:** Protocol used to transmit requests. All APIs use **HTTPS**.
- **Endpoint:** Domain name or IP address of the server bearing the REST service endpoint. Obtain this value from [Regions and Endpoints](#).
- **resource-path:** Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API for obtaining a user token is **/v3/auth/tokens**.
- **query-string:** Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "Parameter name=Parameter value". For example, **? limit=10** indicates that a maximum of 10 data records will be displayed.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server:

- **GET:** requests the server to return specified resources.

- **PUT**: requests the server to update specified resources.
- **POST**: requests the server to add resources or perform special operations.
- **DELETE**: requests the server to delete specified resources, for example, an object.
- **HEAD**: same as GET except that the server must return only the response header.
- **PATCH**: requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows:

Table 5-1 Common request headers

Parameter	Mandatory	Description	Example
Content-Type	Yes	Specifies the request body type or format. This field is mandatory and its default value is application/json . Other values of this field will be provided for specific APIs if any.	application/json
X-Auth-Token	This field is mandatory only for authentication using tokens.	Specifies a user token only for token-based API authentication. The user token is a response to the API used to obtain a user token . This API is the only one that does not require authentication.	-
X-Project-ID	No	Specifies a subproject ID. This parameter is mandatory only in multi-project scenarios.	e9993fc787d94b6c886cbaa340f9c0f4
Authorization	This field is mandatory for authentication using AK/SK.	Specifies the signature authentication information. The value is obtained from the request signing result.	-

Parameter	Mandatory	Description	Example
X-Sdk-Date	This field is mandatory for authentication using AK/SK.	Specifies the time when a request is sent. The time is in <i>YYYYMMDDTHHMMSSZ</i> format. The value is the current Greenwich Mean Time (GMT) time of the system.	20150907T101459Z
Host	This field is mandatory for authentication using AK/SK.	Specifies the information about the requested server. The value can be obtained from the URL of the service API. The value is in the <i>hostname[:port]</i> format. Default port used for https requests is port 443 .	code.test.com or code.test.com:443
Content-Length	This field is mandatory for POST and PUT requests, but must be left blank for GET requests.	Specifies the length of the request body. The unit is byte.	3495
X-Language	No	Request language	en-us

 NOTE

In addition to token-based authentication, authentication using access key ID/secret access key (AK/SK) is also supported. During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request.

For more information, see [AK/SK-based Authentication](#).

Request Body

The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

For the API of [obtaining a user token](#), obtain the request parameters and parameter descriptions in the API request. The following provides an example

request with a body included. Replace *username*, *domainname*, ******* (login password), and *xxxxxxxxxxxxxxxxxxxx* (project name) with the actual values. You can obtain the endpoint from [Regions and Endpoints](#).

NOTE

The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under an account. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see [Obtaining a User Token](#).

If all data required for the API request is available, you can send the request to call the API through [curl](#), [Postman](#), or code. In the response to the API used to [obtain a user token](#), **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

5.1.2 Making a Service Plane API Request

This section describes the structure of a REST API on the service plane of GES.

Request URI

A request URI of a service plane API of GES is in the following format:

{URI-scheme} :// {SERVER_URL} / {resource-path} ? {query-string}

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

- **URI-scheme**: Protocol used to transmit requests. All APIs use **HTTPS**.
- **SERVER_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- **resource-path**: Access path of an API for performing a specified operation. Obtain the value from the URI module of the API, for example, **ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=query**.
- **query-string**: Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "Parameter name=Parameter value". For example, **? limit=10** indicates that a maximum of 10 data records will be displayed.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server:

- **GET**: requests the server to return specified resources.
- **PUT**: requests the server to update specified resources.
- **POST**: requests the server to add resources or perform special operations.
- **DELETE**: requests the server to delete specified resources, for example, an object.
- **HEAD**: same as GET except that the server must return only the response header.
- **PATCH**: requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows:

Table 5-2 Common request headers

Parameter	Mandatory	Description	Example
Content-Type	Yes	Specifies the request body type or format. This field is mandatory and its default value is application/json . Other values of this field will be provided for specific APIs if any.	application/json
X-Auth-Token	Yes	Specifies a user token only for token-based API authentication. The user token is a response to the API used to obtain a user token . This API is the only one that does not require authentication.	-
X-Language	Yes	Request language	en-us

Request Body

The body of a request is often sent in a structured format as specified in **Content-Type**. The request body transfers content except the request header.

The request body varies according to APIs. Certain APIs do not require the request body, such as **GET** and **DELETE**.

For the **Obtaining a User Token** API, obtain the request parameters and parameter description in the API request. The following provides an example request with a body included. Replace *username*, *domainname*, ******* (login password), and *xxxxxxxxxxxxxxxxxxx* (project name) with the actual values. You can obtain the endpoint by referring to **Regions and Endpoints**.

NOTE

The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under an account. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see **Obtaining a User Token**.

If all data required for the API request is available, you can send the request to call the API through [curl](#), [Postman](#), or code. In the response to the API used to [obtain a user token](#), **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

5.2 Authentication

5.2.1 Authentication of Management Plane APIs

Requests for calling a management plane API of GES can be authenticated using either of the following methods:

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. AK/SK-based authentication is recommended because it is more secure.

Token-based Authentication

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

When calling the API to [obtain a user token](#), you must set **auth.scope** in the request body to **project**.

For details about how to obtain *username* and *domainname*, see [Obtaining the Account Name and Account ID](#). *password* indicates the user password.

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****#",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxx"
      }
    }
  }
}
```

NOTE

- The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.
- To obtain the token, the GES **scope** must be **project** (cannot be **domain**).

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request:

AK/SK-based Authentication

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

NOTE

AK/SK-based authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token-based authentication is recommended.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK-based authentication, you can use an AK/SK to sign requests based on the signature algorithm or use the signing SDK to sign requests.

NOTE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

5.2.2 Authentication of Service Plane APIs

Calling a service plane API of GES can only be authenticated using tokens.

Token-based Authentication

A token is used to grant temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

When calling the API to [obtain a user token](#), you must set **auth.scope** in the request body to **project**.

For details about how to obtain *username* and *domainname*, see [Obtaining the Account Name and Account ID](#). *password* indicates the user password.

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****#",
          "domain": {
            "name": "domainname"
          }
        }
      }
    }
  }
}
```

```
    },  
    "scope": {  
      "project": {  
        "name": "xxxxxxxxx"  
      }  
    }  
  }  
}
```

NOTE

- A token is valid for 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.
- To obtain the token, the GES **scope** must be **project** (cannot be **domain**).

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request:

5.3 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see [Status Codes](#).

If status code **201** is returned for the calling of the IAM API used to [obtain a user token](#), the request is successful.

Response Header

Similar to a request, a response also has a header, for example, **Content-Type**. [Table 5-3](#) list the response header parameters.

Table 5-3 Response header parameters

Parameter	Description
Content-Length	Specifies the length (in bytes) of the response body.
Date	Specifies the time when a response is returned.
Content-type	Specifies the MIME type of the response body.
TraceID	Specifies the ID returned by the request, facilitating fault locating.

[Figure 5-1](#) shows the response header fields for the IAM API used to [obtain a user token](#). The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

Figure 5-1 Header fields of the response to the request for obtaining a user token

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token → MIIYXQYJKoZIhvcNAQcCoIIYITJCCGEoCAQExDTALBgIghkgBZQMEAgEwgharBgkqhkiG9w00BBwGgghacBIIWmHsidG9rZW4iOansiZXhwaXJlc19hdCI6IiwMTktMDItMTNUMC
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkqjACgklqO1wi4JIGzrpd18LGXK5bdfq4lqHCYb8P4NaY0NYejcAgz/VeFYtLWT1GSO0zxKZmlQHq82HBqHdglZO9fuEebL5dMhdavj+33wEI
xHRCE9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXl1jipPEGA270g1FruooL6jgglFkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==
x-xss-protection → 1; mode=block;
```

Response Body

The body of a response is often returned in structured format as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to [obtain a user token](#).

```
{
  "token": {
    "expires_at": "2019-02-13T06:52:13.855000Z",
    "methods": [
      "password"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "region_id": "ap-southeast-1",
            .....
          }
        ]
      }
    ]
  }
}
```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
  "error_msg": "The format of message is error",
  "error_code": "AS.0001"
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

6 Management Plane APIs (V2)

6.1 System Management

6.1.1 Querying Quotas (1.0.0)

Function

This API is used to query the tenant quotas.

URI

GET /v2/{project_id}/graphs/quotas

Table 6-1 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Request Parameters

Table 6-2 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-3 Response body parameters

Parameter	Type	Description
quotas	quotas Object	Resource type list. This field is left blank when the request fails.

Table 6-4 quotas

Parameter	Type	Description
resources	Array of resources objects	GES resource quota list

Table 6-5 resources

Parameter	Type	Description
type	String	Quota type Available values are as follows: "graph" "backup" "metadata"
available	Integer	Number of available graphs When the value is -1 , the number of available graphs is unlimited.

Parameter	Type	Description
edge_volume	Integer	Number of available edges. This parameter is available only when type is graph . When the value is -1 , the number of available edges is unlimited.

Status code: 400

Table 6-6 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
error_msg	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.

Example Request

Query tenant quotas.

```
GET https://{Endpoint}/v2/{project_id}/graphs/quotas
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "quotas" : {
    "resources" : [ {
      "type" : "graph",
      "available" : 10,
      "edge_volume" : 7300
    }, {
      "type" : "backup",
      "available" : 100
    }, {
      "type" : "metadata",
      "available" : 86
    } ]
  }
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "GES.7006",
  "error_msg" : "An internal error occurs in the underlying service of the graph engine."
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.2 Graph Management

6.2.1 Listing Graphs (2.1.18)

Function

This API is used to query all graphs of the current tenant.

URI

GET /v2/{project_id}/graphs

Table 6-7 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Table 6-8 Query parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Start position of the request. The default value is 0 .
limit	No	Integer	Maximum number of resources displayed on a single page. The default value is 10 .

Request Parameters

Table 6-9 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-10 Response body parameters

Parameter	Type	Description
graph_count	Integer	Total number of graphs. Graph object. If the request fails, this parameter is left empty.
graphs	Array of graphs objects	Graph list. Graph object. If the request fails, this parameter is left empty.

Table 6-11 graphs

Parameter	Type	Description
id	String	Graph ID
name	String	Graph name

Parameter	Type	Description
created_by	String	IAM username
is_multi_az	String	Whether cross-AZ HA will be enabled
region_code	String	Region code
az_code	String	AZ code
schema_path	Array of schema_path objects	Path for storing the metadata file
edgeset_path	Array of edgeset_path objects	OBS path for storing the edge data set
vertexset_path	Array of vertexset_path objects	OBS path for storing the vertex data set
edgeset_format	String	Format of the edge data file
edgeset_default_label	String	Default label of the edge data file
vertexset_format	String	Format of the vertex data file
vertexset_default_label	String	Default label of the vertex data file
data_store_version	String	Graph version
sys_tags	Array of strings	Enterprise project information. If this parameter is not specified, this function is disabled (default).

Parameter	Type	Description
status	String	Status code of a graph • 100: indicates that a graph is being prepared. • 200: indicates that a graph is running. • 201: indicates that a graph is upgrading. • 202: indicates that a graph is being imported. • 203: indicates that a graph is being rolled back. • 204: indicates that a graph is being exported. • 205: indicates that a graph is being cleared. • 206: indicates that the size is being prepared to change. • 207: indicates that the size is being changed. • 208: indicates that the size is being rolled back. • 210: Preparing for expansion • 211: Expanding • 300: indicates that a graph is faulty. • 303: indicates that a graph fails to be created. • 400: indicates that a graph is deleted. • 800: indicates that a graph is frozen. • 900: indicates that a graph is stopped. • 901: indicates that a graph is being stopped. • 920: indicates that a graph is being started.
action_progress	String	Progress of graph creation NOTE This field is returned only when status is 100.
graph_size_type_index	String	Graph size type index • 0: Ten-thousand-edge • 1: Million-edge • 2: Ten-million-edge • 3: Hundred-million-edge • 4: Billion-edge • 5: Ten-billion-edge • 6: Hundred-billion-edge • 401: Billion-edge-pro

Parameter	Type	Description
vpc_id	String	VPC ID
subnet_id	String	Subnet ID in the VPC
security_group_id	String	Security group ID
replication	Integer	Number of replicas. The default value is 1 .
created	String	Graph creation time (UTC).
updated	String	Graph update time (UTC).
private_ip	String	Floating IP address of a graph instance. Users can access the instance using the IP address through the ECS deployed on a private network.
public_ip	String	Public network access address of a graph instance. Users can access the instance using the IP address from the Internet.
arch	String	Graph instance's CPU architecture type. The value can be x86_64 or aarch64 .
master_key_id	String	User master key ID
master_key_name	String	User master key name
enable_rbac	Boolean	Whether to enable granular permission control
enable_full_text_index	Boolean	Whether to enable full-text indexes
enable_hyg	Boolean	Whether to enable HyG. This parameter is available only for database edition graphs.
traffic_ip_list	Array of strings	Physical addresses of a graph instance for access from private networks. To prevent service interruption caused by floating IP address switchover, poll the physical IP addresses to access the graph instance.
crypt_algorithm	String	Graph instance cryptography algorithm. The options are: • generalCipher: Chinese cryptographic algorithm • SMcompatible: Commercial cryptography algorithm (compatible with international ones)
enable_https	Boolean	Whether to enable the security mode. This mode may damage GES performance greatly.

Parameter	Type	Description
tags	Array of tags objects	Tag list. Each tag is in <key,value> format.
product_type	String	Graph type. The value can be InMemory (default value) or Persistence . If graph_size_type_index is 6 , the value must be Persistence . • InMemory: memory edition • Persistence: database version
vertex_id_type	vertex_id_type object	ID type of vertices. This parameter is mandatory only for database edition graphs. NOTE The vertex ID type cannot be changed once set. Exercise caution when setting this parameter.
origin_graph_size_type_index	String	Initial size of a graph. This parameter is returned in versions later than 2.3.15.
expand_time	String	Graph expansion time (UTC).
resize_time	String	Graph resize time (UTC).
capacity_ratio	Integer	Capacity rate of a graph. This parameter is only available for Ten-billion-edge graphs of the database edition since version 2.3.18.
sort_key_type	String	Type of the sort key for a database edition graph.
enable_lts	Boolean	Whether the interconnection with LTS is enabled. The options are: • true: The interconnection is enabled. • false: The interconnection is disabled.

Table 6-12 schema_path

Parameter	Type	Description
job_id	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file import status. The options are: • success • partiallyFailed • failed

Table 6-13 edgeset_path

Parameter	Type	Description
job_id	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file import status. The options are: • success • partiallyFailed • failed

Table 6-14 vertexset_path

Parameter	Type	Description
job_id	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file import status. The options are: • success • partiallyFailed • failed

Table 6-15 tags

Parameter	Type	Description
key	String	Tag key
value	String	Tag value

Table 6-16 vertex_id_type

Parameter	Type	Description
id_type	String	Vertex ID type. The value can be fixedLengthString or hash . • fixedLengthString: Vertex IDs are used for internal storage and compute. Specify the length limit. If the IDs are too long, the query performance can be reduced. Specify the length limit based on your dataset vertex IDs. • hash: Vertex IDs are converted into hash code for storage and compute. There is no limit on the ID length. However, there is an extremely low probability, approximately $10^{(-43)}$, that the vertex IDs will conflict. If you cannot determine the maximum length of a vertex ID, set this parameter to Hash.
id_length	Integer	This parameter is mandatory if id_type is fixedLengthString . The value ranges from 1 to 128.

Status code: 400**Table 6-17** Response body parameters

Parameter	Type	Description
error_msg	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.
error_code	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.

Example Request

View the first 10 graphs of the current tenant.

```
GET https://{Endpoint}/v2/{project_id}/graphs?offset=0&limit=10
```


Example Response

Status code: 200

Example response for a successful request

```
{
  "graph_count" : 2,
  "graphs" : [ {
    "id" : "146227d2-bfac-499a-97df-df475349e43d",
    "name" : "TenThousand_Charging",
    "created_by" : "ei_ges_j00356469_01",
    "is_multi_az" : "false",
    "region_code" : "cn-north-7",
    "az_code" : "cn-north-7c",
    "edgeset_format" : "",
    "edgeset_default_label" : "",
    "vertexset_format" : "",
    "vertexset_default_label" : "",
    "data_store_version" : "2.3.2",
    "sys_tags" : [ "0" ],
    "status" : "200",
    "graph_size_type_index" : "0",
    "vpc_id" : "0ac6e3c3-2c9b-4296-84f7-6883cebc7b41",
    "subnet_id" : "2b1755eb-d6d4-421f-88c0-cf6f0bc16801",
    "security_group_id" : "7aa7c8c9-7443-4a01-abf5-8064b586f8f5",
    "replication" : 1,
    "created" : "2022-04-26T02:19:54",
    "private_ip" : "192.168.0.228",
    "arch" : "x86_64",
    "master_key_id" : "",
    "master_key_name" : "",
    "enable_rbac" : false,
    "enable_full_text_index" : false,
    "enable_hyg" : false,
    "traffic_ip_list" : [ "192.168.0.228" ],
    "crypt_algorithm" : "generalCipher",
    "enable_https" : false,
    "origin_graph_size_type_index" : "1",
    "expand_time" : "2023-08-03T02:10:52",
    "resize_time" : "2023-08-02T02:10:52"
  }, {
    "id" : "1172f16c-63c7-4746-89b0-78972eddf706",
    "name" : "GES_UL_2_0_1",
    "created_by" : "ei_ges_j00356469_01",
    "is_multi_az" : "false",
    "region_code" : "cn-north-7",
    "az_code" : "cn-north-7c",
    "edgeset_format" : "",
    "edgeset_default_label" : "",
    "vertexset_format" : "",
    "vertexset_default_label" : "",
    "data_store_version" : "2.0.1",
    "sys_tags" : [ "0" ],
    "status" : "900",
    "graph_size_type_index" : "0",
    "vpc_id" : "0ac6e3c3-2c9b-4296-84f7-6883cebc7b41",
    "subnet_id" : "2b1755eb-d6d4-421f-88c0-cf6f0bc16801",
    "security_group_id" : "7aa7c8c9-7443-4a01-abf5-8064b586f8f5",
    "replication" : 1,
    "created" : "2022-04-26T02:10:52",
    "private_ip" : "192.168.0.123",
    "enable_rbac" : false,
    "enable_full_text_index" : false,
    "enable_hyg" : false,
    "traffic_ip_list" : [ "192.168.0.123" ],
    "crypt_algorithm" : "generalCipher",
    "enable_https" : false,
    "origin_graph_size_type_index" : "1",
    "expand_time" : "2023-08-03T02:10:52",
```

```
"resize_time": "2023-08-02T02:10:52"
} ]
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "GES.7006",
  "error_msg" : "An internal error occurs in the underlying service of the graph engine."
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.2.2 Querying Graph Details (1.0.0)

Function

This API is used to query graph details by graph ID.

URI

GET /v2/{project_id}/graphs/{graph_id}

Table 6-18 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-19 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-20 Response body parameters

Parameter	Type	Description
graph	graph object	Graph object. If the request fails, this parameter is left empty.

Table 6-21 graph

Parameter	Type	Description
id	String	Graph ID
name	String	Graph name
created_by	String	IAM username
is_multi_az	String	Whether to enable cross-AZ HA
region_code	String	Region code
az_code	String	AZ code
schema_path	Array of schema_path objects	Path for storing the metadata file
edgeset_path	Array of edgeset_path objects	OBS path for storing the edge data set

Parameter	Type	Description
vertexset_path	Array of vertexset_path objects	OBS path for storing the vertex data set
edgeset_format	String	Format of the edge data file
edgeset_default_label	String	Default label of the edge data file
vertexset_format	String	Format of the vertex data file
vertexset_default_label	String	Default label of the vertex data file
data_store_version	String	Graph version
sys_tags	Array of strings	Enterprise project information. If this parameter is not specified, this function is disabled (default).

Parameter	Type	Description
status	String	Status code of a graph: ● 100: Indicates that a graph is being prepared. ● 200: indicates that a graph is running. ● 201: indicates that a graph is upgrading. ● 202: indicates that a graph is being imported. ● 203: indicates that a graph is being rolled back. ● 204: indicates that a graph is being exported. ● 205: indicates that a graph is being cleared. ● 206: indicates that the graph is being prepared to resize. ● 207: indicates that the graph is being resized. ● 208: indicates that the size is being rolled back. ● 210: Preparing for expansion ● 211: Expanding ● 300: indicates that a graph is faulty. ● 303: indicates that a graph fails to be created. ● 400: indicates that a graph is deleted. ● 800: indicates that a graph is frozen. ● 900: indicates that a graph is stopped. ● 901: indicates that a graph is being stopped. ● 920: indicates that a graph is being started.
action_progress	String	Progress of graph creation NOTE This field is returned only when status is 100.
graph_size_type_index	String	Graph size type index: ● 0: Ten-thousand-edge ● 1: Million-edge ● 2: Ten-million-edge ● 3: Hundred-million-edge ● 4: Billion-edge ● 5: Ten-billion-edge ● 6: Hundred-billion-edge ● 401: Billion-edge-pro

Parameter	Type	Description
vpc_id	String	VPC ID
subnet_id	String	Subnet ID in the VPC
security_group_id	String	Security group ID
replication	Integer	Number of replicas. The default value is 1 .
created	String	Graph creation time (UTC).
updated	String	Graph update time (UTC).
private_ip	String	Floating IP address of a graph instance. Users can access the instance using the IP address through the ECS deployed on a private network.
public_ip	String	Public network access address of a graph instance. Users can access the instance using the IP address from the Internet.
arch	String	Graph instance's CPU architecture type. The value can be x86_64 or aarch64 .
master_key_id	String	User master key ID
master_key_name	String	User master key name
enable_rbac	Boolean	Whether to enable granular permission control
enable_full_text_index	Boolean	Whether to enable full-text indexes
enable_hyg	Boolean	Whether to enable HyG. This parameter is available only for database edition graphs.
traffic_ip_list	Array of strings	Physical addresses of a graph instance for access from private networks. To prevent service interruption caused by floating IP address switchover, poll the physical IP addresses to access the graph instance.
crypt_algorithm	String	Graph instance cryptography algorithm. The options are: • generalCipher: Chinese cryptographic algorithm • SMcompatible: Commercial cryptography algorithm (compatible with international ones)
enable_https	Boolean	Whether to enable the security mode. This mode may damage GES performance greatly.

Parameter	Type	Description
tags	Array of tags objects	Tag list. Each tag is in <key,value> format.
product_type	String	Graph type. The value can be InMemory (default value) or Persistence . If graph_size_type_index is 6 , the value must be Persistence . • InMemory: memory edition • Persistence: database version
vertex_id_type	vertex_id_type object	ID type of vertices. This parameter is mandatory only for database edition graphs. NOTE The vertex ID type cannot be changed once set. Exercise caution when setting this parameter.
origin_graph_size_type_index	String	Initial size of a graph. This parameter is returned in versions later than 2.3.15.
expand_time	String	Graph expansion time (UTC).
resize_time	String	Graph resize time (UTC).
capacity_ratio	Integer	Capacity rate of a graph. This parameter is only available for Ten-billion-edge graphs of the database edition since version 2.3.18.
sort_key_type	String	Type of the sort key for a database edition graph.
enable_lts	Boolean	Whether the interconnection with LTS is enabled. The options are: • true: The interconnection is enabled. • false: The interconnection is disabled.

Table 6-22 schema_path

Parameter	Type	Description
job_id	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file import status. The options are: • success • partiallyFailed • failed

Table 6-23 edgeset_path

Parameter	Type	Description
job_id	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file import status. The options are: • success • partiallyFailed • failed

Table 6-24 vertexset_path

Parameter	Type	Description
job_id	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file import status. The options are: • success • partiallyFailed • failed

Table 6-25 tags

Parameter	Type	Description
key	String	Tag key
value	String	Tag value

Table 6-26 vertex_id_type

Parameter	Type	Description
id_type	String	Vertex ID type. The value can be fixedLengthString or hash . • fixedLengthString: Vertex IDs are used for internal storage and compute. Specify the length limit. If the IDs are too long, the query performance can be reduced. Specify the length limit based on your dataset vertex IDs. • hash: Vertex IDs are converted into hash code for storage and compute. There is no limit on the ID length. However, there is an extremely low probability, approximately $10^{(-43)}$, that the vertex IDs will conflict. If you cannot determine the maximum length of a vertex ID, set this parameter to Hash.
id_length	Integer	This parameter is mandatory if id_type is fixedLengthString . The value ranges from 1 to 128.

Status code: 400**Table 6-27** Response body parameters

Parameter	Type	Description
error_code	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.
error_msg	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.

Example Request

Query graph details by graph ID.

GET https://{Endpoint}/v2/{project_id}/graphs/{graph_id}

Example Response

Status code: 200

Example response for a successful request

```
{
  "graph" : {
    "id" : "1172f16c-63c7-4746-89b0-78972eddf706",
    "name" : "GES_UI_2_0_1",
    "created_by" : "ei_ges_j00356469_01",
    "is_multi_az" : "false",
    "region_code" : "cn-north-7",
    "az_code" : "cn-north-7c",
    "edgeset_format" : "",
    "edgeset_default_label" : "",
    "vertexset_format" : "",
    "vertexset_default_label" : "",
    "data_store_version" : "2.0.1",
    "sys_tags" : [ "0" ],
    "status" : "900",
    "graph_size_type_index" : "0",
    "vpc_id" : "0ac6e3c3-2c9b-4296-84f7-6883cebc7b41",
    "subnet_id" : "2b1755eb-d6d4-421f-88c0-cf6f0bc16801",
    "security_group_id" : "7aa7c8c9-7443-4a01-abf5-8064b586f8f5",
    "replication" : 0,
    "created" : "2022-04-26T02:10:52",
    "updated" : "2022-04-26T02:10:52",
    "private_ip" : "192.168.0.123",
    "enable_rbac" : false,
    "enable_full_text_index" : false,
    "enable_hyg" : false,
    "traffic_ip_list" : [ "192.168.0.123" ],
    "crypt_algorithm" : "generalCipher",
    "enable_https" : false,
    "origin_graph_size_type_index": "1",
    "expand_time": "2023-08-03T02:10:52",
    "resize_time": "2023-08-02T02:10:52"
  }
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "GES.7006",
  "error_msg" : "An internal error occurs in the underlying service of the graph engine."
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.2.3 Creating a Graph (2.2.2)

Function

This API is used to create a graph.

URI

POST /v2/{project_id}/graphs

Table 6-28 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Request Parameters

Table 6-29 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-30 Request body parameter

Parameter	Mandatory	Type	Description
graph	Yes	graph object	Graph type

Table 6-31 graph

Parameter	Mandatory	Type	Description
name	Yes	String	Graph name. The name can contain 4 to 50 characters and must start with a letter. Only letters, numbers, hyphens (-), and underscores (_) are allowed.
graph_size_type_index	Yes	String	Graph size type index: • 0: Ten-thousand-edge • 1: Million-edge • 2: Ten-million-edge • 3: Hundred-million-edge • 4: Billion-edge • 5: Ten-billion-edge • 6: Hundred-billion-edge • 401: Billion-edge-pro
arch	No	String	Graph instance's CPU architecture type. The value can be x86_64 or aarch64 . The default value is x86_64 . • x86_64: x86 64-bit architecture • aarch64: Arm 64-bit architecture
vpc_id	Yes	String	VPC ID
subnet_id	Yes	String	Subnet ID in the VPC
security_group_id	Yes	String	Security group ID
public_ip	No	public_ip object	Public IP address. If the parameter is not specified, public connection is not used by default.
enable_multi_az	No	Boolean	Whether the created graph supports the cross-AZ mode. The default value is false . If the value is true , the system will create the ECSs in the graph in two AZs. If this parameter is not specified when you create a graph, all ECSs in the graph are created in one AZ.

Parameter	Mandatory	Type	Description
sys_tags	No	Array of sys_tags objects	Enterprise project information. If this parameter is not specified, this function is disabled (default).
tags	No	Array of tags objects	TMS tags for expenses. This function is disabled by default.
enable_rbac	No	Boolean	Whether to enable granular permission control for the created graph. The default value is false , indicating that granular permission control is disabled. If this parameter is set to true , no user has the permission to access the graph. To access the graph, you need to call the granular permission control API of the service plane to set the required permissions.
enable_full_text_index	No	Boolean	Whether to enable full-text index control for the created graph. The default value is false , indicating that full-text index control is disabled. If this parameter is set to true , full-text indexes are available for Billion-edge-pro graphs, and a Cloud Search Service (CSS) cluster will be created when you create a graph.
enable_hyg	No	Boolean	Whether to enable HyG for the graph. This parameter is available for database edition graphs only.
crypt_algorithm	No	String	Graph instance cryptography algorithm. The options are: • generalCipher: Chinese cryptographic algorithm • SMcompatible: Commercial cryptography algorithm (compatible with international ones)
enable_https	No	Boolean	Whether to enable the security mode. This mode may damage GES performance greatly.

Parameter	Mandatory	Type	Description
product_type	No	String	Graph type. The value can be InMemory (default value) or Persistence . If graph_size_type_index is 6, the value must be Persistence . • InMemory: memory edition • Persistence: database version
vertex_id_type	No	vertex_id_type object	ID type of vertices. This parameter is mandatory only for database edition graphs. NOTE The vertex ID type cannot be changed once set. Exercise caution when setting this parameter.
capacity_ratio	No	Integer	Capacity rate of a graph. This parameter is only available for Ten-billion-edge graphs of the database edition since version 2.3.18.
is_dynamic_graph	No	Boolean	Whether a graph is a dynamic one
sort_key_type	No	String	Type of the sort key for a database edition graph. Different sort key values are configured to distinguish duplicate edges (edges with the same source vertex, end vertex, and label). This parameter is required only for database edition graphs. The options are: • int: an integer. • string: a string of less than 40 bytes. • varString: The length is not limited. However, if the value is too long, the read and write performance is affected. It is recommended that the length be within 1 KB bytes, with a maximum of 2 KB bytes.

Table 6-32 public_ip

Parameter	Mandatory	Type	Description
public_bind_type	No	String	Binding type of an EIP. The value can be either of the following: • auto_assign • bind_existing
eip_id	No	String	ID of an EIP. When publicBindType is set to bind_existing , its value is the ID of an EIP that has been created but has not been bound. When publicBindType is set to auto_assign , its value is set to null.

Table 6-33 sys_tags

Parameter	Mandatory	Type	Description
key	No	String	Key of the enterprise project. The value is _sys_enterprise_project_id .
value	No	String	Enterprise project ID. You can obtain it from the enterprise project.

Table 6-34 tags

Parameter	Mandatory	Type	Description
key	No	String	Tag key
value	No	String	Tag value

Table 6-35 vertex_id_type

Parameter	Mandatory	Type	Description
id_type	Yes	String	Vertex ID type. The options are fixedLengthString, varString, and hash. • fixedLengthString: In the variable length string format, vertex IDs are used for internal storage and compute. Specify the length limit. If the IDs are too long, the query performance can be reduced. Specify the length limit based on your dataset vertex IDs. If you cannot determine the maximum length, set the ID type to hash. • varString: In the variable length string format, the length of the vertex IDs written by the user is not limited. However, if the IDs are too long, the read and write performance is affected. It is recommended that the length be within 1 KB, with a maximum of 4 KB. • hash: Vertex IDs are converted into hash code for storage and compute. There is no limit on the ID length. However, there is an extremely low probability, approximately $10^{(-43)}$, that the vertex IDs will conflict. If you cannot determine the maximum length of a vertex ID, set this parameter to Hash. NOTE When creating a HyG graph, id_type can only be fixedLengthString.
id_length	No	Integer	This parameter is mandatory if id_type is fixedLengthString. The value ranges from 1 to 128.

Response Parameters

Status code: 200

Table 6-36 Response body parameters

Parameter	Type	Description
id	String	Graph ID
name	String	Graph name

Status code: 400

Table 6-37 Response body parameters

Parameter	Type	Description
error_msg	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
error_code	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.

Example Request

Create a graph. The graph name is **demo**, the graph size is Million-edge, the graph instance CPU architecture type is x86 64-bit, the VPC ID is **2d8af840-fd57-4e3b-a8f1-cda0f55ccd99**, the subnet ID is **dc018ec3-67d1-46c9-b2fc-19d83367f4e2**, and the security group ID is **11d27338-8649-4076-8579-5ebc1a60f79e**.

POST https://{Endpoint}/v2/{project_id}/graphs

```
{
  "graph": {
    "name": "demo",
    "graph_size_type_index": "1",
    "arch": "x86_64",
    "vpc_id": "2d8af840-fd57-4e3b-a8f1-cda0f55ccd99",
    "subnet_id": "dc018ec3-67d1-46c9-b2fc-19d83367f4e2",
    "security_group_id": "11d27338-8649-4076-8579-5ebc1a60f79e",
    "public_ip": {
      "public_bind_type": "bind_existing",
      "eip_id": "30ef2d58-08a9-4481-b526-b2cbe67d020d"
    }
  }
}
```

```
},
"enable_multi_az" : false,
"sys_tags" : [ {
  "key" : "_sys_enterprise_project_id",
  "value" : "54c0b33c-8627-462f-948e-bae08c0887b4"
} ],
"crypt_algorithm" : "generalCipher",
"enable_https" : false
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "id" : "f1529b88-c958-493e-8452-fccfe932cde1",
  "name" : "demo"
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "GES.7016",
  "error_msg" : "The parameter [subnetId] is not exist."
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.2.4 Stopping a Graph (1.0.0)

Function

This API is used to stop a graph. After a graph is created, you can stop it if it is temporarily not in use.

 NOTE

- Graphs of the database edition cannot be stopped by calling this API.
- Instances of stopped graphs are not charged.
- You can stop a graph instance for a maximum of seven days. If you do not restart the instance manually within that time frame, it will automatically restart.

URI

POST /v2/{project_id}/graphs/{graph_id}/stop

Table 6-38 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-39 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-40 Response body parameters

Parameter	Type	Description
job_id	String	ID of the graph stopping job. Graph object. If the request fails, this parameter is left empty.

Table 6-41 Response body parameters

Parameter	Type	Description
error_msg	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.
error_code	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.

Example Request

Close a graph.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/stop
{ }
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "job_id" : "ff8080816025a0a1016025a5a2700007"
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "GES.7001",
  "error_msg" : "The graph is not running."
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.

Return Value	Description
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.2.5 Starting a Graph (1.0.0)

Function

This API is used to start a graph. You can disable a graph if it is not used temporarily.

URI

POST /v2/{project_id}/graphs/{graph_id}/start

Table 6-42 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-43 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-44 Request body parameters

Parameter	Mandatory	Type	Description
graph_backup_id	No	String	Backup ID associated during graph startup. If this parameter is configured, the graph starts from the backup. If this parameter is left blank, the graph starts from the status when it was closed last time.

Response Parameters

Status code: 200

Table 6-45 Response body parameters

Parameter	Type	Description
job_id	String	ID of the graph startup job. This parameter is left blank when the request fails.

Status code: 400

Table 6-46 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Start a graph whose associated backup ID is **08a898ae-3ff8-40e8-a7ed-03afe05aedbb**.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/start
```

```
{
  "graph_backup_id" : "08a898ae-3ff8-40e8-a7ed-03afe05aedbb"
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "job_id" : "ff8080816025a0a1016025a5a2700007"
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "GES.7006",
  "error_msg" : "An internal error occurs in the underlying service of the graph engine."
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.2.6 Deleting a Graph (1.0.0)

Function

This API is used to delete a graph.

URI

DELETE /v2/{project_id}/graphs/{graph_id}

Table 6-47 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 6-48 Query parameters

Parameter	Mandatory	Type	Description
keep_backup	No	Boolean	Whether to retain the backups of a graph after it is deleted. By default, one automatic backup and two manual backups are retained. If this parameter is left empty, no backups are retained.
delete_eip	No	Boolean	Whether the EIP is concurrently deleted

Request Parameters

Table 6-49 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-50 Response body parameters

Parameter	Type	Description
job_id	String	ID of the graph deletion job. This parameter is left blank when the request fails.

Status code: 400**Table 6-51** Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Delete a graph.

```
DELETE https://{Endpoint}/v2/{project_id}/graphs/{graph_id}
```

Example Response**Status code: 200**

Example response for a successful request

```
{
  "job_id": "ff8080816025a0a1016025a5a2700007"
}
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
  "error_code": "GES.0047"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.2.7 Incrementally Importing Data to a Graph (2.1.14)

Function

This API is used to import data to graphs incrementally.

NOTE

1. To ensure successful data recovery during system restarts, do not delete any graph data stored in OBS while using the graph.
2. The size of a single file in the import directory or the size of a single file to be imported cannot exceed 5 GB. Or the import will fail. You are advised to split the file into multiple files smaller than 5 GB before importing.

URI

POST /v2/{project_id}/graphs/{graph_id}/import-graph

Table 6-52 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-53 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-54 Request body parameters

Parameter	Mandatory	Type	Description
edgeset_path	No	String	Edge file directory or name This parameter is mandatory when vertexset_path and schema_path are both left unspecified.
edgeset_format	No	String	Edge dataset format, which can currently be set to csv or txt , with csv as default.
vertexset_path	No	String	Vertex file directory or name This parameter is mandatory when edgeset_path and schema_path are both left unspecified.
vertexset_format	No	String	Vertex dataset format, which can currently be set to csv or txt , with csv as default.
schema_path	No	String	Path for storing the metadata file of the new data. This parameter is mandatory when edgeset_path and vertexset_path are both left unspecified.
log_dir	No	String	Directory for storing logs of imported graphs. This directory stores the data that fails to be imported during graph creation and detailed error causes.

Parameter	Mandatory	Type	Description
parallel_edge	No	parallel_edge object	How to process repetitive edges.
delimiter	No	String	Field separator in a CSV file. The default value is comma (,). The default element separator in a field of the list/set type is semicolon (;).
trim_quote	No	String	Field quote character in a CSV file. The default value is double quotation marks ("). They are used to enclose a field if the field contains separators or line breaks.
offline	No	Boolean	Whether offline import is selected. The value can be true or false . The default value is false . • true: Offline import is selected. The import speed is high, but the graph is locked and cannot be read or written during the import. • false: Online import is selected. Compared with offline import, online import is slower. However, the graph can be read (cannot be written) during the import.

Table 6-55 parallel_edge

Parameter	Mandatory	Type	Description
action	No	String	Processing mode of repetitive edges. The value can be allow , ignore , or override . The default value is allow . • allow indicates that repetitive edges are allowed. • ignore indicates that subsequent repetitive edges are ignored. • override indicates that the previous repetitive edges are overwritten.

Parameter	Mandatory	Type	Description
ignore_label	No	Boolean	Whether to ignore labels on repetitive edges. The value can be true or false. The default value is true. • true: Indicates that the repetitive edge definition does not contain the label. That is, the <source vertex, target vertex> indicates an edge, excluding the label information. • false: Indicates that the repetitive edge definition contains the label. That is, the <source vertex, target vertex, label> indicates an edge.
sort_key_column	No	String	Position of the sort key in the edge file, which can only be set to lastColumn. If the edge file does not contain a sort key, this parameter is not required. Function of the sort key: Different sort key values are configured to distinguish duplicate edges (edges with the same source vertex, end vertex, and label). This parameter is required only for database edition graphs.

Response Parameters

Status code: 200

Table 6-56 Response body parameters

Parameter	Type	Description
job_id	String	ID of an asynchronous job

Status code: 400

Table 6-57 Response body parameters

Parameter	Type	Description
error_code	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.
error_msg	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.

Example Request

Incrementally import graph data. The edge file directory is **testbucket/demo_movie/edges/** and the edge data set is in CSV format. The vertex file directory is **testbucket/demo_movie/vertices/** and the vertex data set is in CSV format.

POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/import-graph

```
{
  "edgeset_path" : "testbucket/demo_movie/edges/",
  "edgeset_format" : "csv",
  "vertexset_path" : "testbucket/demo_movie/vertices/",
  "vertexset_format" : "csv",
  "schema_path" : "testbucket/demo_movie/incremental_data_schema.xml",
  "log_dir" : "testbucket/importlogdir",
  "parallel_edge" : {
    "action" : "override",
    "ignore_label" : true
  },
  "delimiter" : ";",
  "trim_quote" : "\"",
  "offline" : false
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "job_id" : "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

Status code: 400

Example response for a failed request

```
{
  "error_msg" : "parameter format error",
  "error_code" : "GES.8013"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.2.8 Exporting a Graph (1.0.5)

Function

This API is used to export a graph.

NOTE

The database edition 2.3.14 or later supports this function.

URI

POST /v2/{project_id}/graphs/{graph_id}/export-graph

Table 6-58 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-59 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-60 Request body parameters

Parameter	Mandatory	Type	Description
graph_export_path	Yes	String	OBS path to which a graph is exported
edge_set_name	Yes	String	Name of the folder where the edges are exported
vertex_set_name	Yes	String	Name of the folder where the vertices are exported
schema_name	Yes	String	Name of the exported metadata file
paginate	No	paginate object	Pagination-related parameters. In memory edition 2.3.11 or later, graphs are exported on multiple pages by default.

Table 6-61 paginate

Parameter	Mandatory	Type	Description
enable	No	Boolean	Whether to enable pagination. The default value is true . If pagination is not required, set this parameter to false . NOTE This parameter is unavailable for database editions.

Parameter	Mandatory	Type	Description
row_count_per_file	No	Integer	Maximum number of rows in each file when graphs are exported by page. The default value is 10000000 .
num_thread	No	Integer	Number of concurrent threads when graphs are exported by page. The default value is 8 .

Response Parameters

Status code: 200

Table 6-62 Response body parameters

Parameter	Type	Description
job_id	String	ID of an asynchronous job

Status code: 400

Table 6-63 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Export a graph. The OBS path for exporting the graph is **demo_movie/**, the name of the exported edge file is **set_edge**, the name of the exported vertex file is **set_vertex**, and the name of the exported metadata file is **set_schema.xml**.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/export-graph
{
```

```
"graph_export_path" : "demo_movie/",  
"edge_set_name" : "set_edge",  
"vertex_set_name" : "set_vertex",  
"schema_name" : "set_schema.xml"  
}
```

Example Response

Status code: 200

Example response for a successful request

```
{  
  "job_id" : "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"  
}
```

Status code: 400

Example response for a failed request

```
{  
  "error_msg": "The export path cannot be empty.",  
  "error_code": "GES.7243"  
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.2.9 Clearing a Graph (2.1.2)

Function

This API is used to clear all data in a graph.

URI

POST /v2/{project_id}/graphs/{graph_id}/clear-graph

Table 6-64 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 6-65 Query parameters

Parameter	Mandatory	Type	Description
clear-metadata	No	Boolean	Whether to clear graph metadata. Set this parameter to true . The value can be true or false . The default value is false . • true: The metadata will be cleared. • false: The metadata will not be cleared.

Request Parameters

Table 6-66 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-67 Response body parameters

Parameter	Type	Description
job_id	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Status code: 400**Table 6-68** Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Clear all data in a graph.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/clear-graph?clear-metadata=true
{ }
```

Example Response**Status code: 200**

Example response for a successful request

```
{
  "job_id": "ff8080816025a0a1016025a5a2700007"
}
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
  "error_code": "GES.0047"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.2.10 Upgrading a Graph (1.0.5)

Function

This API is used to upgrade a graph. The GES version is periodically upgraded. You can upgrade your graphs as required.

URI

POST /v2/{project_id}/graphs/{graph_id}/upgrade

Table 6-69 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-70 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-71 Request body parameters

Parameter	Mandatory	Type	Description
upgrade_version	Yes	String	Target version, which must be later than the current version
force_upgrade	No	Boolean	Whether to upgrade forcibly. The value can be true or false . The default value is false . • true: forcible upgrades, which will interrupt running tasks, such as long algorithm execution tasks. As a result, a small number of requests may fail. • false: non-forcible upgrades, which will wait for running services to complete. The upgrade process may be slow.

Response Parameters

Status code: 200

Table 6-72 Response body parameters

Parameter	Type	Description
job_id	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Status code: 400**Table 6-73** Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Upgrade a graph to version 1.1.8 as needed.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/upgrade
{
  "upgrade_version" : "1.1.8",
  "force_upgrade" : false
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "job_id" : "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
}
```

```
"error_code": "GES.0047"  
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.2.11 Binding an EIP (1.0.6)

Function

This API enables you to access GES by binding an elastic IP (EIP).

URI

POST /v2/{project_id}/graphs/{graph_id}/bind-eip

Table 6-74 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-75 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-76 Request body parameters

Parameter	Mandatory	Type	Description
eip_id	Yes	String	ID of the elastic IP address. For details about how to query the EIP ID, see Querying an EIP .

Response Parameters

Status code: 200

None

Status code: 400

Table 6-77 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.

Example Request

Bind an EIP to access GES. The ID of the EIP is **02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042**.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/bind-eip
{
  "eip_id" : "02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042"
}
```

Example Response

Status code: 200

Example response for a successful request

```
{ }
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
  "error_code": "GES.0047"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.2.12 Unbinding an EIP (1.0.6)

Function

If you do not need to use the EIP, this API enabled you to unbind the EIP to release network resources.

URI

POST /v2/{project_id}/graphs/{graph_id}/unbind-eip

Table 6-78 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-79 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-80 Request body parameters

Parameter	Mandatory	Type	Description
eip_id	Yes	String	ID of the elastic IP address. For details about how to query the EIP ID, see Querying an EIP .

Response Parameters

Status code: 200

None

Status code: 400

Table 6-81 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.

Example Request

Unbind an EIP to release network resources. The ID of the EIP is **02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042**.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/unbind-eip
{
  "eip_id" : "02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042"
}
```

Example Response

Status code: 200

Example response for a successful request

```
{ }
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
  "error_code": "GES.0047"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found

Return Value	Description
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.2.13 Resizing a Graph (2.2.21)

Function

This API is used to resize a graph.

NOTE

Database edition graphs cannot be resized by calling this API.

URI

POST /v2/{project_id}/graphs/{graph_id}/resize

Table 6-82 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-83 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. Used to obtain the permission to use APIs. For how to obtain the token, see Authentication . (The token is the value of X-Subject-Token in the response header.)

Table 6-84 Request body parameter

Parameter	Mandatory	Type	Description
resize	Yes	resize object	Graph specifications after the graph is resized.

Table 6-85 [resize](#)

Parameter	Mandatory	Type	Description
graph_size_type_index	Yes	String	Graph size type. The value can be 1, 2, 3, 4, 401, and 5, indicating that a Million-edge graph, Ten-million-edge graph, Hundred-million-edge graph, Billion-edge graph, Billion-edge-pro graph, or Ten-billion-edge graph, respectively.

Response Parameters

Status code: 200

Table 6-86 Response body parameters

Parameter	Type	Description
job_id	String	ID of the graph resize task. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Status code: 400

Table 6-87 Response body parameters

Parameter	Type	Description
error_code	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
error_msg	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.

Example Request

Resize a graph. The graph size is **Ten-million-edge**.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/resize
{
  "resize": {
    "graph_size_type_index": "2"
  }
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "job_id": "ff8080816025a0a1016025a5a2700007"
}
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
  "error_code": "GES.0047"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.

Return Value	Description
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

6.2.14 Forcibly Restarting a Graph (2.2.21)

Function

This API is used to forcibly start a graph, especially for running graphs or those are being imported, exported, and cleared. If a graph is forcibly restarted, asynchronous tasks of the graph are failed state and the graph is stopped and started.

URI

POST /v2/{project_id}/graphs/{graph_id}/restart

Table 6-88 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-89 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-90 Response body parameters

Parameter	Type	Description
job_id	String	ID of a forcible restart job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

Status code: 400

Table 6-91 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Forcibly start a graph.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/restart
{ }
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "job_id" : "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

Status code: 400

Example response for a failed request

```
{
  "error_msg" : "The request is invalid.",
  "error_code" : "GES.7016"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.2.15 Expanding a Graph (2.2.23)

Function

This API is used to dynamically expand multiple secondary nodes to enable concurrent handling of increased read requests.

 NOTE

- This API is not supported by Ten-thousand-edge and Ten-billion-edge graphs.
- Graphs cannot be resized after expansion.
- To resize and expand a graph, first resize it and then expand it.
- Graphs of the database edition cannot be expanded by calling this API.

URI

POST /v2/{project_id}/graphs/{graph_id}/expand

Table 6-92 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-93 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-94 Request body parameter

Parameter	Mandatory	Type	Description
expand	Yes	expand object	expand is an object.

Table 6-95 expand

Parameter	Mandatory	Type	Description
replication	Yes	Integer	Number of new nodes to expand

Response Parameters

Status code: 200

Table 6-96 Response body parameters

Parameter	Type	Description
job_id	String	ID of the expansion job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

Status code: 400

Table 6-97 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.

Example Request

Expand a graph. The number of new replicas is 1.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/expand
```

```
{
  "expand":{
    "replication": 1
```

```
}  
}
```

Example Response

Status code: 200

Example response for a successful request

```
{  
  "job_id" : "ff8080816025a0a1016025a5a2700007"  
}
```

Status code: 400

Example response for a failed request

```
{  
  "error_code" : "GES.7015",  
  "error_msg" : "The graph is not running or stopped."  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.2.16 Changing the Security Group (2.4.3)

Function

This API is used to change the security group of a graph after the graph is created.

URI

POST /v2/{project_id}/graphs/{graph_id}/sg/change

Table 6-98 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-99 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. For how to obtain the token, see Authentication . (The token is the value of X-Subject-Token in the response header.)

Table 6-100 Request body parameter

Parameter	Mandatory	Type	Description
security_group_id	Yes	String	ID of the target security group

Response Parameters

Status code: 200

None

Status code: 404

Table 6-101 Response body parameters

Parameter	Type	Description
error_code	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.

Parameter	Type	Description
error_msg	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.

Example Request

Changing the security group of a graph instance

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/sg/change
{
  "security_group_id" : "b1038649-1f77-4ae9-b64d-9af56e422652"
}
```

Example Response

Status code: 200

Example response for a successful request

```
{ }
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
  "error_code": "GES.0047"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.3 Backup Management

6.3.1 Viewing the List of All Backups (1.0.0)

Function

This API is used to query a backup list according to search criteria. Before using this API:

URI

GET /v2/{project_id}/graphs/backups

Table 6-102 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Table 6-103 Query parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of resources displayed on a single page. The default value is 10 .
offset	No	Integer	Start position of the request. The default value is 0 .

Request Parameters

Table 6-104 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-105 Response body parameters

Parameter	Type	Description
backup_count	Integer	Total number of backups. This parameter is left blank when the request fails.
backup_list	Array of backup_list objects	List of all backups under the current project ID. This parameter is left blank when the request fails.

Table 6-106 backup_list

Parameter	Type	Description
id	String	Indicates the backup ID.
name	String	Backup name
backup_method	String	Backup method. The value can be auto , manual , or import .
graph_id	String	ID of the graph associated with the backup
graph_name	String	Name of the graph associated with the backup
graph_status	String	Status of the graph associated with the backup
graph_size_type_index	String	Size of the graph associated with the backup
data_store_version	String	Version of the graph associated with the backup
arch	String	CPU architecture of the graph node associated with the backup
status	String	Backup status: ● backing_up: indicates that a graph is being backed up. ● success: indicates that a graph is successfully backed up. ● failed: indicates that a graph fails to be backed up.
start_timestamp	Long	Start timestamp of a backup job

Parameter	Type	Description
start_time	String	Start time of a backup job.
product_type	String	Graph product type. The options are: • InMemory: memory edition • Persistence: database edition
end_timestamp	Long	End timestamp of a backup job
end_time	String	Indicates the backup end time.
size	Long	Backup file size (MB)
duration	Long	Backup duration (seconds)

Status code: 400

Table 6-107 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Query the list of all backups.

```
GET https://{Endpoint}/v2/{project_id}/graphs/backups?offset=0&limit=3
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "backup_count" : 3,
  "backup_list" : [ {
    "id" : "ada3e720-ab87-48cb-bff7-3ec5ae1a9652",
    "name" : "ges060803_nodelete-20210608135513",
```

```
"backup_method" : "manual",
"graph_id" : "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
"graph_name" : "ges060803_nodelete",
"graph_status" : "200",
"graph_size_type_index" : "1",
"data_store_version" : "2.2.21",
"arch" : "x86_64",
"status" : "success",
"start_timestamp" : 1623160513000,
"start_time" : "2021-06-08T13:55:13",
"end_timestamp" : 1623160568000,
"end_time" : "2021-06-08T13:56:08",
"size" : 1,
"duration" : 54
}, {
  "id" : "7ed3f51d-816d-4651-9129-fe21b64b5c91",
  "name" : "ges060803_nodelete_20210609203323_auto",
  "backup_method" : "auto",
  "graph_id" : "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
  "graph_name" : "ges060803_nodelete",
  "graph_status" : "200",
  "graph_size_type_index" : "1",
  "data_store_version" : "2.2.21",
  "arch" : "x86_64",
  "status" : "success",
  "start_timestamp" : 1623242004000,
  "start_time" : "2021-06-09T12:33:24",
  "end_timestamp" : 1623242004000,
  "end_time" : "2021-06-09T12:33:24",
  "size" : 1,
  "duration" : 0
}, {
  "id" : "604bfb46-04dd-45fc-a9ae-df24a0705b9d",
  "name" : "ges060802_nodelete-20210608135523",
  "backup_method" : "manual",
  "graph_id" : "9b9a05c2-0cdb-41ac-b55f-93caffb0519a",
  "graph_name" : "ges060802_nodelete",
  "graph_status" : "400",
  "graph_size_type_index" : "0",
  "data_store_version" : "2.2.23",
  "arch" : "x86_64",
  "status" : "success",
  "start_timestamp" : 1623160524000,
  "start_time" : "2021-06-08T13:55:24",
  "end_timestamp" : 1623160577000,
  "end_time" : "2021-06-08T13:56:17",
  "size" : 1,
  "duration" : 53
}]
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "GES.7006",
  "error_msg" : "An internal error occurs in the underlying service of the graph engine."
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed

Return Value	Description
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.3.2 Viewing the Backup List of a Graph (1.0.0)

Function

This API is used to query the backup list of a graph.

URI

GET /v2/{project_id}/graphs/{graph_id}/backups

Table 6-108 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 6-109 Query parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of resources displayed on a single page. The default value is 10 .
offset	No	Integer	Start position of the request. The default value is 0 .

Request Parameters

Table 6-110 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-111 Response body parameters

Parameter	Type	Description
backup_count	Integer	Total number of backups. This parameter is left blank when the request fails.
backup_list	Array of backup_list objects	Backup list of a graph under the current project ID. This parameter is left blank when the request fails.

Table 6-112 backup_list

Parameter	Type	Description
id	String	Backup ID
name	String	Backup name
backup_method	String	Backup method. The value can be auto , manual , or import .
graph_id	String	ID of the graph associated with the backup
graph_name	String	Name of the graph associated with the backup
graph_status	String	Status of the graph associated with the backup
graph_size_type_index	String	Size of the graph associated with the backup

Parameter	Type	Description
data_store_version	String	Version of the graph associated with the backup
arch	String	CPU architecture of the graph node associated with the backup
status	String	Backup status: • backing_up: indicates that a graph is being backed up. • success: indicates that a graph is successfully backed up. • failed: indicates that a graph fails to be backed up.
start_timestamp	Long	Start timestamp of a backup job
start_time	String	Start time of a backup job.
product_type	String	Graph product type. The options are: • InMemory: memory edition • Persistence: database edition
end_timestamp	Long	End timestamp of a backup job
end_time	String	Indicates the backup end time.
size	Long	Backup file size (MB)
duration	Long	Backup duration (seconds)

Status code: 400**Table 6-113** Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Query the backup list of a graph.

```
GET https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/backups?offset=0&limit=2
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "backup_count" : 2,
  "backup_list" : [ {
    "id" : "ada3e720-ab87-48cb-bff7-3ec5ae1a9652",
    "name" : "ges060803_nodelete-20210608135513",
    "backup_method" : "manual",
    "graph_id" : "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
    "graph_name" : "ges060803_nodelete",
    "graph_status" : "200",
    "graph_size_type_index" : "1",
    "data_store_version" : "2.2.21",
    "arch" : "x86_64",
    "status" : "success",
    "start_timestamp" : 1623160513000,
    "start_time" : "2021-06-08T13:55:13",
    "end_timestamp" : 1623160568000,
    "end_time" : "2021-06-08T13:56:08",
    "size" : 1,
    "duration" : 54
  }, {
    "id" : "7ed3f51d-816d-4651-9129-fe21b64b5c91",
    "name" : "ges060803_nodelete_20210609203323_auto",
    "backup_method" : "auto",
    "graph_id" : "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
    "graph_name" : "ges060803_nodelete",
    "graph_status" : "200",
    "graph_size_type_index" : "1",
    "data_store_version" : "2.2.21",
    "arch" : "x86_64",
    "status" : "success",
    "start_timestamp" : 1623242004000,
    "start_time" : "2021-06-09T12:33:24",
    "end_timestamp" : 1623242004000,
    "end_time" : "2021-06-09T12:33:24",
    "size" : 1,
    "duration" : 0
  } ]
}
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
  "error_code": "GES.0047"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.3.3 Adding a Backup (1.0.0)

Function

Add a backup. If data in the current graph is incorrect or faulty, you can start the backup graph to restore the data.

URI

POST /v2/{project_id}/graphs/{graph_id}/backups

Table 6-114 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-115 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-116 Response body parameters

Parameter	Type	Description
job_id	String	ID of the graph backup job. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .
backup_id	String	Graph backup ID.

Status code: 400

Table 6-117 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Add a graph backup.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/backups
{ }
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "job_id" : "ff8080815f9a3c84015f9a438ff70001"
  "backup_id": "0000b54f-2f1d-47a5-94bc-04f0d8a2e2db"
}
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
  "error_code": "GES.0047"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.3.4 Deleting a Backup (1.0.0)

Function

This API is used to delete a backup.

URI

DELETE /v2/{project_id}/graphs/{graph_id}/backups/{backup_id}

Table 6-118 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
backup_id	Yes	String	Graph backup ID
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-119 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

None

Status code: 400

Table 6-120 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.

Parameter	Type	Description
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Delete a backup.

```
DELETE https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/backups/{backupId}
```

Example Response

Status code: 200

Example response for a successful request

```
{ }
```

Status code: 400

Example response for a failed request

```
{
  "error_msg": "The graph backup does not exist or has been deleted.",
  "error_code": "GES.7024"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.3.5 Exporting a Backup (2.3.16)

Function

This API is used to export a GES graph instance backup to OBS.

URI

POST /v2/{project_id}/graphs/{graph_id}/backups/export

Table 6-121 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-122 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-123 Request body parameters

Parameter	Mandatory	Type	Description
backup_id	Yes	String	Backup ID
export_path	Yes	String	Export path in OBS

Response Parameters

Status code: 200

Table 6-124 Response body parameter

Parameter	Type	Description
job_id	String	ID of the asynchronous job

Status code: 400**Table 6-125** Response body parameters

Parameter	Type	Description
error_code	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
error_msg	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.

Example Request

Export a backup.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/backups/export
{
  "backup_id": "{backup_id}",
  "export_path": "{obs_path}"
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "job_id": "ff8080815f9a3c84015f9a438ff70001"
}
```

Status code: 400

Example response for a failed request

```
{
  "errorCode": "GES.7049",
  "errorMessage": "The parameter [backup_id] is not exist."
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.3.6 Importing a Backup (2.3.16)

Function

This API is used to import a GES graph instance backup from OBS.

URI

POST /v2/{project_id}/graphs/{graph_id}/backups/import

Table 6-126 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-127 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-128 Request body parameter

Parameter	Mandatory	Type	Description
import_path	Yes	String	OBS path the backup is to be imported from

Response Parameters

Status code: 200

Table 6-129 Response body parameter

Parameter	Type	Description
job_id	String	ID of the asynchronous job

Status code: 400

Table 6-130 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.

Parameter	Type	Description
error_msg	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.

Example Request

Import a backup.

```
POST https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/backups/import
{
  "import_path" : "{obs_path}"
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "job_id" : "ff8080815f9a3c84015f9a438ff70001"
}
```

Status code: 400

Example response for a failed request

```
{
  "errorCode": "GES.7049",
  "errorMessage": "The parameter [import_path] is not exist."
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.3.7 Obtaining a Download Link for Backup Files (2.4.3)

Function

This API is used to obtain a download link for backup files. Within the validity period of 3,600 seconds, you can directly download the backup file through the URL.

URI

GET /v2/{project_id}/graphs/{graph_id}/backup-files

Table 6-131 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID
backup_id	Yes	String	Backup ID. For how to obtain the backup ID, see "Viewing the Backup List of a Graph".

Request Parameters

Table 6-132 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to use APIs. For how to obtain the token, see Authentication . (The token is the value of X-Subject-Token in the response header.)

Response Parameters

Status code: 200

Table 6-133 Response body parameters

Parameter	Type	Description
bucket	String	Name of the bucket where the file is

Parameter	Type	Description
files	Array of objects	List of backup files. For details, see Table 6-134 .

Table 6-134 files field data structure descriptions

Parameter	Type	Description
name	String	File name
size	Long	File size, in KB
download_link	String	File download link NOTE The way a browser opens a link depends on the type of backup file. For instance, text files can be directly viewed in the browser, while binary files will automatically download to the local host when opened in the browser. To download all backup files, you are advised to use an HTTP library in a programming language such as Java's HttpClient or Python's Requests.
link_expired_time	String	Link expiration time, in yyyy-mm-ddThh:mm:ssZ format.

Status code: 400**Table 6-135** Response body parameters

Parameter	Type	Description
error_code	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
error_msg	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.

Example Request

Obtain the backup download link.

```
GET https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/backup-files?backup_id=22aa0177-  
b20b-4f3a-8556-8147d6658ed8
```

Example Response

Status code: 200

Example response for a successful request

```
{  
  "bucket": "string",  
  "files": [  
    {  
      "name": "43e4feaab48f11e89039fa163ebaa7e4br01.xxx",  
      "size": 2803,  
      "download_link": "https://obs.domainname.com/rdsbucket.username.1/xxxxxx",  
      "link_expired_time": "2018-08-016T10:15:14+0800"  
    }  
  ]  
}
```

Status code: 400

Example response for a failed request

```
{  
  "error_msg": "The graph backup does not exist or has been deleted.",  
  "error_code": "GES.7024"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.4 Metadata Management

6.4.1 Constraints

[Table 6-136](#) and [Table 6-137](#) list the metadata types.

Table 6-136 Metadata property constraints

Data Type	Constraints
char	• Less than '<' • Greater than '>' • Equal to '=' • Not equal to '!=' • In range 'range' • Greater than or equal to '>=' • Less than or equal to '<='
char array	• Less than '<' • Greater than '>' • Equal to '=' • Not equal to '!=' • In range 'range' • Greater than or equal to '>=' • Less than or equal to '<='
float	• Less than '<' • Greater than '>' • Equal to '=' • Not equal to '!=' • In range 'range' • Greater than or equal to '>=' • Less than or equal to '<='
double	• Less than '<' • Greater than '>' • Equal to '=' • Not equal to '!=' • In range 'range' • Greater than or equal to '>=' • Less than or equal to '<='
bool	• Equal to '=' • Not equal to '!='

Data Type	Constraints
long	• Less than '<' • Greater than '>' • Equal to '=' • Not equal to '!=' • In range 'range' • Greater than or equal to '>=' • Less than or equal to '<='
int	• Less than '<' • Greater than '>' • Equal to '=' • Not equal to '!=' • In range 'range' • Greater than or equal to '>=' • Less than or equal to '<='
date	• Less than '<' • Greater than '>' • Equal to '=' • Not equal to '!=' • In range 'range' • Greater than or equal to '>=' • Less than or equal to '<='
enum	• Equal to '=' • Not equal to '!='
string	• Less than '<' • Greater than '>' • Equal to '=' • Not equal to '!=' • In range 'range' • Greater than or equal to '>=' • Less than or equal to '<='

Table 6-137 Property-level constraints

Property Level	Constraints	Description
Single value/ Multiple values	has	This property is contained.

Property Level	Constraints	Description
Single value/ Multiple values	hasNot	This property is not contained.

6.4.2 Querying the Metadata List (1.0.2)

Function

Query the metadata list.

URI

GET /v2/{project_id}/graphs/metadatas

Table 6-138 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Table 6-139 Query parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of resources displayed on a single page. The default value is 10 .
offset	No	Integer	Start position of the request. The default value is 0 .

Request Parameters

Table 6-140 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-141 Response body parameters

Parameter	Type	Description
schema_count	Integer	Number of returned metadata files. This parameter is left blank when the request fails.
schema_list	Array of schema_list objects	List of all metadata files under the current project ID. This parameter is left blank when the request fails.

Table 6-142 schema_list

Parameter	Type	Description
id	String	Metadata ID
name	String	Metadata name
start_time	String	Metadata creation time
last_update_time	String	Last update time of the metadata
description	String	Metadata description
metadata_path	String	Metadata path
status	String	Whether the metadata is available

Status code: 400**Table 6-143** Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Query the metadata list.

```
GET https://{Endpoint}/v2/{project_id}/graphs/metadatas?offset=0&limit=2
```

Example Response**Status code: 200**

Example response for a successful request

```
{
  "schema_count" : 2,
  "schema_list" : [ {
    "start_time" : "2022-01-21T10:13:31",
    "last_update_time" : "2022-01-21T10:13:31",
    "name" : "schema_748e",
    "description" : "xxxxx",
    "id" : "6634c50e-13aa-4395-8088-6b327f7da694",
    "metadata_path" : "devdata/schema_748e.xml",
    "status" : "200"
  }, {
    "start_time" : "2022-04-12T03:15:17",
    "last_update_time" : "2022-11-16T08:18:32",
    "name" : "unionsdk_schema",
    "id" : "6b74069d-3cf3-4cc0-9118-2478e23b87aa",
    "metadata_path" : "devdata/unionsdk/unionsdk_schema.xml",
    "status" : "200"
  } ]
}
```

Status code: 400

Example response for a failed request

```
{
  "error_code" : "GES.7006",
  "error_msg" : "An internal error occurs in the underlying service of the graph engine."
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.4.3 Querying Metadata (1.0.2)

Function

This API is used to query a specific metadata file.

URI

GET /v2/{project_id}/graphs/metadatas/{metadata_id}

Table 6-144 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
metadata_id	Yes	String	Metadata ID

Request Parameters

Table 6-145 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-146 Response body parameters

Parameter	Type	Description
master_key_name	String	Key name.
master_key_id	String	Key ID.
ges_metadata	ges_metadata object	Object for storing metadata message information.

Table 6-147 ges_metadata

Parameter	Type	Description
labels	Array of labels objects	Label data structure set

Table 6-148 labels

Parameter	Type	Description
name	String	Label name
properties	Array of Map<String,String> objects	Property map

Status code: 400**Table 6-149** Response body parameters

Parameter	Type	Description
error_code	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.
error_msg	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.

Example Request

Query metadata of a graph.

```
GET https://{Endpoint}/v2/{project_id}/graphs/metadatas/{metadata_id}
```

Example Response**Status code: 200**

Example response for a successful request

```
{
  "ges_metadata" : {
    "labels" : [ {
      "name" : "friends"
    }, {
      "name" : "movie",
      "properties" : [ {
        "data_type" : "string",
        "name" : "ChineseTitle",
        "cardinality" : "single"
      }, {
        "data_type" : "int",
        "name" : "Year",
        "cardinality" : "single"
      }, {
        "data_type" : "string",
        "name" : "Genres",
        "cardinality" : "set"
      }
    ]
  }, {
    "name" : "user",
    "properties" : [ {
      "data_type" : "string",
      "name" : "ChineseName",
      "cardinality" : "single"
    }, {
      "type_name1" : "F",
      "type_name2" : "M",
      "data_type" : "enum",
```

```

    "name" : "Gender",
    "type_name_count" : "2",
    "cardinality" : "single"
  }, {
    "type_name1" : "Under 18",
    "type_name2" : "18-24",
    "type_name3" : "25-34",
    "type_name4" : "35-44",
    "type_name5" : "45-49",
    "type_name6" : "50-55",
    "type_name7" : "56+",
    "data_type" : "enum",
    "name" : "Age",
    "type_name_count" : "7",
    "cardinality" : "single"
  }, {
    "data_type" : "string",
    "name" : "Occupation",
    "cardinality" : "single"
  }, {
    "data_type" : "char array",
    "name" : "Zip-code",
    "max_data_size" : "12",
    "cardinality" : "single"
  }
}]
}, {
  "name" : "rate",
  "properties" : [ {
    "data_type" : "int",
    "name" : "Score",
    "cardinality" : "single"
  }, {
    "data_type" : "date",
    "name" : "Datetime",
    "cardinality" : "single"
  }
]
}
}]
}
```

Status code: 400

Example response for a failed request

```

{
  "error_msg" : "6b74069d-3cf3-4cc0-9118-2478e23b87a does not exist.",
  "error_code" : "GES.2067"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.4.4 Adding Metadata (2.1.18)

Function

This API is used to add metadata.

URI

POST /v2/{project_id}/graphs/metadatas

Table 6-150 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Request Parameters

Table 6-151 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. For how to obtain the token, see Authentication . (The token is the value of X-Subject-Token in the response header.)

Table 6-152 Request body parameters

Parameter	Mandatory	Type	Description
metadata_path	Yes	String	Path for storing the metadata
name	Yes	String	Metadata name, which contains 1 to 64 characters consisting of only letters, digits, and underscores (_)
description	Yes	String	Metadata description

Parameter	Mandatory	Type	Description
is_overwrite	Yes	Boolean	Whether to overwrite the file.
ges_metadata	Yes	ges_metadata object	Object for storing metadata message information.

Table 6-153 ges_metadata

Parameter	Mandatory	Type	Description
labels	Yes	Array of labels objects	Label list

Table 6-154 labels

Parameter	Mandatory	Type	Description
name	No	String	Name of a label
properties	No	Array of Map<String,String> objects	Label property map

Response Parameters

Status code: 200

Table 6-155 Response body parameters

Parameter	Type	Description
id	String	Metadata ID
name	String	Metadata name

Status code: 400

Table 6-156 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Add metadata. The metadata is stored in **devdata/unionsdk/unionsdk_test.xml**. The metadata does not overwrite existing data, the metadata name is **unionsdk_test**, and the metadata description is **test**.

POST https://{Endpoint}/v2/{project_id}/graphs/metadatas

```
{
  "metadata_path": "devdata/unionsdk/unionsdk_test.xml",
  "is_overwrite": false,
  "name": "unionsdk_test",
  "description": " test",
  "ges_metadata": {
    "labels": [ {
      "name": "friends"
    }, {
      "name": "movie",
      "properties": [ {
        "dataType": "string",
        "name": "ChineseTitle",
        "cardinality": "single"
      }, {
        "dataType": "int",
        "name": "Year",
        "cardinality": "single"
      }, {
        "dataType": "string",
        "name": "Genres",
        "cardinality": "set"
      } ]
    } ]
  }, {
    "name": "user",
    "properties": [ {
      "dataType": "string",
      "name": "ChineseName",
      "cardinality": "single"
    }, {
      "typeName1": "F",
      "typeName2": "M",
      "typeNameCount": "2",
      "dataType": "enum",
      "name": "Gender",
      "cardinality": "single"
    } ], {
```



```
"typeName1" : "Under 18",
"typeName2" : "18-24",
"typeName3" : "25-34",
"typeName4" : "35-44",
"typeNameCount" : "7",
"dataType" : "enum",
"name" : "Age",
"typeName5" : "45-49",
"typeName6" : "50-55",
"cardinality" : "single",
"typeName7" : "56+"
}, {
  "dataType" : "string",
  "name" : "Occupation",
  "cardinality" : "single"
}, {
  "dataType" : "char array",
  "name" : "Zip-code",
  "maxDataSize" : "12",
  "cardinality" : "single"
}]
}, {
  "name" : "rate",
  "properties" : [ {
    "dataType" : "int",
    "name" : "Score",
    "cardinality" : "single"
  }, {
    "dataType" : "date",
    "name" : "Datetime",
    "cardinality" : "single"
  } ]
} ]
} ]
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "id" : "ff8080815f9a3c84015f9a438ff70001",
  "name" : "unionsdk_test"
}
```

Status code: 400

Example response for a failed request

```
{
  "error_msg" : "The metadata file already exists.",
  "error_code" : "GES.2067"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions

Return Value	Description
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.4.5 Deleting Metadata (1.0.2)

Function

This API is used to delete metadata.

URI

DELETE /v2/{project_id}/graphs/metadatas/{metadata_id}

Table 6-157 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
metadata_id	Yes	String	Metadata ID

Request Parameters

Table 6-158 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

None

Status code: 400

Table 6-159 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.

Example Request

Delete metadata.

```
DELETE /v2/{project_id}/graphs/metadatas/{metadata_id}
```

Example Response

Status code: 200

OK

```
{ }
```

Status code: 400

Bad Request

```
{
  "error_code" : "GES.7024",
  "error_msg" : "The metadata is not exist or has been deleted."
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed

Return Value	Description
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.4.6 Importing Metadata from OBS (1.0.0)

Function

This API is used to import metadata from OBS.

URI

POST /v2/{project_id}/graphs/metadata/upload-from-obs

Table 6-160 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Request Parameters

Table 6-161 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token Used to obtain the permission to call APIs. For how to obtain the token, see Authentication . (The token is the value of X-Subject-Token in the response header.)

Table 6-162 Request body parameters

Parameter	Mandatory	Type	Description
metadata_path	Yes	String	Path for storing the metadata
name	Yes	String	Metadata name
description	No	String	Metadata description

Response Parameters

Status code: 200

Table 6-163 Response body parameters

Parameter	Type	Description
id	String	Metadata ID
name	String	Metadata name

Status code: 400

Table 6-164 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Import metadata from OBS. The metadata is stored in **devdata/unionsdk/schema.xml** and the metadata name is **test_schema**.

```
POST https://{Endpoint}/v2/{project_id}/graphs/metadata/upload-from-obs
```

```
{  
  "metadata_path" : "devdata/unionsdk/schema.xml",  
}
```

```
{
  "name" : "test_schema",
  "description" : "xx"
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "id" : "d30d2e94-f2ee-4344-af49-eb27fd002eea",
  "name" : "test_schema"
}
```

Status code: 400

Example response for failed request

```
{
  "error_msg" : "test_schema The name already exists.",
  "error_code" : "GES.2067"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.5 Task Center

6.5.1 Querying the Job Status on the Management Plane (1.0.0)

Function

This interface is used to query the execution status of a task, such as ECS creation, ECS deletion, ECS batch operation, and NIC operation. Asynchronous APIs that are used to create, stop, start, delete, and import graphs will return job IDs after

commands are sent. You can query the job execution status according to the job IDs.

URI

GET /v2/{project_id}/graphs/{graph_id}/jobs/{job_id}/status

Table 6-165 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID
job_id	Yes	String	ID of the asynchronous job

Request Parameters

Table 6-166 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-167 Response body parameters

Parameter	Type	Description
job_id	String	Job ID

Parameter	Type	Description
status	String	Job status. The options are: • pending • running • success • partiallyFailed • failed
job_type	String	Job type.
job_name	String	Task name
related_graph	String	Associated graph name
begin_time	String	Job start time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.
end_time	String	Job end time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.
job_detail	job_detail object	This parameter is returned only when job_name is set to ImportGraph and is used to display graph import details.
fail_reason	String	Job failure cause
job_progress	Double	Job execution progress. It is a reserved field, and not used currently.

Table 6-168 job_detail

Parameter	Type	Description
schema_path	Array of schema_path objects	Path for storing metadata
edgeset_path	Array of edgeset_path objects	Path for storing the edge data set
vertexset_path	Array of vertexset_path objects	Path for storing the vertex data set

Table 6-169 schema_path

Parameter	Type	Description
path	String	OBS storage path

Parameter	Type	Description
log	String	Import log
status	String	OBS file status: • success: Imported successfully. • Failed: Failed to import the file. • partFailed: Partially failed.
cause	String	Import failure cause
total_lines	Long	Total number of imported lines. The value -1 indicates that this field is not returned in the current version.
failed_lines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successful_lines	Long	Number of lines that are successfully imported. The value -1 indicates that this field is not returned in the current version.

Table 6-170 edgeset_path

Parameter	Type	Description
path	String	OBS storage path
log	String	Import log
status	String	OBS file status: • success: Imported successfully. • Failed: Failed to import the file. • partFailed: Partially failed.
cause	String	Import failure cause
total_lines	Long	Total number of imported lines. The value -1 indicates that this field is not returned in the current version.
failed_lines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successful_lines	Long	Number of lines that are successfully imported. The value -1 indicates that this field is not returned in the current version.

Table 6-171 vertexset_path

Parameter	Type	Description
path	String	OBS storage path
log	String	Import log
status	String	OBS file status: • success: Imported successfully. • Failed: Failed to import the file. • partFailed: Partially failed.
cause	String	Import failure cause
total_lines	Long	Total number of imported lines. The value -1 indicates that this field is not returned in the current version.
failed_lines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successful_lines	Long	Number of lines that are successfully imported. The value -1 indicates that this field is not returned in the current version.

Status code: 400**Table 6-172** Response body parameters

Parameter	Type	Description
error_code	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.
error_msg	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.

Example Request

Query the execution status of a job.

```
GET https://{Endpoint}/v2/{project_id}/graphs/{graph_id}/jobs/{job_id}/status
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "job_id" : "ff80808167f09aaa0167f19b35ec0305",
  "status" : "success",
  "job_type" : "GraphManagement",
  "job_name" : "ImportGraph",
  "related_graph" : "GES_UI_AUTO",
  "begin_time" : "2018-11-27T21:39:00",
  "end_time" : "2018-11-27T21:39:56",
  "job_detail" : {
    "vertexset_path" : [ {
      "path" : "ges-ui/auDatas/list_set_vertex.csv",
      "log" : null,
      "cause" : null,
      "status" : "success"
    } ],
    "edgeset_path" : [ {
      "path" : "ges-ui/auDatas/list_set_edge.csv",
      "log" : null,
      "cause" : null,
      "status" : "success"
    } ],
    "schema_path" : [ {
      "path" : "ges-ui/auDatas/list_set_schema.xml",
      "log" : null,
      "cause" : null,
      "status" : "success"
    } ]
  },
  "job_progress" : 0
}
```

Status code: 400

Example response for a failed request

```
{
  "error_msg" : "can not find job, jobId is ff808081646e81d40164c5fb414b2b1a1",
  "error_code" : "GES.8301"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.5.2 Querying Job Details in the Job Center (1.1.8)

Function

This API is used to query asynchronous job details in the job center on the management plane. Asynchronous jobs include creating, closing, starting, deleting, adding, importing, exporting, and upgrading graphs, as well as adding backups.

URI

GET /v2/{project_id}/graphs/jobs

Table 6-173 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation.

Table 6-174 Query parameters

Parameter	Mandatory	Type	Description
end_time	No	String	Job end date. Currently, only the date is supported. The format is <i>yyyy-MM-dd</i> , for example, 2019-03-27.
graph_name	No	String	Associated graph name
limit	No	Integer	Maximum number of resources displayed on a single page. The default value is 10 .
offset	No	Integer	Start position of the request. The default value is 0 .
start_time	No	String	Job start date. Currently, only the date is supported. The format is <i>yyyy-MM-dd</i> , for example, 2019-03-27.

Parameter	Mandatory	Type	Description
status	No	String	The task status. Options: • running • waiting • success • partiallyFailed • failed

Request Parameters

Table 6-175 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-176 Response body parameters

Parameter	Type	Description
job_count	Integer	Total number of jobs
job_list	Array of job_list objects	Task list

Table 6-177 job_list

Parameter	Type	Description
job_id	String	Job ID

Parameter	Type	Description
status	String	Job status. The options are: • pending • running • success • partiallyFailed • failed
job_type	String	Job type.
job_name	String	Task name
related_graph	String	Associated graph name
begin_time	String	Job start time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.
end_time	String	Job end time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.
job_detail	job_detail object	This parameter is returned only when job_name is set to ImportGraph and is used to display graph import details.
fail_reason	String	Job failure cause
job_progress	Double	Job execution progress. It is a reserved field, and not used currently.

Table 6-178 job_detail

Parameter	Type	Description
schema_path	Array of schema_path objects	Path for storing metadata
edgeset_path	Array of edgeset_path objects	Path for storing the edge data set
vertexset_path	Array of vertexset_path objects	Path for storing the vertex data set

Table 6-179 schema_path

Parameter	Type	Description
path	String	OBS storage path

Parameter	Type	Description
log	String	Import log
status	String	OBS file status: • success: Imported successfully. • Failed: Failed to import the file. • partFailed: Partially failed.
cause	String	Import failure cause
total_lines	Long	Total number of imported lines. The value -1 indicates that this field is not returned in the current version.
failed_lines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successful_lines	Long	Number of lines that are successfully imported. The value -1 indicates that this field is not returned in the current version.

Table 6-180 edgeset_path

Parameter	Type	Description
path	String	OBS storage path
log	String	Import log
status	String	OBS file status: • success: Imported successfully. • Failed: Failed to import the file. • partFailed: Partially failed.
cause	String	Import failure cause
total_lines	Long	Total number of imported lines. The value -1 indicates that this field is not returned in the current version.
failed_lines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successful_lines	Long	Number of lines that are successfully imported. The value -1 indicates that this field is not returned in the current version.

Table 6-181 vertexset_path

Parameter	Type	Description
path	String	OBS storage path
log	String	Import log
status	String	OBS file status: • success: Imported successfully. • Failed: Failed to import the file. • partFailed: Partially failed.
cause	String	Import failure cause
total_lines	Long	Total number of imported lines. The value -1 indicates that this field is not returned in the current version.
failed_lines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successful_lines	Long	Number of lines that are successfully imported. The value -1 indicates that this field is not returned in the current version.

Status code: 400**Table 6-182** Response body parameters

Parameter	Type	Description
error_code	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.
error_msg	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.

Example Request

Query asynchronous job details in the job center on the management plane.

GET https://{Endpoint}/v2/{project_id}/graphs/jobs?offset=0&limit=100

Example Response

Status code: 200

Example response for a successful request

```
{
  "job_count": 2,
  "job_list": [ {
    "job_id": "ff80808167bb90340167bc3c7b5b026a",
    "status": "success",
    "job_type": "GraphManagement",
    "job_name": "ImportGraph",
    "related_graph": "test1217",
    "begin_time": "2018-12-17T12:55:40",
    "end_time": "2018-12-17T12:56:32",
    "job_detail": {
      "vertexset_path": null,
      "edgeset_path": [ {
        "path": "hkmovie/edge.csv",
        "log": null,
        "cause": null,
        "status": "success"
      } ],
      "schema_path": [ {
        "path": "hkmovie/schema.xml",
        "log": null,
        "cause": null,
        "status": "success"
      } ]
    },
    "job_progress": 0
  }, {
    "job_id": "ff80808167bb90340167bc5d0b1d0358",
    "status": "success",
    "job_type": "GraphManagement",
    "job_name": "DeleteGraph",
    "related_graph": "test1218",
    "begin_time": "2018-12-17T13:31:14",
    "end_time": "2018-12-17T13:34:48",
    "job_progress": 0
  } ]
}
```

Status code: 400

Example response for a failed request

```
{
  "error_msg": "failed",
  "error_code": "GES.9999"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.

Return Value	Description
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.6 Plugin Management

6.6.1 Querying Scene Analysis Plugin Information

Function

This API is used to query the information about the application analysis capability in a scene, including information about the applications, parameters, and function details.

URI

GET /v2/{project_id}/graphs/scenes

Table 6-183 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Table 6-184 Query parameters

Parameter	Mandatory	Type	Description
scene_name	No	String	Scene name. If only scene_name is specified, all application details in the specified scene will be returned. If only scene_name and application_name are specified, details about the applications requested by application_name will be returned. If scene_name , application_name , and graph_id are left empty, details of all scene applications will be returned.
application_name	No	String	Application name. If only scene_name and application_name are specified, details about the applications requested by application_name will be returned. If scene_name , application_name , and graph_id are left empty, details of all scene applications will be returned.
graph_id	No	String	Graph ID. If only graph_id is set, details about all the subscribed applications of the graph ID will be returned. If scene_name , application_name , and graph_id are all left blank, details of all scene applications will be returned.

Request Parameters

Table 6-185 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 6-186 Response body parameters

Parameter	Type	Description
results	Array of results objects	Scene analysis plugin information

Table 6-187 results

Parameter	Type	Description
scene	String	Scene name
name	String	Application name
params	Array of params objects	Parameter list
description	String	Description of an application in a scene

Table 6-188 params

Parameter	Type	Description
name	String	Parameter name

Parameter	Type	Description
type	String	Parameter type. The value range is ["string","int"]. Currently, only " string " is supported.
default_value	String	The value can be left blank or a specified value. If left blank, you must pass in this parameter when using it.

Status code: 400

Table 6-189 Response body parameters

Parameter	Type	Description
error_code	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.
error_msg	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.

Example Request

Query the information about the application analysis capability in a scene.

```
GET /v2/{project_id}/graphs/scenes?scene_name=xxx&application_name=xxx&graph_id=xxx
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "results": [ {
    "name": "movie_recommendation",
    "description": "Recommend movies that friends are interested in. Graph constraints: (user)-[friends]->(user), (user)-[rates]->(movie)",
    "params": [ {
      "name": "user",
      "default_value": "",
      "type": "string"
    } ],
    "scene": "MovieSocialNetwork_V2"
  }, {
    "name": "friend_recommendation",
```

```
{
  "description": "Recommend people who you may be interested in (considering the relationships between
potential friends and movie preference). Graph constraints: (user) -[friends]-> (user), (user) -[rates]->
(movie)",
  "params": [ {
    "name": "user",
    "default_value": "",
    "type": "string"
  } ],
  "scene": "MovieSocialNetwork_V2"
}
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "The graph does not exist or has been deleted.",
  "error_code": "GES.7000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Management Plane APIs](#).

6.6.2 Subscribing to a Scene Analysis Plugin

Function

This API is used to subscribe to a scene analysis plugin so that you can use the function through the service plane APIs.

** NOTE**

A subscribed plugin cannot be subscribed repeatedly. To update the plugin, cancel the subscription and subscribe to it again after update.

URI

POST /v2/{project_id}/graphs/{graph_id}/scenes/register

Table 6-190 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-191 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-192 Request body parameters

Parameter	Mandatory	Type	Description
scenes	No	Array of scenes objects	Scene you want to subscribe to

Table 6-193 scenes

Parameter	Mandatory	Type	Description
name	No	String	Scene name
applications	No	Array of strings	List of applications you want to subscribe to

Response Parameters

Status code: 200

Table 6-194 Response body parameters

Parameter	Type	Description
result	String	Subscription result

Status code: 400**Table 6-195** Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Subscribe to an application scenario analysis plugin. The scenario name is **Service ticket association analysis**, and the list of applications is **In-depth analysis of dynamic high-dimension relationships**.

```
POST /v2/{project_id}/graphs/{graph_id}/scenes/register
```

```
{
  "scenes": [ {
    "name": "Service ticket association analysis",
    "applications": ["In-depth analysis of dynamic high-dimension relationships"]
  } ]
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
{
  "error_msg": "The graph does not exist or has been deleted.",
}
```



```
"error_code": "GES.7000"  
}
```

Status Code

Return Value	Description
400 Bad Request	Request error
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

6.6.3 Unsubscribing from a Scene Analysis Plugin

Function

This API is used to unsubscribe from a scene analysis plugin. After the subscription is canceled, you cannot use the function through application service plane APIs.

URI

POST /v2/{project_id}/graphs/{graph_id}/scenes/unregister

Table 6-196 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 6-197 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 6-198 Request body parameters

Parameter	Mandatory	Type	Description
scenes	No	Array of scenes objects	List of scenes you want to unsubscribe from

Table 6-199 scenes

Parameter	Mandatory	Type	Description
name	No	String	Scene name
applications	No	Array of strings	List of applications you want to unsubscribe from

Response Parameters

Status code: 200**Table 6-200** Response body parameters

Parameter	Type	Description
success	Array of strings	Scene application that is successfully unsubscribed from
failure	Array of strings	Scene application that fails to be unsubscribed from

Status code: 400

Table 6-201 Response body parameters

Parameter	Type	Description
error_code	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
error_msg	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Example Request

Unsubscribe from a scenario analysis plugin. The scenario name is **Service ticket association analysis**, and the list of applications is **In-depth analysis of dynamic high-dimension relationships**.

```
POST /v2/{project_id}/graphs/{graph_id}/scenes/unregister
{
  "scenes": [ {
    "name": "Service ticket association analysis",
    "applications": ["In-depth analysis of dynamic high-dimension relationships"]
  } ]
}
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "success": [ "testCase" ],
  "failure": [ "xxx" ]
}
```

Status code: 400

Example response for a failed request

```
{
  "error_msg": "The graph does not exist or has been deleted.",
  "error_code": "GES.7000"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error

Return Value	Description
401 Unauthorized	Authorization failed
403 Forbidden	No operation permissions
404 Not Found	No resources found
500 Internal Server Error	Internal server error
503 Service Unavailable	Service unavailable

Error Code

See [Error Codes for Management Plane APIs](#).

7 Service Plane APIs

7.1 Memory Edition

7.1.1 Vertex Operation APIs

7.1.1.1 Querying Vertices That Meet Filter Criteria (1.0.0)

Function

This API is used to query vertices that meet filter criteria.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=query

Table 7-1 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-2 Request body parameters

Parameter	Mandatory	Type	Description
labels	Either labels or vertexFilters is mandatory.	String	Filter criteria of the vertex type
vertexFilters	Either labels or vertexFilters is mandatory.	Object	Filter criteria, in JSONArray format. Vertices are filtered by property.
offset	No	Integer	Start position of the request
limit	No	Integer	Maximum number of resources displayed on a single page. The default value is 10 .
sorts	No	Object	Result sorting property, in JSONArray format

Table 7-3 sorts parameter description

Parameter	Mandatory	Type	Description
key	Either key or propertyName is mandatory.	String	Possible values are id , label , and property . These values indicate that IDs, labels, or properties are sorted.
propertyName	Either key or propertyName is mandatory.	String	Property name
orderValue	No	String	Possible values are incr and decr , which indicate ascending and descending order respectively. The default value is incr .

Table 7-4 vertexFilters parameter structure

Parameter	Mandatory	Type	Description
propertyName	Yes	String	Property name

Parameter	Mandatory	Type	Description
predicate	Yes	String	Predicate. Available values are =, <, >, <=, >=, range , has , hasNot , full_text_match , full_text_prefix , full_text_wildcard , full_text_regexp , full_text_fuzzy , and full_text_combination . NOTE has/hasNot : whether the property exists.
values	No	String	Property value. If predicate is full_text_combination , values cannot be a string. For details, see Example 2 for vertexFilters (full_text_combination) .
type	No	String	Logical operator of the filter criteria. Possible values are and and or . The default value is and .

Response Parameters

Table 7-5 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the vertex query job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Querying Job Status on the Service Plane (1.0.0) .
jobType	Integer	Job type. This parameter is left blank when the request fails.

Example Request

Query vertices that meet filter criteria. The start position of the request is **0**, the maximum number of resources on each page is **2**, the attribute criteria for filtering are **movie** and **user**, and the attribute name for filtering is **Age**.

POST https://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=query

```
{
  "offset":0,
  "limit":2,
  "labels": ["movies",
            "user"],
  "vertexFilters":[{"propertyName":"Age",
                    "predicate":"=",
                    "values":["18-24"]}
]
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example 1 for vertexFilters

```
[
{
  "propertyName":"Gender",
  "predicate":"=",
  "values":["F"]
},
{
  "propertyName":"Age",
  "predicate":"range",
  "values":["18-24","56+"],
  "type":"or"
}
]
```

Example 2 for vertexFilters, with **predicate** set to **full_text_combination**

```
"vertexFilters": [
  {
    "propertyName": "propertyName",
    "predicate": "full_text_combination",
    "values": [
      {
        "propertyName": "movieid",
        "value": "0"
      },
      {
        "propertyName": "title",
        "value": "american"
      }
    ]
  }
]
```

NOTE

When **predicate** is set to **full_text_match**, **full_text_prefix**, **full_text_wildcard**, **full_text_regexp**, **full_text_fuzzy**, or **full_text_combination**, the **vertexFilters** list can contain only one element, that is, multi-layer filters are not allowed. The **labels** parameter cannot be used at the same time. When **predicate** is **full_text_combination**, set the outermost **propertyName** to **propertyName**. In this case, the **values** parameter is not a simple string list. Each element of the **values** list has two members: **propertyName** and **value**. To use the full-text index, you need to call the API for creating a full-text index first.

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "jobId": "03e774f5-29ea-4187-9508-5435f3892ead016886200",
  "jobType": 1
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Bad Request, parameter labels and vertexFilters cannot all be null",
  "errorCode": "GES.8203"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.1.2 Querying Vertex Details (1.0.0)

Function

This API is used to query the vertex information (such as the label and property) based on the vertex ID.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/detail?
vertexIds={vertex_ids}

Table 7-6 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
vertexIds	Yes	String	IDs of the vertices to be queried. When multiple IDs are specified by vertexIds , separate the IDs with commas (,) in the URL.

Request Parameters

None

Response Parameters

Table 7-7 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	Object	Query results.

Example Request

Query node information by node ID and return node details.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/detail?vertexIds=Ray
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "data": {
    "vertices": [
      {
        "id": "Ray",
        "label": "user",
        "properties": {
          "Occupation": [
            "college/grad student"
          ],
          "Name": [
            "Lei"
          ],
          "Zip-code": [
            "90241"
          ],
          "Gender": [
            "M"
          ],
          "Age": [
            "18-24"
          ]
        }
      }
    ]
  }
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.1.3 Adding a Vertex (1.0.6)

Function

This API is used to add a vertex.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices

Table 7-8 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-9 Request body parameter descriptions

Parameter	Mandatory	Type	Description
vertex	Yes	String	Vertex name
label	Yes	String	Label of a vertex. If no label exists, set it to __DEFAULT__ .
properties	No	Json	Value of each property

Response Parameters

Table 7-10 Parameter description

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Request

Add a vertex named **Lily** and set the vertex label to **user**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices
{
  "vertex": "Lily",
  "label": "user",
  "properties": {
    "Age": [
      "under 18"
    ],
    "Gender": [
      "F"
    ],
    "Occupation": [
      "artist"
    ],
    "Zip-code": [
      "98133"
    ]
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "vertex [Lily] already exists",
  "errorCode": "GES.8000"
}
```

Status Code

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.1.4 Deleting a Vertex (1.0.6)

Function

This API is used to delete a vertex.

URI

DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}

Table 7-11 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
vertex_id	Yes	String	Vertex ID

Response Parameters

Table 7-12 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Request

Delete a vertex.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/Lily
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "vertex [Lily] does not exist",
  "errorCode": "GES.8000"
}
```

Status Code

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.1.5 Updating Vertex Properties (1.1.6)

Function

This API is used to update vertex property values. The actions include **add**, **update**, and **del**.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/properties/
action?action_id={actionId}

Table 7-13 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
vertex_id	Yes	String	Vertex ID

Parameter	Mandatory	Type	Description
action_id	Yes	String	Operator. The options are: • UPDATE: Update the value of a property. • ADD: Add the value to a property. When the property's cardinality is single, the operation is the same as that of UPDATE. When cardinality is list or set, the operator adds a value to a set. • DEL: Delete a property value.

Request Parameters

Table 7-14 Request body parameter descriptions

Parameter	Mandatory	Type	Description
properties	Yes	Object	Value of each property
label	No	String	Name of a label

Response Parameters

Table 7-15 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Request

Update the property value of a vertex. The value of the **Age** property is **under 18**, and the value of the **Gender** property is **F**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/Lily/properties/action?
action_id={actionId}
{
  "properties":{
    "Age":[
      "under 18"
    ],
    "Gender":[
      "F"
    ]
  }
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "vertex [Lily] does not exist",
  "errorCode": "GES.8220"
}
```

Status Code

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.1.6 Batch Querying Vertex Data (1.1.9)

Function

This API is used to query the vertex data (such as the labels and properties) in batches based on the vertex IDs.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?
action_id=batch-query

Table 7-16 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-17 Request body parameter descriptions

Parameter	Mandatory	Type	Description
vertices	Yes	List	IDs of the vertices to query

Response Parameters

Table 7-18 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
data	Object	The data field is contained when the query is successful, and the data field contains the vertices query result.

Example Request

Query nodes in batches by node ID. The vertex IDs to be queried are **27003509_CCTV Station Building** and **39636392_Badaling Great Wall**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-query
{
  "vertices":
  ["27003509_CCTV Station Building",
  "39636392_Badaling Great Wall"]
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "data": {
    "vertices": [
      {
        "id": "27003509_CCTV Station Building",
        "label": "tag",
        "properties": {
          "popularity": [
            0
          ],

```

```
      "name": [
        "CCTV Station Building"
      ],
      "alias": [
        "CCTV, Guanghai Road Office",
        "CCTV Headquarters",
        "Giant Underpants",
        "CCTV Headquarters Building"
      ]
    },
    {
      "id": "39636392_Badaling Great Wall",
      "label": "tag",
      "properties": {
        "popularity": [
          0
        ],
        "name": [
          "Badaling Great Wall"
        ],
        "alias": [
          "Great Wall"
        ]
      }
    }
  ],
},
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": " Bad Request, parameter vertices cannot be null",
  "errorCode": "GES.8214"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Codes for Service Plane APIs](#).

7.1.1.7 Batch Adding Vertices (2.1.16)

Function

This API is used to add vertices in batches.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?
action_id=batch-add

Table 7-19 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-20 Request body parameter descriptions

Parameter	Mandatory	Type	Description
vertices	Yes	List	Vertices to add. You can add a maximum of 10,000 vertices at a time. For details about this array, see Table 7-21 .
overrideExists	No	Boolean	Whether to overwrite the existing vertices in the vertices parameter. The default value is false . - If this parameter is set to false, existing vertices are ignored. - If this parameter is set to true, the existing vertices in the vertices parameter are overwritten.

Table 7-21 vertices parameter descriptions

Parameter	Mandatory	Type	Description
vertex	Yes	String	Vertex ID

Parameter	Mandatory	Type	Description
label	Yes	String	Vertex label
properties	No	Json	Value of each property

Response Parameters

Table 7-22 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
result	String	Response result. The value is success if the query is successful; this parameter is not returned if the query fails.

Example Request

Add vertices in batches. The names of the vertices to add are **150** and **6**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-add
{
  "vertices": [
    {
      "vertex": "150",
      "label": "movie",
      "properties": {
        "movieid": [
          "150"
        ],
        "title": [
          "testmoive"
        ],
        "genres": [
          "Comedy"
        ]
      }
    }
  ],
}
```

```
    "vertex": "6",
    "label": "movie",
    "properties": {
      "movieid": [
        "6"
      ],
      "title": [
        "testmoive_exist_id"
      ],
      "genres": [
        "Comedy"
      ]
    }
  ],
  "overrideExists": true
}
```

 NOTE

- **SERVER_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- In the example, if vertex **6** already exists in the graph, properties of vertex **6** are overwritten.

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Bad Request, parameter vertices cannot be null",
  "errorCode": "GES.8214"
}
```

Status Codes

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.1.8 Batch Deleting Vertices (2.1.9)

Function

This API is used to delete vertices in batches based on the vertex IDs.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?
action_id=batch-delete

Table 7-23 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-24 Request body parameter descriptions

Parameter	Mandatory	Type	Description
vertices	Yes	List	IDs of the vertices to delete
ignoreError (2.2.28)	No	Boolean	Whether to ignore errors, for example, the vertices you want to delete does not exist. The value can be true or false . The default value is false , indicating that errors will not be ignored. Note that JSON format errors will not be ignored.

Response Parameters

Table 7-25 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Request

Delete vertices in batches by vertex ID. The IDs of the vertices to delete are **Vivian** and **46**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-delete
{
  "vertices": [
    "Vivian",
    "46"
  ],
  "ignoreError": true
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
```

```
"errorMessage": " Bad Request, parameter vertices cannot be null",  
"errorCode": "GES.8214"  
}
```

Status Codes

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Codes for Service Plane APIs](#).

7.1.1.9 Batch Updating Vertex Properties (2.1.10)

Function

This API is used to update vertex properties in batches.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/properties/action?
action_id={actionId}

Table 7-26 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Parameter	Mandatory	Type	Description
action_id	Yes	String	Operator. The options are: • batch-update: Update the value of a property. • batch-add: Add the value to a property. When the property's cardinality is single, the operation is the same as that of batch-update. When cardinality is list or set, the operator adds a value to a set. • batch-del: Delete a property value.

Request Parameters

Table 7-27 Request body parameter descriptions

Parameter	Mandatory	Type	Description
vertices	Yes	Json	Vertices to update. For details about this array, see Table 7-28 .
ignoreError	No	Boolean	Whether to ignore the update error of specific vertices. The default value is false . • The value false indicates that if an error that causes the update failure is detected, for example, the vertex to be updated does not exist, an error is reported and no vertex will be updated. • The value true indicates that similar errors will be ignored and other vertex properties without errors will be updated.

Table 7-28 vertices parameter descriptions

Parameter	Mandatory	Type	Description
vertex	Yes	String	Vertex ID

Parameter	Mandatory	Type	Description
label	No	String	Vertex label
properties	Yes	Json	Value of each property to be updated

Response Parameters

Table 7-29 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Request

Update vertex properties in batches. The names of the vertices to update are **150** and **6**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/properties/action?
action_id=batch-update
{
  "vertices": [
    {
      "vertex": "150",
      "label": "movie",
      "properties": {
        "movieid": [
          "150"
        ],
        "title": [
          "testmoive"
        ],
        "genres": [
          "Comedy"
        ]
      }
    }
  ],
}
```

```
{
  "vertex": "6",
  "properties": {
    "title": [
      "testmoive_exist_id"
    ],
    "genres": [
      "Comedy"
    ]
  }
},
"ignoreError": true
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "vertex [150] does not exist",
  "errorCode": "GES.8220"
}
```

Status Codes

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.1.1.10 Adding a Vertex Label (1.1.6)

Function

This API is used to add a vertex label.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/labels

Table 7-30 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
vertex_id	Yes	String	Vertex ID

Request Parameters

Table 7-31 Request body parameter

Parameter	Mandatory	Type	Description
label	Yes	String	Vertex label

Response Parameters

Table 7-32 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.

Example Request

Add a vertex label named **user**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/labels
{
  "label": "user"
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{ }
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Same label [user] already exists",
  "errorCode": "GES.8213"
}
```

Status Code

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.

Return Value	Description
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.1.11 Deleting a Vertex Label (1.1.6)

Function

This API is used to delete a vertex label.

URI

DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/{vertex_id}/labels/{label_name}

Table 7-33 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
vertex_id	Yes	String	Vertex ID
label_name	Yes	String	Vertex label

Response Parameters

Table 7-34 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Request

Delete a vertex label.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/46/labels/movie
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Vertex [46] does not have label [movie]",
  "errorCode": "GES.8182"
}
```

Status Code

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.

Return Value	Description
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.1.12 Exporting Filtered Vertices (2.2.7)

Function

This API is used to export vertex sets that meet the filter criteria.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=export

Table 7-35 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-36 Request body parameter descriptions

Parameter	Mandatory	Type	Description
labels	Either labels or vertexFilters is mandatory.	String	Filter criteria of the vertex type
vertexFilters	Either labels or vertexFilters is mandatory.	Json	Filter criteria, in JSONArray format. Vertices are filtered by property. For details, see Table 7-4 .
exportPath	Yes	String	Export path

Parameter	Mandatory	Type	Description
fileName	No	String	Name of the exported file
obsParameters	Yes	String	OBS authentication parameters. For details, see Table 7-38 .

Table 7-37 vertexFilters parameters

Parameter	Mandatory	Type	Description
propertyName	Yes	String	Property name
predicate	Yes	String	Predicate. The options are =, <, >, <=, >=, range , has , full_text_match , full_text_prefix , full_text_wildcard , full_text_regexp , full_text_fuzzy , full_text_combination , and hasNot .
values	No	String	Property value. If predicate is full_text_combination , the value cannot be a string.
type	No	String	Logical operator of the filter criteria. The options are and and or . The default value is and .

Table 7-38 obsParameters parameter descriptions

Parameter	Mandatory	Type	Description
accessKey	Yes	String	AK
secretKey	Yes	String	SK

Response Parameters

Table 7-39 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the edge query job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	Integer	Type of the asynchronous job if the request is executed asynchronously.

Example Request

Export the vertex sets that meet the filter criteria (only the asynchronous mode is supported). The export path is **demo_movie/**, and the export file name is **export_movie_and_user.csv**.

```
POST https://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=export
{
  "labels": [
    "movies",
    "user"
  ],
  "vertexFilters": [
    {
      "propertyName": "Age",
      "predicate": "=",
      "values": [
        "18-24"
      ]
    }
  ],
  "exportPath": "demo_movie/",
  "fileName": "export_movie_and_user.csv",
  "obsParameters": {
    "accessKey": "XXXX",
    "secretKey": "XXXX"
  }
}
```

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "jobId": "03e774f5-29ea-4187-9508-5435f3892ead016886200",
  "jobType": 1
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Bad Request, parameter labels and vertexFilters cannot all be null",
  "errorCode": "GES.8203"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.1.13 Deleting Filtered Vertices (2.2.7)

Function

This API is used to delete the vertex sets that meet the filter criteria.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=delete

Table 7-40 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-41 Request body parameter descriptions

Parameter	Mandatory	Type	Description
labels	Either labels or vertexFilters is mandatory.	String	Filter criteria of the vertex type
vertexFilters	Either labels or vertexFilters is mandatory.	Object	Filter criteria, in JSONArray format. Vertices are filtered by property. For details, see Table 7-42 .
begin	No	Integer	Start of a filter range. The value ranges from 0 to uint32_MAX. The default value is 0 .
end	No	Integer	End of a filter range. The value ranges from 0 to uint32_MAX. The default value is uint32_MAX . Note: If begin and end are not set, the vertices in the entire graph are filtered by default. You can set both parameters based on site requirements (for example, 10 million).
limit	No	Integer	Maximum number of vertices that can be deleted. By default, all vertices that meet the conditions are deleted.

Table 7-42 vertexFilters parameters

Parameter	Mandatory	Type	Description
propertyName	Either propertyName or degree is mandatory.	String	Property name
predicate	Yes	String	Predicate. The options are =, <, >, <=, >=, range , has , full_text_match , full_text_prefix , full_text_wildcard , full_text_regexp , full_text_fuzzy , full_text_combination , and hasNot .
values	No	String	Value of the property or the degree of a vertex. If predicate is full_text_combination , the value cannot be a string.
type	No	String	Logical operator of the filter criteria. The options are and and or . The default value is and .
degree (2.4.0)	Either propertyName or degree is mandatory.	String	Degree of a vertex. The options are in , out , and both . The predicate can be =, <, >, <=, >=, or !=.

Response Parameters

Table 7-43 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.

Parameter	Type	Description
jobId	String	ID of the vertex query job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Querying Job Status on the Service Plane (1.0.0) .
jobType	Integer	Job type. This parameter is left blank if the request fails.

Example Request

Delete the vertex sets that meet the filter criteria (only the asynchronous mode is supported). The vertex type filter criteria are **movies** and **user**, and the property name is **Age**.

```
POST https://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=delete
{
  "labels": [
    "movies",
    "user"
  ],
  "vertexFilters": [
    {
      "propertyName": "Age",
      "predicate": "=",
      "values": [
        "18-24"
      ]
    }
  ]
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "03e774f5-29ea-4187-9508-5435f3892ead016886200",
  "jobType": 1
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Bad Request, parameter labels and vertexFilters cannot all be null",
  "errorCode": "GES.8203"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.2 Edge Operation APIs

7.1.2.1 Querying Edges That Meet Filter Criteria (1.0.0)

Function

This API is used to query edges that meet filter criteria.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=query

Table 7-44 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-45 Request body parameter descriptions

Parameter	Mandatory	Type	Description
labels	Either labels or edgeFilters is mandatory.	String	Filter criteria of the relationship type
edgeFilters	Either labels or edgeFilters is mandatory.	String	Filter criteria, in JSONArray format. Vertices are filtered by property.
offset	No	Integer	Start position of the request
limit	No	Integer	Expected number of edges returned by a query
sorts	No	Object	Result sorting property. It is in JSONArray format.

Table 7-46 sorts parameter descriptions

Parameter	Mandatory	Type	Description
key	Either Key or propertyName is mandatory.	String	The options are label and property , which indicate that labels are sorted and properties are sorted, respectively.
propertyName	Either Key or propertyName is mandatory.	String	Property name
orderValue	No	String	The options are incr and decr , which indicate ascending and descending order, respectively. The default value is incr .

Table 7-47 edgeFilters parameter structure

Parameter	Mandatory	Type	Description
propertyName	Yes	String	Property name

Parameter	Mandatory	Type	Description
predicate	Yes	String	Logical relationship. The options are =, <, >, <=, >=, range , has , full_text_match , full_text_prefix , full_text_wildcard , full_text_regexp , full_text_fuzzy , full_text_combination , and hasNot . NOTE has/hasNot : whether the property exists.
values	No	String	Property value. When predicate is full_text_combination , values cannot be a string. For details, see the following note.
type	No	String	Logical operator of the filter criteria. The options are and and or . The default value is and .

 NOTE

When **predicate** is set to **full_text_match**, **full_text_prefix**, **full_text_wildcard**, **full_text_regexp**, **full_text_fuzzy**, or **full_text_combination**, the **edgeFilters** list can contain only one element, that is, multi-layer filters are not allowed. The **labels** parameter cannot be used at the same time. When **predicate** is **full_text_combination**, set the outermost **propertyName** to **propertyName**. In this case, the **values** parameter is not a simple string list. Each element of the **values** list has two members: **propertyName** and **value**. To use the full-text index, you need to call the API for creating a full-text index first.

Response Parameters

Table 7-48 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.

Parameter	Type	Description
jobId	String	ID of the edge query job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Example Request

Query edges that meet filter criteria. The start position of the request is **0**, the number of edges to be returned is **20**, and the filter criterion of the relationship type is **rate**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=query
{
  "offset":0,
  "limit":20,
  "labels":[
    "rate"
  ],
  "edgeFilters":[
    {
      "propertyName":"Score",
      "predicate":">=",
      "values":[
        "2"
      ]
    },
    {
      "propertyName":"Datetime",
      "predicate":"range",
      "values":[
        "1998-12-27 01:00:00",
        "2000-12-31 00:12:38"
      ],
      "type":"or"
    }
  ]
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "f9987cab-64d3-4b3d-ac43-e91ae0c21bef168127124",
  "jobType": 0
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Bad Request, parameter labels and edgeFilters cannot all be null",
  "errorCode": "GES.8103"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.2.2 Querying Edge Details (1.0.0)

Function

This API is used to query the detailed information about an edge based on the source vertex, target vertex, and index of the edge. Information about edges and properties is returned.

URI

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?
source={sourceVertex}&target={targetVertex}&index={index}
```

Table 7-49 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge

Parameter	Mandatory	Type	Description
index	No	Integer	Edge index. If this parameter is not set, all edges between the source and target vertices will be queried.

Request Parameters

None

Response Parameters

Table 7-50 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	Object	Query results. If the query is successful, the query result will be returned. If the query fails, this parameter will be left blank.

Table 7-51 data parameter descriptions

Parameter	Mandatory	Type	Description
edges	Yes	List	Edge result set. If no edge is found, this parameter will be left blank.

Example Request

Query details about an edge.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?
source=Ray&target=Rocky&index=6
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "data": {
    "edges": [
      {
        "index": "6",
        "source": "Ray",
        "label": "rate",
        "properties": {
          "Score": [
            3
          ],
          "Datetime": [
            "2000-11-22 19:23:05"
          ]
        },
        "target": "Rocky"
      }
    ]
  }
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.2.3 Adding an Edge (2.1.2)

Function

This API is used to add an edge.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges

Table 7-52 URI parameters

Parameter	Mandato ry	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-53 Request body parameter descriptions

Parameter	Mandato ry	Type	Description
source	Yes	String	Source vertex name
target	Yes	String	Target vertex name
label	No	String	Label of an edge. If no label exists, set it to __DEFAULT__ .
properties	No	Object	Value of each property
parallel Edge	No	parallelEdge Object	How to process repetitive edges

Table 7-54 parallelEdge parameter descriptions

Parameter	Mandatory	Type	Description
action	No	String	Processing mode of repetitive edges. The value can be allow, ignore, or override. The default value is allow. • allow indicates that repetitive edges are allowed. • ignore indicates that subsequent repetitive edges are ignored. • override indicates that the previous repetitive edges are overwritten.
ignoreLabel	No	Boolean	Whether to ignore labels on repetitive edges. The value can be true or false. The default value is true. • true: indicates that the repetitive edge definition does not contain the label. That is, <Source vertex, Target vertex> indicates an edge. • false: indicates that the repetitive edge definition contains the label. That is, <Source vertex, Target vertex, Label> indicates an edge.
targetProperties	No	targetProperties Array	List of properties used to determine repetitive edges. If this parameter is specified, ignoreLabel is set to false to define repetitive edges that contain properties in the list specified by this parameter. (This parameter is mandatory only when action is set to override.) NOTE • The current version supports only overwriting by property and does not support ignoring by property. In addition, only one property takes effect for each label. • Properties of the non-single type are considered unequal.

Table 7-55 targetProperties parameter descriptions

Parameter	Mandatory	Type	Description

label	No	String	Labels whose repetitive edges need to be determined by property. Note: label and properties are both optional. If label is set, properties is also required.
properties	No	Array	List of properties whose repetitive edges need to be determined by property. Currently, only one property is supported. If multiple properties are entered, the first property is used.

Response Parameters

Table 7-56 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .
data	Object	Query results. If the query is successful, the query results are returned. If the query fails, this parameter is left blank.

Example Request

Add an edge. The source name is **Lily**, the target name is **Rocky**, and the label name of the edge is **rate**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges
{
  "source": "Lily",
  "target": "Rocky",
  "label": "rate",
  "properties": {
    "Score": [
      5
    ],
    "Datetime": [
```

```
    "2018-01-01 20:30:05"
  ]
},
"parallelEdge": {
  "action": "override",
  "ignoreLabel": true,
  "targetProperties": [
    {
      "label": "rate",
      "properties": [
        "Datetime"
      ]
    },
    {
      "label": "superclassOf",
      "properties": [
        "popularity"
      ]
    }
  ]
}
]
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success",
  "data": { "index": "0" }
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Edge source vertex [Lily] does not exist",
  "errorCode": "GES.8000"
}
```

Status Code

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.

Return Value	Description
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.2.4 Deleting an Edge (1.0.6)

Function

This API is used to delete an edge based on the specified property value or index.

URI

DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/edges?
source={sourceVertex}&target={targetVertex}&index={index}&label={label}&proper
ty={name}&value={value}

Table 7-57 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
source	Yes	String	Source vertex name
target	Yes	String	Target vertex name
index	No	Integer	Edge index - If property has been set, ignore this parameter. - If property is not set, the edge is deleted based on index. - If neither property nor index is set, all edges between source and target are deleted.

Parameter	Mandatory	Type	Description
label	No	String	Indicates the label of an edge, which can accelerate the search of property values. This parameter must be used together with property .
property	No	String	Property name of the edge to be deleted. This parameter must be used together with value .
value	No	String	Indicates the property value of the edge to be deleted. This parameter must be used together with property .

Request Parameters

None

Response Parameters

Table 7-58 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Request

Delete an edge. The source vertex name of the edge to be deleted is **Vivian**, the target vertex name is **Lethal Weapon**, the edge index value is **0**, the edge label value is **rate**, the property name is **Score**, and the property value is **5**.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges?
source=Vivian&target=Lethal Weapon&index=0&label=rate&property=Score&value=5
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Edge source vertex [Vivian] does not exist",
  "errorCode": "GES.8000"
}
```

Status Code

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.2.5 Updating Edge Properties (1.1.6)

Function

This API is used to update edge property values. The actions include **add**, **update**, and **del**.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?
action_id={actionId}&source={sourceVertex}&target={targetVertex}&index={index}

Table 7-59 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
action_id	Yes	String	Operator. The options are: • update: Update a property value. • add: Add a property value. When the property's cardinality is single, the operation is the same as that of UPDATE. When cardinality is list or set, the operator adds a value to a set. • del: Delete a property value.
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge
index	No	Integer	Edge index. If this parameter is not set, properties of the first edge between the vertices will be modified.

Request Parameters

Table 7-60 Request body parameter descriptions

Parameter	Mandatory	Type	Description
properties	Yes	Object	Value of each property

Parameter	Mandatory	Type	Description
targetProperties	No	Array	Properties used to determine duplicate edges. • If this parameter is not left blank, other properties of duplicate edges (with the same source vertex and target vertex) that has the same property value as the input property value will be overwritten. If there are multiple specified properties, the properties of the first edge that is matched based on the property input sequence are modified. • If this parameter is left blank or no property is specified for the input edge, the first edge that meets the criteria is updated. For details about the property elements, see Table 7-61.

Table 7-61 targetProperty parameter descriptions

Parameter	Mandatory	Type	Description
label	Yes	String	Label name. The label of duplicate edges is determined by the property.
properties	Yes	Array	Value of each property. The property list of duplicate edges is determined by the property. Currently, only a single property is supported. If multiple properties are entered, the first property is used.

Example Request

Update the property value of an edge. The value of property **Rating** is **7**, the value of property **Datetime** is **2020-12-27 23:44:41**, and the label name is **rate**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?
action_id=update&source=Lily&target=Tom&index=1
```

```
{
  "properties": {
    "Rating": ["7"],
    "Datetime": ["2020-12-27 23:44:41"]
  },
  "targetProperties": [
    {
      "label": "rate",
      "properties": [
        "Rating"
      ]
    }
  ]
}
```

```
}  
]  
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Response Parameters

Table 7-62 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200  
{  
  "result": "success"  
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400  
{  
  "errorMessage": "Edge [Lily-Tom-1] does not exist",  
  "errorCode": "GES.8221"  
}
```

Status Code

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.2.6 Batch Querying Edge Data (1.1.6)

Function

This API is used to query the detailed information about edges in batches based on the source vertices, target vertices, and indexes of the edges. Information about edges and properties is returned.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-query

Table 7-63 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-64 Request body parameter

Parameter	Mandatory	Type	Description
edges	Yes	Object	Edge array to be queried

Table 7-65 edges parameter description

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge
index	No	String	Edge index

Response Parameters

Table 7-66 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	String	The data field is contained when the query is successful, and the data field contains the edges query result.

Example Request

Query details about edges in batches based on the source vertex, target vertex, and index. The source nodes of the edges to be queried are **39631050_Landscape** and **27803870_Landmark building**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-query
{
  "edges": [{
    "source": "39631050_Landscape",
    "target": "27803870_Landmark building"
  }, {
    "index": "0",
    "source": "27803870_Landmark building",
    "target": "27661363_Beijing Jiuhua Hot Spring"
  }]
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
{
  "data": {
    "edges": [
      {
        "index": "24",
        "source": "39631050_Landscape",
        "label": "superclassOf",
        "properties": {
          "popularity": [
            0
          ]
        }
      },
      {
        "index": "0",
        "source": "27803870_Landmark building",
        "label": "superclassOf",
        "properties": {
          "popularity": [
            0
          ]
        }
      },
      {
        "index": "0",
        "source": "27803870_Landmark building",
        "label": "superclassOf",
        "properties": {
          "popularity": [
            0
          ]
        }
      },
      {
        "index": "0",
        "source": "27803870_Landmark building",
        "label": "superclassOf",
        "properties": {
          "popularity": [
            0
          ]
        }
      }
    ]
  }
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter does not contain 'source'",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.2.7 Batch Adding Edges (2.1.16)

Function

This API is used to add edges in batches.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add

Table 7-67 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-68 Request body parameter descriptions

Parameter	Mandatory	Type	Description
edges	Yes	edges Object	Edge array to be added
parallelEdge	No	parallelEdge Object	How to process repetitive edges
createNotExists	No	Boolean	Whether to add source or target vertices that do not exist in the edges parameter before adding edges. The default value is false , which does not affect the original functions and semantics. If this parameter is set to true , source or target vertices that do not exist in the edges parameter are added prior to the edges.

Table 7-69 edges parameter descriptions

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge
label	Yes	String	Edge label
properties	No	Object	Value of each property

Table 7-70 parallelEdge parameter descriptions

Parameter	Mandatory	Type	Description
action	No	String	Processing mode of repetitive edges. The value can be allow, ignore, or override. The default value is allow. • allow indicates that repetitive edges are allowed. • ignore indicates that subsequent repetitive edges are ignored. • override indicates that the previous repetitive edges are overwritten.
ignoreLabel	No	Boolean	Whether to ignore labels on repetitive edges. The value can be true or false. The default value is true. • true: Indicates that the repetitive edge definition does not contain the label. That is, the <source vertex, target vertex> indicates an edge, excluding the label information. • false: indicates that the repetitive edge definition contains the label. That is, <Source vertex, Target vertex, Label> indicates an edge.
targetProperties	No	targetProperties Array	List of properties used to determine repetitive edges. If this parameter is specified, ignoreLabel is set to false to define repetitive edges that contain properties in the list specified by this parameter. (This parameter is mandatory only when action is set to override.) NOTE • The current version supports only overwriting by property and does not support ignoring by property. In addition, only one property takes effect for each label. • Properties of the non-single type are considered unequal.

Table 7-71 targetProperties parameter descriptions

Parameter	Mandatory	Type	Description
label	Yes	String	Labels whose repetitive edges need to be determined by property

properties	Yes	Array	List of properties whose repetitive edges need to be determined by property. Currently, only one property is supported. If multiple properties are entered, the first property is used.

Response Parameters

Table 7-72 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Request

Add edges in batches. The source vertex is **46**, the target vertices are **39** and **38**, and the edge label is **rate**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add
{
  "parallelEdge": {
    "action": "override",
    "ignoreLabel": false,
    "targetProperties": [
      {
        "label": "rate",
        "properties": [
          "Datetime"
        ]
      },
      {
        "label": "superclassOf",
        "properties": [
          "popularity"
        ]
      }
    ]
  },
  "createNotExists": true,
  "edges": [
```

```
{
  {
    "source": "46",
    "target": "39",
    "label": "rate",
    "properties": {
      "Rating": [
        5
      ],
      "Datetime": [
        "2018-01-01 20:30:05"
      ]
    }
  },
  {
    "source": "46",
    "target": "39",
    "label": "rate",
    "properties": {
      "Rating": [
        4
      ],
      "Datetime": [
        "2018-01-01 20:30:05"
      ]
    }
  }
}
```

NOTE

- **SERVER_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- In the example, if vertices **46** and **39** are not in the original graph, create vertices **46** and **39**, retain the default value of each label, and add an edge.

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success",
  "data": {
    "edges": [
      {
        "index": "7",
        "source": "46",
        "target": "39"
      },
      {
        "index": "0",
        "source": "46",
        "target": "38"
      }
    ]
  }
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Edge source vertex [Lily] does not exist",
}
```

```
"errorCode": "GES.8000"  
}
```

Status Codes

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.2.8 Batch Deleting Edges (2.1.9)

Function

This API is used to delete edges in batches based on the source vertices, target vertices, and indexes of the edges.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-delete

Table 7-73 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-74 Request body parameter descriptions

Parameter	Mandatory	Type	Description
edges	Yes	Object	Edge array to be deleted
executionMode (2.2.14)	No	String	sync indicates the synchronous mode, and async indicates the asynchronous mode. The default value is sync .
ignoreError (2.2.17)	No	Boolean	Whether to ignore errors, for example, the edge to delete does not exist. The default value is false , indicating that errors will not be ignored. When set to true , errors in JSON format are not ignored.

Table 7-75 edges parameter descriptions

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge
index	No	String	Edge index
label (2.2.17)	No	String	Label of an edge. If the index parameter is set, this parameter is ignored. If the index parameter is not set, an edge that meets the source , target , and label conditions is deleted. If the specified label value does not exist in the schema or the edge with the same label does not exist, no edge will be deleted.

Response Parameters

- Synchronous call

Table 7-76 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

- Asynchronous call (2.2.14)

Table 7-77 Response body parameter descriptions

Parameter	Mandatory	Type	Description
errorMessage	No	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	No	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
jobId	No	String	ID of the deletion job. This parameter is left blank when the request fails. This ID can be used as a request parameter to obtain the deletion result through the API for querying the job status.
jobType	No	Integer	Job type. This parameter is left blank when the request fails.

Example Request

Delete edges in batches. The source vertices of the edges are **39631050_Landscape** and **27803870_Landmark building**, and the target vertices of the edges are **27803870_Landmark building** and **27661363_Villa hot spring**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-delete
{
  "edges": [
    {
      "source": "39631050_Landscape",
      "target": "27803870_Landmark building"
    },
    {
      "index": "0",
      "source": "27803870_Landmark building",
      "target": "27661363_Villa hot spring"
    }
  ],
  "ignoreError": true
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

- Synchronization mode

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Parameter does not contain 'source'",
  "errorCode": "GES.8000"
}
```

- Asynchronous mode

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "jobId": "500dea8f-9651-41fe-8299-c20f13a032ea",
  "jobType": 3
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.2.9 Batch Updating Edge Properties (2.1.10)

Function

This API is used to update edge properties in batches.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?
action_id={actionId}

Table 7-78 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Parameter	Mandatory	Type	Description
action_id	Yes	String	Operator. The options are: • batch-update: Update the value of a property. • batch-add: Add the value to a property. When the property's cardinality is single, the operation is the same as that of batch-update. When cardinality is list or set, the operator adds a value to a set. • batch-del: Delete a property value.

Request Parameters

Table 7-79 Request body parameter descriptions

Parameter	Mandatory	Type	Description
edges	Yes	Json	Edge array to be updated
ignoreError	No	Boolean	Whether to ignore the update error of specific edges. The default value is false , indicating that an error that causes the update failure must be detected. For example, if the edge to be updated does not exist, an error is reported and no edge is updated. If set to true , similar errors are ignored and other edge properties without errors are updated.

Table 7-80 edges parameter descriptions

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge
index	No	String	Edge index. If this parameter is not set, the first edge between vertices is updated.
properties	Yes	Object	Value of each property

Response Parameters

Table 7-81 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Request

Update edge properties in batches. The source vertex of the edge is **46**, and the target vertices of the edge are **39** and **38**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?
action_id=batch-update
{
  "edges": [
    {
      "source": "46",
      "target": "39",
      "properties": {
        "Rating": [
          5
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    },
    {
      "source": "46",
      "target": "38",
      "index": "0",
      "properties": {
        "Rating": [
          4
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    }
  ]
}
```

```
"ignoreError": true
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Edge source vertex [46] does not exist",
  "errorCode": "GES.8221"
}
```

Status Codes

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.2.10 Exporting Filtered Edges (2.2.7)

Function

This API is used to export the edge set that meets the filter criteria.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=export

Table 7-82 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-83 Request body parameters

Parameter	Mandatory	Type	Description
exportPath	Yes	String	Export path
fileName	No	String	Name of the exported file
obsParameters	Yes	String	OBS authentication parameters. For details, see Table 7-259 .
labels	Either labels or edgeFilters is mandatory.	String	Filter criteria of the relationship type
edgeFilters	Either labels or edgeFilters is mandatory.	String	Filter criteria, in JSONArray format. Vertices are filtered by property. For details, see Table 7-47 .

Response Parameters

Table 7-84 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of the edge query job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	String	Type of an asynchronous job

Example Request

Export edges that meet filter criteria (only the asynchronous mode is supported). The relationship type filter criterion is **rate**, the property names are **Score** and **Datetime**, and the export path is **demo_movie/**.

```
POST https://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=export
{
  "labels": [
    "rate"
  ],
  "edgeFilters": [
    {
      "propertyName": "Score",
      "predicate": ">=",
      "values": [
        "2"
      ]
    },
    {
      "propertyName": "Datetime",
      "predicate": "range",
      "values": [
        "1998-12-27 01:00:00",
        "2000-12-31 00:12:38"
      ],
      "type": "or"
    }
  ],
  "exportPath": "demo_movie/",
  "fileName": "export_rate.csv",
  "obsParameters": {
    "accessKey": "XXXXXXX",
```

```
"secretKey": "XXXXXXX"
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "03e774f5-29ea-4187-9508-5435f3892ead016886200",
  "jobType": 0
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.2.11 Deleting Filtered Edges (2.2.7)

Function

This API is used to delete the edge set that meets the filter criteria.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=delete

Table 7-85 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain a project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-86 Request body parameters

Parameter	Mandatory	Type	Description
labels	Either labels or edgeFilters is mandatory.	String	Filter criteria of the relationship type
edgeFilters	Either labels or edgeFilters is mandatory.	String	Filter criteria, in JSONArray format. Vertices are filtered by property. For details, see Table 7-47 .

Response Parameters

Table 7-87 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.

Parameter	Type	Description
jobId	String	ID of the edge query job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	String	Type of an asynchronous job

Example Request

Delete edges that meet filter criteria (only the asynchronous mode is supported). The relationship type filter criterion is **rate**, and the property names are **Score** and **Datetime**.

```
POST https://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=delete
{
  "labels": [
    "rate"
  ],
  "edgeFilters": [
    {
      "propertyName": "Score",
      "predicate": ">=",
      "values": [
        "2"
      ]
    },
    {
      "propertyName": "Datetime",
      "predicate": "range",
      "values": [
        "1998-12-27 01:00:00",
        "2000-12-31 00:12:38"
      ],
      "type": "or"
    }
  ]
}
```

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "jobId": "f9987cab-64d3-4b3d-ac43-e91ae0c21bef168127124",
  "jobType": 0
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.3 Metadata Operation APIs

7.1.3.1 Adding a Label (1.1.6)

Function

This API is used to add a label.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels

Table 7-88 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

- Request parameter descriptions

Table 7-89 Request body parameter descriptions

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a label A label name can contain a maximum of 256 characters. Only letters, numbers, spaces, and special characters (%@#\$.?*.+_) are allowed.
type	No	String	Label type, indicating that the label is used for vertices or edges. The value can be vertex , edge , or all . The default value is all , indicating that the label applies to vertices and edges.
properties	Yes	Object	Properties you want to add to the label. For details about the parameters, see Table 7-90 .

Table 7-90 properties parameter descriptions

Parameter	Mandatory	Type	Description
property	No	Object	Label properties. For details about the parameters, see Table 7-91 .

Table 7-91 property parameter descriptions

Parameter	Mandatory	Type	Description
name	Yes	String	Property name 1. A property name can contain a maximum of 256 characters. 2. A property name cannot contain <, >, &, ASCII 14, 15, or 30. 3. The property under a label must be unique.
cardinality	Yes	String	Cardinality type of a property. The options are: • single • list • set

Parameter	Mandatory	Type	Description
dataType	Yes	String	Data type of a property. For details, see the metadata type in Table 6-136 .
typeNameCount	No (This parameter is mandatory if dataType is enum .)	String	Total number of parameters of the enum type. This parameter controls the typeName quantity.
typeName*	No (This parameter is mandatory if dataType is enum .)	String	Names of parameters of the enum type. For example, if the value of typeNameCount is 2 , the parameter contains typeName1:science and typeName2:literature .

Response Parameters

Table 7-92 Parameter description

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Request

Add a label. The label name is **book**. The label has three properties to add.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels
{
  "name": "book",
  "type": "vertex",
  "properties": [
    {
      "property": {
        "name": "Title",
        "cardinality": "single",
```

```
    "dataType": "string"
  },
  {
    "property": {
      "name": "Version",
      "cardinality": "single",
      "dataType": "string"
    }
  },
  {
    "property": {
      "name": "Category",
      "typeName1": "science",
      "typeName2": "literature",
      "typeNameCount": "2",
      "cardinality": "single",
      "dataType": "enum"
    }
  }
]
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Label : book already exists",
  "errorCode": "GES.8801"
}
```

Status Codes

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.

Return Value	Description
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.3.2 Updating a Label (1.1.7)

Function

This API is used to either append properties to existing labels or overwrite the entire label.

URI

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{label_name}/properties
```

Table 7-93 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
label_name	Yes	String	Label name

Request Parameters

Table 7-94 Request body parameter descriptions

Parameter	Mandatory	Type	Description
properties	Yes	Object	Property array to be appended. For details about the parameters, see Table 7-95 .
override	No	Bool	The default value is false . If the value is true , the entire label is overwritten.

Table 7-95 properties parameter descriptions

Parameter	Mandatory	Type	Description
property	No	Object	Label properties. For details about the parameters, see Table 7-96 .

Table 7-96 property parameter descriptions

Parameter	Mandatory	Type	Description
name	Yes	String	Property name 1. A property name can contain a maximum of 256 characters. 2. A property name cannot contain <, >, &, ASCII 14,15 and 30. 3. The property under a label must be unique.
cardinality	Yes	String	Cardinality type of a property. The options are: • single • list • set
dataType	Yes	String	Data type of a property. For details, see the metadata type in Table 6-136 .
typeNameCount	No (This parameter is mandatory if dataType is enum .)	String	Total number of parameters of the enum type. This parameter controls the typeName quantity.
typeName*	No (This parameter is mandatory if dataType is enum .)	String	Names of parameters of the enum type. For example, if the value of typeNameCount is 2, the parameter contains typeName1:science and typeName2:literature .

Response Parameters

Table 7-97 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .

Example Request

Update a label. The label name is **book**. The label has three properties to update.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/book/properties
{
  "properties": [
    {
      "property": {
        "name": "Title",
        "cardinality": "single",
        "dataType": "string"
      }
    },
    {
      "property": {
        "name": "Version",
        "cardinality": "single",
        "dataType": "string"
      }
    },
    {
      "property": {
        "name": "Category",
        "typeName1": "science",
        "typeName2": "literature",
        "typeNameCount": "2",
        "dataType": "enum"
      }
    }
  ]
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Bad Request, properties is null",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.3.3 Querying Graph Metadata Details (1.0.0)

Function

This API is used to query graph metadata details.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema

Table 7-98 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-99 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
data	Object	Query results. This parameter is left blank when the request fails.

Table 7-100 data parameter descriptions

Parameter	Type	Description
schema	List	Definitions of each label and associated property field

Example Request

Query metadata details of a graph.

GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "data": {
    "schema": [
      {
        "label": "__DEFAULT__",
        "type": "all"
      },
      {
        "label": "friends",
        "type": "edge"
      },
      {
        "label": "movie",
        "type": "vertex",
        "properties": [
          {
            "name": "ChineseTitle",
            "type": "string",
            "cardinality": "single"
          },
          {
            "name": "Year",
            "type": "int",
            "cardinality": "single"
          },
          {
            "name": "Genres",
            "type": "string",
            "cardinality": "set"
          }
        ]
      }
    ],
    {
      "label": "user",
      "type": "vertex",
      "properties": [
        {
          "name": "Name",
          "type": "string",
          "cardinality": "single"
        },
        {
          "name": "Gender",
          "probableValue": [
            "",
            "F",
            "M"
          ],
          "type": "enum",
          "cardinality": "single"
        },
        {
          "name": "Age",
```

```

        "probableValue": [
            "",
            "Under 18",
            "18-24",
            "25-34",
            "35-44",
            "45-49",
            "50-55",
            "56+"
        ],
        "type": "enum",
        "cardinality": "single"
    },
    {
        "name": "Occupation",
        "type": "string",
        "cardinality": "single"
    },
    {
        "name": "Zip-code",
        "type": "char array",
        "cardinality": "single"
    }
]
},
{
    "label": "rate",
    "type": "edge",
    "properties": [
        {
            "name": "Score",
            "type": "int",
            "cardinality": "single"
        },
        {
            "name": "Datetime",
            "type": "date",
            "cardinality": "single"
        }
    ]
}
]
}
}
}

```

Status code: 400

Example response for a failed request

```

Http Status Code: 400
{
    "errorMessage": "Not found. Please check the input parameters.",
    "errorCode": "GES.8000"
}

```

Status Code

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.

Return Value	Description
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.3.4 Deleting a Label (2.2.18)

Function

This API is used to delete a label as well as the vertices and edges associated with the label.

URI

DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{label_name}

Table 7-101 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
label_name	Yes	String	Name of a label

Response Parameters

Table 7-102 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .
data	Object	If some labels fail to be added, this field contains the names of the failed labels and the failure causes.

Example Request

Delete a label as well as the vertices and edges associated with the label.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/{labelName}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232",
  "jobType": 3
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.

Return Value	Description
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.3.5 Batch Adding Labels (2.2.21)

Function

This API is used to add labels in batches.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/action?action_id=batch-add

Table 7-103 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-104 Request body parameter descriptions

Parameter	Mandatory	Type	Description
labels	Yes	Object	Metadata labels

Table 7-105 label parameter descriptions

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a label. Enter a maximum of 256 characters. Only letters, numbers, spaces, and the following special characters (%@#\$.?*.+_-) are allowed.
type	No	String	Label type, indicating that the label is used for vertices or edges. The value can be vertex , edge , or all . The default value is all , indicating that the label applies to vertices and edges.
properties	Yes	Object	Property array to be added. Each array element is a property. For details, see Table 7-106 .

Table 7-106 property parameter descriptions

Parameter	Mandatory	Type	Description
name	Yes	String	Property name 1. A property name can contain a maximum of 256 characters. 2. A property name cannot contain <, >, &, ASCII 14, 15 or 30. 3. The property under a label must be unique.
cardinality	Yes	String	Cardinality type of a property. The options are: • single • list • set
dataType	Yes	String	Data type of a property. For details, see the metadata type in Table 6-136 .
typeNameCount	No (This parameter is mandatory if dataType is enum .)	String	Total number of parameters of the enum type. This parameter controls the typeName quantity.

Parameter	Mandatory	Type	Description
typeName*	No (This parameter is mandatory if dataType is enum .)	String	Names of parameters of the enum type. For example, if the value of typeNameCount is 2 , the parameter contains typeName1:science and typeName2:literature .

Response Parameters

Table 7-107 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the execution is successful, the value is success .
data	Object	If some labels fail to be added, this field contains the names of the failed labels and the failure causes.

Example Request

Add metadata labels and their properties in batches. The names of the metadata labels are **book** and **movie**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels/action?
action_id=batch-add
{
  "labels": [
    {
      "name": "book",
      "type": "vertex",
      "properties": [
        {
          "property": {
            "name": "title",
```

```
        "cardinality": "single",
        "dataType": "string"
      }
    ],
    {
      "name": "movie",
      "type": "vertex",
      "properties": [
        {
          "property": {
            "name": "movieid",
            "cardinality": "single",
            "dataType": "int"
          }
        }
      ]
    }
  ]
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Example response for a partially successful request

```
Http Status Code: 200
{
  "result": "partial success",
  "data": {
    "failed": [
      {
        "cause": "label name is invalid which can only contain letters, digits, space, %, @, #, $, :, ?, *, ., +, - and",
        "labelName": "book<"
      }
    ]
  }
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Label : movie already exists",
  "errorCode": "GES.8812"
}
```


Status Code

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.3.6 Querying a Schema

Function

This API is used to query the generated schema structure.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure?detail={details}

Table 7-108 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
details	No	String	Detailed information of the schema. The value can be SIMPLE or FULL . If you set this parameter to SIMPLE , only the labels of vertices or edges are returned. If you set it to FULL , the number of vertices or edges is returned in addition to the labels. If this parameter is left empty, the default value SIMPLE is used.

Response Parameters

Table 7-109 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
schema	Object	Schema structure. For details, see schema parameter description .

Table 7-110 schema parameter description

Parameter	Type	Description
vertices	Object	Vertex result set. If the graph is empty, the return value is empty. For details, see vertices parameter description .
edges	Object	Edge result set. If the graph is empty, the return value is empty. For details, see edges parameter description .

Table 7-111 vertices parameter description

Parameter	Type	Description
vertex	String	Label name
weight	Integer	Number of vertices that have the label

Table 7-112 edges parameter description

Parameter	Type	Description
source	String	Label the start vertex.
target	String	Label the end vertex.

Parameter	Type	Description
relation	String	Relationship label
weight	Integer	Number of edges that have the label

Example Request

Query the generated schema structure.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure?detail=SIMPLE
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "schema": {
    "vertices": [
      {
        "vertex": "user",
        "weight": 100
      },
      {
        "vertex": "movie",
        "weight": 46
      }
    ],
    "edges": [
      {
        "weight": 1209,
        "source": "user",
        "target": "movie",
        "relation": "rate"
      },
      {
        "weight": 450,
        "source": "user",
        "target": "user",
        "relation": "default"
      }
    ]
  }
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.3.7 Generating a Schema

Function

This API is used to generate a schema where labels are represented with vertices and the relationship between the labels are represented with edges, and store the schema in an OBS bucket.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure/build

Table 7-113 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Response Parameters

Table 7-114 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of the vertex query job. This parameter is left blank when the request fails.
jobType	String	Type of an asynchronous job

Example Request

Generate a schema where labels are represented with vertices and the relationship between the labels are represented with edges.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/structure/build
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "jobId": "2e0c08e1-3fbb-4b33-8776-4809176068d7154236181",
  "jobType": 1
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.3.8 Generating Data Assets

Function

This API is used to generate data assets.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets

Table 7-115 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-116 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the edge query job. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	Integer	Job type. This parameter is left blank if the request fails.

Example Request

Generate data assets.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets
{}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "40cef36d-9b06-4fe1-bea8-0d8f0393a362000313144",
  "jobType": 1
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.1.3.9 Obtaining Data Assets

Function

This API is used to obtain data assets.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets

Table 7-117 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-118 Response body parameter descriptions

Parameter	Type	Description
generating	Boolean	Whether data assets are being generated
progress	String	Progress of generating data assets
last_generate_time	String	Last time when a data asset is generated
data_assets	Object	Number of vertices and edges under different labels
errorCode	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
errorMessage	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.

Table 7-119 data_assets parameter descriptions

Parameter	Type	Description
vertex	Object	Number of vertices under different labels
edge	Object	Number of edges under different labels

Example Request

Obtain data assets.

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets
```

Example Response

Status code: 200

There are several scenarios for example response for successful requests, including:

- A data asset has never been generated.

```
Http Status Code: 200
{ }
```

- A data asset has never been generated and is being generated.

```
Http Status Code: 200
{
  "progress": "10.05%",
  "generating": true
}
```

- A data asset has been generated and a latest one is being generated.

```
Http Status Code: 200
{
  "last_generate_time": "2022-1-3 12:34:12",
  "data_assets": {
    "vertex": {
      "label1": 3,
      "label2": 14
    },
    "edge": {
      "label3": 123,
      "label4": 435
    }
  },
  "progress": "10.05%",
  "generating": true
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

7.1.4 Index Operation APIs

7.1.4.1 Creating an Index (1.1.6)

Function

This API is used to create indexes based on the specified information such as indexName and IndexType. Currently, composite indexes are supported.

- Composite indexes include global vertex indexes (GlobalCompositeVertexIndex) and global edge indexes (GlobalCompositeEdgeIndex). Composite indexes can be used to create indexes on labels and properties. Indexes can accelerate the query speed.

Index Feature

Feature	Fuzzy Search	Speed	Flexibility
Composite indexes	No	Fast	Fixed composite property keys only

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/indices

Table 7-120 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-121 Request body parameter descriptions

Parameter	Mandatory	Type	Description
indexName	Yes	String	Index name. The name can contain a maximum of 63 characters. Only letters, numbers, hyphens (-), and underscores (_) are allowed.
indexType	Yes	String	Index type. The value is case-sensitive. GlobalCompositeVertexIndex is a global composite vertex index.
hasLabel	No	String	Whether labels exist. The default value is false . • true • false
indexProperty	No (If hasLabel is false or null , this parameter is mandatory.)	String	Index property list

NOTE

- If a property is of type string or char array, the value must not exceed 40 bytes. Any extra characters will be removed.
- You can create up to 10 indexes.

Response Parameters

Table 7-122 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs.
jobType	String	Type of an asynchronous job

Example Request

Create a composite index. The index name is **ageIndex** and the index type is global vertex index.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
{
  "indexName": "ageIndex",
  "indexType": "GlobalCompositeVertexIndex",

  "hasLabel": "true",
  "indexProperty": ["age"]
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232",
  "jobType": 8
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.4.2 Deleting an Index (1.1.6)

Function

This API is used to delete an index based on a specified index name.

URI

DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/indices/{index_name}

Table 7-123 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
index_name	Yes	String	Index name

Response Parameters

Table 7-124 Parameter description

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	String	Type of an asynchronous job

Example Request

Delete an index by name. The index name is **ageIndex**.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices/ageIndex
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "d7e4481b-29e6-443b-85e9-0356f0a371ad000313144",
  "jobType": 10
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
}
```

```
}
  "errorCode": "GES.8000"
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.4.3 Querying Indexes (1.1.6)

Function

This API is used to query all indexes created on a graph.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/indices

Table 7-125 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Response Parameters

Table 7-126 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
data	Object	Index data you want to query
result	String	Query results. If the query is successful, success is displayed.
indices	List	Indexes of the query results
indexType	String	Index types of the query results
indexName	String	Index names of the query results
indexProperty	List	Index properties of the query results
hasLabel	String	Whether the indexes of the query results contain labels NOTE For full-text indexes, the default value is false .

Example Request

Query all indexes created on a graph.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
```



```
"data": {
  "result": "success",
  "indices": [
    {
      "indexType": "GlobalCompositeVertexIndex",
      "indexName": "ageIndx",
      "indexProperty": [
        "age"
      ],
      "hasLabel": "true"
    }
  ]
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.5 Gremlin Operation APIs

7.1.5.1 Executing Gremlin Queries (1.0.0)

Function

This API is used to return the query result of a Gremlin statement.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-gremlin-query

Table 7-127 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-128 Request body parameter

Parameter	Mandatory	Type	Description
command	Yes	String	Query command (Gremlin language)

Response Parameters

Table 7-129 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	Object	Query results. This parameter is left blank when the request fails.

Example Request

Perform the Gremlin query operation. If the query command is **g.V().limit(100)**, all vertices are queried, but the number of returned vertices is limited to 100.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-gremlin-query
```

```
{
  "command": "g.V().limit(100)"
}
```

 NOTE

- **SERVER_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- The size of the request body cannot exceed 64 MB.

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "data": {
    "runtime": 0.775425022,
    "vertices": [
      {
        "id": "Vivian",
        "label": "user",
        "properties": {
          "Occupation": [
            "artist"
          ],
          "Name": [
            "Vivian"
          ],
          "Zip-code": [
            "98133"
          ],
          "Gender": [
            "F"
          ],
          "Age": [
            "25-34"
          ]
        }
      },
      .....
    ]
  }
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "org.apache.tinkerpop.gremlin.driver.exception.ResponseException: No such property: g1  
for class: Script4",
  "errorCode": "GES.8503"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error.

Return Value	Description
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.6 Algorithm APIs

7.1.6.1 Running Algorithms (1.0.0)

Function

This API is used to run specified algorithms based on input parameters.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm

Table 7-130 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

For details about the parameters, see [Common algorithm parameters](#).

Response Parameters

Table 7-131 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	Integer	Job type. This parameter is left blank when the request fails.

Example Request

Execute a specified algorithm. The algorithm name is **pagerank**, the weight coefficient is **0.85**, the convergence precision is **0.00001**, the maximum number of iterations is **1000**, and traversal is performed along edge directions.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm
{
  "algorithmName": "pagerank",
  "parameters": {
    "alpha": 0.85,
    "convergence": 0.00001,
    "max_iterations": 1000,
    "directed": true
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

Status code: 4200

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.6.2 Algorithm API Parameter References

7.1.6.2.1 Common Algorithm Parameters

Request Example

```
{
  "algorithmName": "XXX",
  "parameters": {
    ...
  }
}
```

Request Parameters

Table 7-132 Request body parameter descriptions

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name. The options are (algorithm names you can call): • pagerank • personalrank • kcore • k_hop • shortest_path • all_shortest_paths • filtered_shortest_path • sssp • shortest_path_of_vertex_sets • n_paths • closeness • subgraph_matching • topicrank • label_propagation • louvain • link_prediction • node2vec • realtime_recommendation • common_neighbors • connected_component • degree_correlation • triangle_count • cluster_coefficient • common_neighbors_of_vertex_sets • all_shortest_paths_of_vertex_sets • filtered_circle_detection • filtered_all_pairs_shortest_paths • filtered_all_shortest_paths • filtered_n_paths
parameters	Yes	Object	Algorithm parameters. For details, see the parameter descriptions of each algorithm.

Table 7-133 New request body parameters in version 2.1.7

Parameter	Mandatory	Type	Description
executionMode	No	String	• sync: synchronous • async: asynchronous The default value is async . Supported algorithms are as follows (algorithm names you can call): • k_hop • shortest_path • all_shortest_paths • filtered_shortest_path • shortest_path_of_vertex_sets • n_paths • realtime_recommendation • filtered_all_pairs_shortest_paths • filtered_all_shortest_paths
offset	No	Integer	Synchronization result offset. The default value is 0 . NOTE This parameter is available when executionMode is sync . Supported algorithms are as follows (algorithm names you can call): • k_hop • shortest_path • all_shortest_paths • shortest_path_of_vertex_sets • n_paths • realtime_recommendation • filtered_all_pairs_shortest_paths • filtered_all_shortest_paths

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of returned synchronization results. The maximum value is 100000. The default value is 100000. NOTE This parameter is available when executionMode is sync. Supported algorithms are as follows (algorithm names you can call): • k_hop • shortest_path • all_shortest_paths • shortest_path_of_vertex_sets • n_paths • realtime_recommendation • filtered_all_pairs_shortest_paths • filtered_all_shortest_paths

Table 7-134 New request body parameters in version 2.2.4

Parameter	Mandatory	Type	Description
vertex_filter	No	Object	Filter criteria for the vertices on a path. Supported algorithms are as follows (algorithm names you can call): • filtered_shortest_path • filtered_all_pairs_shortest_paths • filtered_all_shortest_paths For details, see Table 7-273 in the Filtered-query API.
edge_filter	No	Object	Filter criteria for the edges (relationships) on a path. Supported algorithms are as follows (algorithm names you can call): • filtered_shortest_path • filtered_all_pairs_shortest_paths • filtered_all_shortest_paths For details, see Table 7-273 in the Filtered-query API.

Parameter	Mandatory	Type	Description
filters	No	Object	Filter criteria. Each element in the array corresponds to a filter. This parameter applies only to filtered circle detection. For details about the format, see filters element formats . Supported algorithms: • filtered_n_paths • filtered_circle_detection • topicrank

Response Parameters

Table 7-135 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.
status	String	Job status returned for a successful query. The options are pending , running , and success . This parameter is left blank when the query fails.
data	Object	Algorithm execution result. This parameter is left blank when the query fails.

Example Response

Algorithms are executed based on input parameters. You can call [Querying Job Status and Execution Results](#) to use the **job_id** returned by the algorithm to obtain the algorithm execution result.

Status code: 200

Example response for a successful request

```
{
  "data": {
    "outputs": {
      $response_data //Result of each algorithm. The results vary with the algorithm.
      "runtime": 1.365867,
      "data_return_size": 3,
      "data_offset": 0,
```

```
    "data_total_size": 100
  }
},
"status": "success"
}
```

 NOTE

response_data indicates the result of each algorithm. The results vary with algorithms.

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Running algorithm [XXXX] error: YYYYYYYY!",
  "errorCode": "GES.8301"
}
```

7.1.6.2.2 PageRank (1.0.0)

Table 7-136 parameters parameter description

Parameter	Mandatory	Type	Description
alpha	No	Double	Weight coefficient (also called damping coefficient). The value range is (0, 1). The default value is 0.85 .
convergence	No	Double	Convergence The value range is (0, 1). The default value is 0.00001 .
max_iterations	No	Integer	Maximum iterations An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000]. The default value is 1000 .
num_thread	No	Integer	Number of concurrent threads. The value ranges from 1 to 40. If the value is less than 1, it is automatically set to 1 . If the value is greater than 40, it is automatically set to 40 . The default value is 4 .
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is true .

 NOTE**convergence** and **max_iterations**:

The algorithm is terminated when either the maximum number of iterations is reached or the convergence precision is met.

1. Generally, a smaller convergence precision and larger number of iterations lead to a better effect of the algorithm.
2. To meet a certain convergence precision, you should set the number of iterations as large as possible.
3. A larger number of iterations means a longer algorithm running time. To ensure that the algorithm runs at a certain number of iterations (that is, in a fixed duration), you should set the convergence precision as small as possible.

Table 7-137 response_data parameter description

Parameter	Type	Description
results	List	PageRank value of each node, in the format of [{"id":vertexId,"pagerank":rankValue},...], where, • vertexId is of the string type. • rankValue is of the double type.

7.1.6.2.3 PersonalRank (1.0.0)

Table 7-138 parameters parameter description

Parameter	Mandatory	Type	Description
source	Yes	String	Node ID
alpha	No	Double	Weight coefficient (also called damping coefficient). The value range is (0, 1). The default value is 0.85 .
convergence	No	Double	Convergence The value range is between 0 and 1, excluding 0 and 1. The default value is 0.00001 .
max_iterations	No	Integer	Maximum iterations. An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000]. The default value is 1000 .

Parameter	Mandatory	Type	Description
num_thread	No	Integer	Number of concurrent threads. The value ranges from 1 to 40. If the value is less than 1, it is automatically set to 1 . If the value is greater than 40, it is automatically set to 40 . The default value is 4 .
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is true .
weight	No	String	Null or string • Null: The default weight and distance of edges are 1. • String: The property of the corresponding edge is the weight. If the edge does not have a property, the weight is 1 by default. NOTE The weight of an edge must be greater than 0.

 NOTE

For how to adjust the values of **max_iterations** and **convergence**, see [Iterations and Convergence of PageRank](#).

Table 7-139 response_data parameter description

Parameter	Type	Description
source	String	-
results	List	PersonalRank value of each node, in the format of [{"id":vertexId,"personalrank":rankValue},...], where, • vertexId is of the string type. • rankValue is of the double type.

7.1.6.2.4 K-core (1.0.0)

Table 7-140 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
k	Yes	Number of cores The algorithm returns vertices whose number of cores is greater than or equal to k.	Integer	Greater than or equal to 0	-

Table 7-141 response_data parameter description

Parameter	Type	Description
results	List	Coreness value ($\text{coreness} \geq k$) of each vertex, in the format of <code>[{"id":vertexId,"coreness":corenessValue},...]</code> , where, • vertexId is of the string type. • rankValue is of the double type.

7.1.6.2.5 K-hop (1.0.0)

Table 7-142 parameters parameter description

Parameter	Mandatory	Type	Description
k	Yes	Integer	Number of hops. The value ranges from 1 to 100.
num_thread	No	Integer	Number of concurrent threads. The value ranges from 1 to 40. If the value is less than 1, it is automatically set to 1 . If the value is greater than 40, it is automatically set to 40 . The default value is 4 .
source	Yes	String	Vertex ID

Parameter	Mandatory	Type	Description
mode	No	String	Direction. The options are as follows: • OUT: Hop from the outgoing edges • IN: Hop from the incoming edges • ALL: Hop from edges in both directions The default value is OUT .
statistics	No	Boolean	Whether to only return the neighbor count statistics results, with a value of true or false . The default value is false .

Table 7-143 response_data parameter description

Parameter	Type	Description
vertices	List	ID of the vertex within k hops. The format is as follows: [vertexId,...], where vertexId is of the string type.
source	String	Source vertex ID
k	Integer	Number of hops
k_hop_neighbors	Integer	Number of vertices within k hops (excluding the source vertex)

7.1.6.2.6 Common Neighbors (1.0.0)

Table 7-144 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Source vertex ID	String	-	-
target	Yes	Target vertex ID	String	-	-

Table 7-145 response_data parameter description

Parameter	Type	Description
vertices	List	Common neighbor vertices. The format is as follows: [vertexId,...], where vertexId is of the string type
common_neighbors	Integer	Number of common neighbor vertices
source	String	Source vertex ID
target	String	Target vertex ID

7.1.6.2.7 Common Neighbors of Vertex Sets (2.2.13)

Table 7-146 Parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
sources	Yes	Source vertex ID set	String	The value is in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana . The maximum ID number is 100000.	-
targets	Yes	Target vertex ID set	String	The value is in the standard CSV format. IDs are separated by commas (,), for example, Mike,Amy . The maximum ID number is 100000.	-

Parameter	Mandatory	Description	Type	Value Range	Default Value
restricted	No	Whether other constraints are included	Boolean	true or false false: There is no additional constraint. The found common neighbors are the intersection of the neighborhoods corresponding to the source vertex set and target vertex set. true: There are additional constraints. The found common neighbors are not only the intersection of the neighborhoods corresponding to the source vertex set and target vertex set, but each vertex in the common neighbor set has at least two neighboring vertices in the source vertex set and target vertex set.	true

Table 7-147 response_data parameter description

Parameter	Type	Description
vertices	List	Common neighbor vertices. The format is as follows: [vertexId,...], where vertexId is of the string type.
common_neighbors	Integer	Number of common neighbors

7.1.6.2.8 Link Prediction (1.0.0)

Table 7-148 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Source vertex ID	String	-	-
target	Yes	Target vertex ID	String	-	-

Table 7-149 response_data parameter description

Parameter	Type	Description
source	String	Source vertex ID
target	String	Target vertex ID
link_prediction	Double	Link prediction result

7.1.6.2.9 Shortest Path (2.1.5)

Table 7-150 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Source vertex ID of a path	String	-	-
target	Yes	Target vertex ID of a path The target value must not be identical to the source value.	String	-	-

Parameter	Mandatory	Description	Type	Value Range	Default Value
weight	No	Weight of an edge	String	Empty or character string - Empty: The default weight and distance of edges are 1. - Character string: The property of the corresponding edge is the weight. If the edge does not have a property, the weight is 1 by default. NOTE The weight of an edge must be greater than 0.	-
directed	No	Whether to consider the edge direction	Boolean	The value can be true or false .	false
timeWindow	No	Time window used for time filtering	Object	For details, see Table 7-151 . NOTE timeWindow does not support the shortest path with weight. That is, parameters timeWindow and weight cannot be both specified.	-

Table 7-151 timeWindow parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
filterName	Yes	Name of the time property used for time filtering	String	Character string: The property on the corresponding vertex/edge is used as the time.	-
filterType	No	Filtering by vertex or edge	String	V : filtering by vertex E : filtering by edge BOTH : filtering by vertex and edge	BOTH
startTime	No	Start time	String	Date character string or timestamp	-

Parameter	Mandatory	Description	Type	Value Range	Default Value
endTime	No	End time	String	Date character string or timestamp	-

Table 7-152 response_data parameter description

Parameter	Type	Description
path	List	Shortest path. The format is as follows: [vertexId,...] where vertexId is of the string type.
source	String	Source vertex ID
target	String	Target vertex ID

7.1.6.2.10 All Shortest Paths (1.0.12)

Table 7-153 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Source vertex ID of a path	String	-	-
target	Yes	Target vertex ID of a path	String	-	-
directed	No	Whether to consider the edge direction	Boolean	true or false	false

Table 7-154 response_data parameter description

Parameter	Type	Description
paths	List	All shortest paths between the source vertex and target vertex. The format is as follows: [[path1],[path2]] For the format of path , see Shortest Path .

Parameter	Type	Description
paths_number	Integer	Number of paths
source	String	Source vertex ID
target	String	Target vertex ID

7.1.6.2.11 Filtered Shortest Path (2.2.4)

Request

- Parameter descriptions

Table 7-155 parameters parameter descriptions

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex ID of a path
target	Yes	String	Target vertex ID of a path
directed	No	Boolean	Whether to consider the edge direction. The default value is false .
num_thread	No	Integer	Number of concurrent threads. The value ranges from 1 to 40 . If the value is less than 1 , it is automatically set to 1 . If the value is greater than 40 , it is automatically set to 40 . The default value is 4 .

- Request example
 - Synchronization

```
{
  "executionMode": "sync",
  "algorithmName": "filtered_shortest_path",
  "edge_filter": {
    "property_filter": {
      "leftvalue": {
        "label_name": "labelName"
      },
      "predicate": "IN",
      "rightvalue": {
        "value": [
          "xxx",
          "rate"
        ]
      }
    }
  },
  "vertex_filter": {
    "property_filter": {
```

```
    "leftvalue": {
      "property_name": "title"
    },
    "predicate": "PREFIX",
    "rightvalue": {
      "value": "tr_"
    }
  }
},
"parameters": {
  "source": "tr_1",
  "target": "tr_117",
  "directed": true
}
}
```

– Asynchronization

```
{
  "executionMode": "async",
  "algorithmName": "filtered_shortest_path",
  "edge_filter": {
    "property_filter": {
      "leftvalue": {
        "label_name": "labelName"
      },
      "predicate": "IN",
      "rightvalue": {
        "value": [
          "xxx",
          "rate"
        ]
      }
    }
  },
  "vertex_filter": {
    "property_filter": {
      "leftvalue": {
        "property_name": "title"
      },
      "predicate": "PREFIX",
      "rightvalue": {
        "value": "tr_"
      }
    }
  },
  "parameters": {
    "source": "tr_1",
    "target": "tr_117",
    "directed": true
  }
}
```

Response

- Synchronous parameter descriptions

Table 7-156 response_data parameter descriptions

Parameter	Mandatory	Type	Description
path	Yes	List	Vertex result set. If the last layer of filters is vertex filtering, the data contains vertices.
source	Yes	String	Source vertex ID

Parameter	Mandatory	Type	Description
target	Yes	String	Target vertex ID
runtime	Yes	Double	Algorithm running time

- Response example

- Synchronous response example (successful request)

```
{
  "data": {
    "outputs": {
      "path": [
        "tr_1",
        "tr_5",
        "tr_26",
        "tr_117"
      ],
      "runtime": 0.735766,
      "source": "tr_1",
      "target": "tr_117"
    }
  }
}
```

- Synchronous response example (failed request)

```
{
  "errorMessage": "graph [tesdt_117] is not found",
  "errorCode": "GES.8402"
}
```

- Asynchronous response parameters

Table 7-157 response_data parameter descriptions

Parameter	Mandatory	Type	Description
errorMessage	No	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.
errorCode	No	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.
jobId	No	String	ID of the algorithm execution job. This parameter is left blank when the request fails.
jobType	No	Integer	Job type. This parameter is left blank when the request fails.

- Example response

- Asynchronous response example (successful request)

```
{  
  "jobId": "500dea8f-9651-41fe-8299-c20f13a032ea",  
  "jobType": 2  
}
```

- Asynchronous response example (failed request)

```
{  
  "errorMessage": "Not found. Please check the input parameters.",  
  "errorCode": "GES.8000"  
}
```

7.1.6.2.12 SSSP (1.0.0)

Table 7-158 parameters parameter descriptions

Parameter	Mandatory	Type	Value Range	Default Value	Description
source	Yes	String	-	-	Vertex ID.
directed	No	Boolean	true or false	true	Whether an edge is directed.

Table 7-159 response_data parameter descriptions

Parameter	Type	Description
distance	List	Path length of each vertex in the graph from the source vertex. The format is as follows: [{vertexId:distanceValue},...], where vertexId is of the string type. distanceValue is of the double type.

7.1.6.2.13 Shortest Path of Vertex Sets (2.1.5)

Table 7-160 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
sources	Yes	Source vertex ID set	String	The value is in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana . The maximum ID number is 100000.	-
targets	Yes	Target vertex ID set	String	The value is in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana . The maximum ID number is 100000.	-
directed	No	Whether an edge is directed	Boolean	true or false	false
timeWindow	No	Time window used for time filtering	Object	For details, see Table 7-161 .	-

Table 7-161 timeWindow parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
filterName	No	Name of the time property used for time filtering	String	Character string: The property on the corresponding vertex/edge is used as the time.	-
filterType	No	Filtering by vertex or edge	String	V : filtering by vertex E : filtering by edge BOTH : filtering by vertex and edge	BOTH
startTime	No	Start time	String	Date character string or timestamp	-
endTime	No	End time	String	Date character string or timestamp	-

Table 7-162 response_data parameter description

Parameter	Type	Description
path	List	Shortest path. The format is as follows: [vertexId,...] where vertexId is of the string type.
source	String	Source vertex ID
target	String	Target vertex ID

7.1.6.2.14 n-Paths (1.1.2)

Table 7-163 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Source vertex ID of a path	String	-	-
target	Yes	Target vertex ID of a path	String	-	-
directed	No	Whether to consider the edge direction	Boolean	true or false	false
n	No	Number of paths	Integer	[1,100]	10
k	No	Number of hops	Integer	[1,10]	5

Table 7-164 response_data parameter description

Parameter	Type	Description
paths	List	Paths between the source vertex and target vertex. The format is as follows: [[path1],[path2]] For the format of path , see Shortest Path .
paths_number	Integer	Number of paths

Parameter	Type	Description
source	String	Source vertex ID
target	String	Target vertex ID

7.1.6.2.15 Filtered n-Paths (2.2.22)

Introduction

The filtered n-Paths algorithm is used to find no more than n k-hop loop-free paths between the source and target vertices. The start vertex (source), end vertex (target), number of hops (k), number of paths (n), and filter criteria (filters) are the parameters for the algorithm.

- Algorithm name: filtered_n_paths
- filtered_n_paths

Applicable Scope

Any network

Request Parameters

Table 7-165 Body format

Field	Mandatory	Type	Description
algorithmName	Yes	String	The value is filtered_n_paths .
parameters	Yes	Json	For details about the format, see Table 7-166 .
filters	Yes	JSon Array	Filter criteria. Each element in the array corresponds to a filter. For details about the format, see Table 7-167 .

Table 7-166 Parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Source vertex	String	Internal vertices	None
target	Yes	Target vertex	String	Internal vertices	None

Parameter	Mandatory	Description	Type	Value Range	Default Value
k	Yes	Number of hops	Integer	[2,6]	2
n	Yes	Number of paths	Integer	[1,1000]	1

Table 7-167 filters element format

Parameter	Mandatory	Type	Value Range	Default Value	Description
edge_filter	No	json	N/A	N/A	Filter criteria for full-graph edge query
vertex_filter	No	json	None	None	Filter criteria for full-graph vertex query

Table 7-168 response_data parameter description

Field	Type	Description
path_length	integer	Path length
paths_number	integer	Number of paths
paths	Json Array	Path set, in the format of [[path1],[path2]]
source	String	Source vertex
target	String	Target vertex

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm
{
  "algorithmName": "filtered_n_paths",
  "filters": [
    {
      "edge_filter":
      {
        "property_filter":
        {
          "leftvalue":
          {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue":
          {
```

```
        "value": "default"
      }
    }
  }
},
"parameters":
{
  "k": 6,
  "n": 100,
  "source": "111",
  "target": "56"
}
}
```

Response

```
{
  "jobId": "b14f6380-f115-46ab-990e-9a76a984ebd2154236181",
  "jobType": 2
}
```

Example for Querying a Job

GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{jobId}/status

Query Result

```
{
  "data": {
    "outputs": {
      "data_return_size": 3,
      "paths": [
        [
          "111",
          "119",
          "58",
          "96",
          "82",
          "57",
          "56"
        ],
        [
          "111",
          "119",
          "58",
          "61",
          "76",
          "57",
          "56"
        ],
        [
          "111",
          "119",
          "58",
          "79",
          "76",
          "57",
          "56"
        ]
      ],
      "runtime": 0.000308,
      "source": "111",
      "path_length": 6,
      "data_offset": 0,
      "paths_number": 3,
      "data_total_size": 3,
      "target": "56"
    }
  }
}
```

```
},  
  "status": "success"  
}
```

7.1.6.2.16 Filtered All Pairs Shortest Paths (2.2.17)

Table 7-169 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
sources	Yes	Set of start vertex IDs. The value is in the standard CSV input format, that is, multiple vertex IDs are separated by commas (,).	String	The number of source vertices cannot exceed 10,000.	-
targets	Yes	Set of end vertex IDs. The value is in the standard CSV input format, that is, multiple vertex IDs are separated by commas (,).	String	The number of target vertices cannot exceed 10,000.	-
directed	No	Whether an edge is directed	Boolean	The value can be true or false .	false
cutoff	No	Maximum length	Integer	[1,100]	6
path_limit	No	Maximum number of paths	Integer	- For synchronous tasks: The value ranges from 1 to 100000. The default value is 100000. - For asynchronous tasks: The value ranges from 1 to 1000000. The default value is 1000000.	100000/1000000

 NOTE

- Synchronous tasks: Number of source vertices x Number of target vertices x Maximum path length (**cutoff**) ≤ 1000000, Maximum number of paths (**path_num**) x Maximum path length (**cutoff**) ≤ 1000000.
- This algorithm checks memory capacity. When the memory is insufficient, the error "memory is not enough" is reported.

Table 7-170 response_data parameter description

Parameter	Type	Description
batch_paths	List	Batch paths. Format: [paths_element,...] where Paths_element indicates the path from a source to a target. The format is as follows: { "paths": [["Alice", "Janet", "Sue", "Serena", "Bonnie"]], "source": "Alice", "target": "Bonnie" },
paths_number	Integer	Number of paths

7.1.6.2.17 All Shortest Paths of Vertex Sets (2.2.15)

Table 7-171 Parameter description

Parameter	Mandatory	Type	Value Range	Default Value	Description
sources	Yes	String	The value is in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana . The maximum ID number is 100000.	-	Source vertex ID set
targets	Yes	String	The value is in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana . The maximum ID number is 100000.	-	Target vertex ID set
directed	No	Boolean	true or false . It is a Boolean value.	false	Whether an edge is directed

Table 7-172 response_data parameter description

Parameter	Type	Description
paths	List	All shortest paths between the source vertex and target vertex. The format is as follows: [[path1],[path2]] For the format of path , see Shortest Path .
sources	String	Source ID set of a path
targets	String	Target ID set of a path

7.1.6.2.18 Filtered All Shortest Paths (2.2.17)

Parameters

Table 7-173 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Source vertex ID	String	- -	-
target	Yes	Target vertex ID	String	-	-
directed	No	Whether an edge is directed	Boolean	The value can be true or false .	false

Table 7-174 response_data parameter description

Parameter	Type	Description
paths	List	Paths between the source and target vertices. The format is as follows: [[path1],[path2]] where For the format of each path, see Shortest Path .
paths_number	Integer	Number of paths
source	String	Source vertex ID
target	String	Target vertex ID

Example Request

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm
{
  "algorithmName": "filtered_all_shortest_paths",
  "edge_filter": {
    "property_filter": {
      "leftvalue": {
        "label_name": "labelName"
      },
      "predicate": "=",
      "rightvalue": {
        "value": "friends"
      }
    }
  },
  "parameters": {
    "source": "Alice",
    "target": "Jay",
  }
}
```

```
    "directed": true
  }
}
```

Response

```
{
  "data": {
    "outputs": {
      "data_return_size": 8,
      "paths": [
        [
          "Alice",
          "Janet",
          "Yvette",
          "Willy",
          "Jay"
        ],
        ...
        [
          "Alice",
          "Jacob",
          "Jimmy",
          "Cary",
          "Jay"
        ]
      ],
      "runtime": 0.005276,
      "source": "Alice",
      "data_offset": 0,
      "paths_number": 8,
      "data_total_size": 8,
      "target": "Jay"
    }
  }
}
```

7.1.6.2.19 Connected Component (1.0.0)

 NOTE

This algorithm can run without specifying its **parameters**.

Table 7-175 response_data parameter description

Parameter	Type	Description
Max_WCC_size	Integer	Maximum number of vertices in the largest connected component
Max_WCC_id	String	ID of the largest connected component
community	List	Connected component set (community) corresponding to each vertex, in [{vertexId:communityId},...] format. where vertexId is of the string type. communityId is of the string type.

7.1.6.2.20 Label Propagation (2.1.8)

Table 7-176 parameters parameter descriptions

Parameter	Mandatory	Type	Value Range	Description	Default Value
convergence	No	Double	The value is a real number between 0 and 1.	Convergence.	0.00001
max_iterations	No	Integer	Maximum iterations. An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000]. The default value is 1000 .	Maximum iterations.	1000
initial	No	String	Empty or character string - Empty: Each vertex is allocated with a unique initialization label. This method is applicable to scenarios where no vertex label information exists. - Character string: The value of the property field corresponding to each vertex is used as the initialization label (the type is string, and the initialization label field is left blank for a vertex with unknown labels). This method is applicable to scenarios where some vertex labels are marked to predict unknown vertex labels. NOTE If the value of initial is a character string, the number of vertices with initialization labels must be greater than 0 and less than the total number of vertices.	Name of the property used as the initialization label on a vertex.	-

Parameter	Mandatory	Type	Value Range	Description	Default Value
weight	No	String	Null or string - Null: The default weight and distance of edges are 1. - String: The property of the corresponding edge is the weight. If the edge does not have a property, the weight is 1 by default. NOTE The weight of an edge must be greater than 0.	Weight of an edge.	Null

 NOTE

For how to adjust the values of **max_iterations** and **convergence**, see [Iterations and Convergence of PageRank](#).

Table 7-177 response_data parameter descriptions

Parameter	Type	Description
community	List	Community corresponding to each vertex. The format is as follows: [{vertexId:communityId},...] where vertexId is of the string type. communityId is of the string type.

7.1.6.2.21 Louvain (2.2.1)**Table 7-178** parameters parameter descriptions

Parameter	Mandatory	Description	Type	Value Range	Default Value
convergence	No	Convergence	Double	The value is a real number between 0 and 1.	0.00001

Parameter	Mandatory	Description	Type	Value Range	Default Value
max_iterations	No	Maximum iterations	Integer	An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000]. The default value is 100 .	100
weight	No	Weight of an edge	String	Empty or null character string - Empty: The default weight and distance of edges are 1. - Character string: The property of the corresponding edge is the weight. If the edge does not have a property, the weight is 1 by default. NOTE The weight of an edge must be greater than 0 .	Null

 NOTE

For how to adjust the values of **max_iterations** and **convergence**, see [Iterations and Convergence of PageRank](#).

Table 7-179 response_data parameter descriptions

Parameter	Type	Description
modularity	Double	Modularity
community_num	Integer	Number of communities

Parameter	Type	Description
community	List	Community corresponding to each vertex. The format is as follows: [{vertexId:communityId},...] where vertexId is of the string type. communityId is of the string type.

7.1.6.2.22 Node2vec (1.0.5)

Table 7-180 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
P	No	Rollback parameter	Double	Greater than 0	1
Q	No	Forward parameter	Double	Greater than 0	1
dim	No	Mapping dimension	Integer	An integer between 1 and 200 (including 1 and 200)	50
walkLength	No	Random walk length	Integer	An integer between 1 and 100 (including 1 and 100)	40
walkNumber	No	Number of random walk steps of each vertex.	Integer	An integer between 1 and 100 (including 1 and 100)	10
iterations	No	Number of iterations	Integer	An integer between 1 and 100 (including 1 and 100)	10

Table 7-181 response_data parameter description

Parameter	Type	Description
embedding	List	Vector representation of each vertex mapped to the Euclidean space. The format is as follows: [{vertexId:vectorValue}] where vertexId is of the string type. vectorValue : is a euclidean vector, for example, [-0.485, -0.679, 0.356].

7.1.6.2.23 Real-time Recommendation (2.2.21)

Table 7-182 parameters parameter descriptions

Parameter	Mandatory	Description	Type	Value Range	Default Value
sources	Yes	Vertex ID. Multiple vertices are supported.	String	The number of source vertices cannot exceed the upper limit (30 by default, which can be changed using source_limit) . Use commas (,) to separate the IDs.	-
alpha	No	Weight coefficient. A larger value indicates a longer step.	Double	A real number between 0 and 1 (excluding 0 and 1)	0.85
N	No	Total number of walk steps	Integer	1 to 200000	10000

Parameter	Mandatory	Description	Type	Value Range	Default Value
nv	No	Parameter indicating that the walk process ends ahead of schedule: minimum number of access times of a potential recommended vertex NOTE If a vertex is accessed during random walk and the number of access times reaches nv , the vertex will be recorded as the potential recommended vertex.	Integer	1 to 10	5
np	No	Parameter indicating that the walk process ends ahead of schedule: number of potential recommended vertices NOTE If the number of potential recommended vertices of a source vertex reaches np , the random walk for the source vertex ends ahead of schedule.	Integer	1 to 2000	1000
label	No	Expected type of the vertex to be output. NOTE - Expected type of the vertex to be output. If the value is null, the original calculation result of the algorithm is output without considering the vertex type. - If the value is not null, vertices with the label are filtered from the calculation result.	String	Vertex label	-
directed	No	Whether to consider the edge direction	Boolean	true or false	true
source_limit	No	Maximum number of source vertices	Int	1~100000	30

Parameter	Mandatory	Description	Type	Value Range	Default Value
restricted	No	Whether to accept invalid source vertices restricted=true : If a vertex that does not exist in the graph is passed to sources , an error is reported. restricted=false : A vertex that does not exist in the graph can be passed to sources . However, if all source vertices do not exist, an error is reported.	Bool	true or false	true

Table 7-183 response_data parameter descriptions

Parameter	Type	Description
score	List	Score of each vertex, which reflects the recommendation degree. A larger value indicates a higher recommendation degree. The format is as follows: [{vertexId: scoreValue} ,...] where vertexId is of the string type. scoreValue is of the double type.
sources	List	Array of source vertex IDs

7.1.6.2.24 Degree Correlation (1.0.0)

NOTE

This algorithm can run without specifying its **parameters**.

Table 7-184 response_data parameter description

Parameter	Type	Description
degree_correlation	Double	Degree correlation

7.1.6.2.25 Triangle Count (1.0.0)

Table 7-185 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range
statistics	No	Whether to export only the total statistical result. • true: Export only the statistical result. • false: Export the number of triangles corresponding to each vertex.	Boolean	true or false . The default value is true .

Table 7-186 response_data parameter description

Parameter	Type	Description
triangle_count	Integer	Number of triangles
vertex_triangles	List	Number of triangles on each vertex. The format is as follows: [{vertexId : vertexTriangleCount},...], where vertexId is of the string type. vertexTriangleCount is of the integer type.

7.1.6.2.26 Cluster Coefficient (1.0.0)

 NOTE

This algorithm can run without specifying its **parameters**.

Table 7-187 response_data parameter description

Parameter	Type	Description
cluster_coefficient	Double	Clustering coefficient

7.1.6.2.27 Closeness Centrality (1.0.0)

Table 7-188 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	ID of the vertex to be calculated	String	-	-

Table 7-189 response_data parameter description

Parameter	Type	Description
closeness	Double	Closeness centrality degree
source	String	ID of the vertex to be calculated

7.1.6.2.28 Betweenness Centrality (2.2.4)

Table 7-190 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
directed	No	Whether an edge is directed	Boolean	The value can be true or false	true
weight	No	Weight of an edge	String	The value can be an empty string. If this parameter is left blank, the weight and distance of this edge are 1 by default. You can set this parameter to a property of the edge, and the property value will be the weight. If the edge does not have the specified property, the weight is 1 by default. NOTE The weight of an edge must be greater than 0 .	-

Parameter	Mandatory	Description	Type	Value Range	Default Value
seeds	No	Vertex ID	String	If the graph is large, precise betweenness calculation can be slow. You can set seeds to the sampling vertices for approximate calculation. The more seeds vertices, the closer results to the precise calculation. The number of vertices cannot be greater than 100,000.	-
k	No	Number of samples	Integer	If the graph is large, precise betweenness calculation can be slow. You can set k to randomly select k sampling vertices from the graph. The larger the value, the closer results to the precise calculation. The value cannot be greater than 100,000.	-

Table 7-191 response_data parameter description

Parameter	Type	Description
betweenness	List	Betweenness value of each node. The format is [{vertexId: betweennessValue}, ...]. • vertexId is of the string type. • betweennessValue is of the double type.

7.1.6.2.29 Edge Betweenness Centrality (2.2.4)

Table 7-192 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
directed	No	Whether an edge is directed	Boolean	The value can be true or false	true
weight	No	Weight of an edge	String	The value can be an empty string. If this parameter is left blank, the weight and distance of this edge are 1 by default. You can set this parameter to a property of the edge, and the property value will be the weight. If the edge does not have the specified property, the weight is 1 by default. NOTE The weight of an edge must be greater than 0 .	-
seeds	No	Vertex ID	String	If the graph is large, precise betweenness calculation can be slow. You can set seeds to the sampling vertices for approximate calculation. The more seeds vertices, the closer results to the precise calculation. The number of vertices cannot be greater than 100,000.	-
k	No	Number of samples	Integer	If the graph is large, precise betweenness calculation can be slow. You can set k to randomly select k sampling vertices from the graph. The bigger value, the closer results to the precise calculation. The value cannot be greater than 100,000.	-

Table 7-193 response_data parameter description

Parameter	Type	Description
edge_betweenness	List	Betweenness value of each edge. The format is [{"source":sourceId,"target":targetId,"index":indexValue,"betweenness":betweennessValue}, ...]. betweennessValue is of the double type.

7.1.6.2.30 Origin-Destination Betweenness Centrality (2.2.4)

Table 7-194 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
directed	No	Whether an edge is directed	Boolean	The value can be true or false	true
weight	No	Weight of an edge	String	The value can be an empty string. If this parameter is left blank, the weight and distance of this edge are 1 by default. You can set this parameter to a property of the edge, and the property value will be the weight. If the edge does not have the specified property, the weight is 1 by default. NOTE The weight of an edge must be greater than 0 .	-
OD	No	Start vertex and end vertex pair (origin-destination)	String	The value must be in the standard CSV format. The start vertex (origin) and end vertex (destination) are separated by commas (,), and the start and end vertex pairs are separated by newline characters (\n), for example, Alice,Nana\nLily,Amy .	-

Parameter	Mandatory	Description	Type	Value Range	Default Value
seeds	No	ID of the hot spot vertex.	String	If OD is not set, the seeds data will be used.	-
modes	No	Hot spot vertex type.	String	• IN: The hot spot vertex ID is used as the start vertex ID. • OUT: The hot spot vertex ID is used as the end vertex ID.	-
capacity	No	Number of participants at each hot spot in seeds .	Integer	-	-

 NOTE

Either **OD_pairs** or **seeds** must be specified. If they are both specified, the **OD_pair** value prevails.

Table 7-195 response_data parameter description

Parameter	Type	Description
betweenness	List	Betweenness value of each edge. The format is [{"source":sourceId,"target":targetId,"index":indexValue,"betweenness":betweennessValue}, ...]. • betweennessValue is of the double type.

7.1.6.2.31 Circle Detection with a Single Vertex (2.2.4)

Table 7-196 Parameters

Parameter	Mandatory	Description	Type	Value Range Default Value
source	Yes	ID of the vertex	String	-
min_circle_length	No	Minimum circle length	Integer	[3,15] 3

Parameter	Mandatory	Description	Type	Value Range Default Value
max_circle_length	No	Maximum circle length. The value must be bigger than min_circle_length .	Integer	[3,15] 10
limit_circle_number	No	Maximum number of circles you want to search for	Integer	[1,100000] 100

Table 7-197 response_data parameter

Parameter	Type	Description
circles	List	A set of circles that contain the given vertex. Format: [[circle1],[circle2], ...]. The circle format is as follows: [vertexId,...]. vertexId is of the string type.
source	String	ID of the given vertex
circle_number	Integer	Number of circles found
runtime	Double	Algorithm running time
min_circle_length	Integer	Minimum circle length
max_circle_length	Integer	Maximum circle length
limit_circle_number	Integer	Maximum number of circles you want to search for

7.1.6.2.32 Filtered Circle Detection (2.2.15)

Request Parameters

Table 7-198 parameters parameter descriptions

Parameter	Mandatory	Description	Type	Value Range	Default Value
sources	No	Set of source vertex IDs to be queried	String	-	The value is in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana .
n	No	Upper limit of the number of enumerated circles that meet the filter criteria	Integer	[1,100000]	100
statistics	No	Whether to export the number of circles that meet the filter criteria	Boolean	true or false	false
batch_number	No	Number of source vertices for batch processing	Integer	[1,1000]	10
output_format	No	Output format	String	vertexId , edgeId , or edgeObject	edgeObject
filters	Yes	Filter criteria. Each element in the array corresponds to a filter.	Object	-	-

Table 7-199 filters element formats

Parameter	Mandatory	Description	Type	Value Range	Default Value
operator	No	Direction of the query to be performed at the current layer	String	out, in, or both	out
edge_filter	No	Filter criteria for the current layer. For details, see Table 7-273 in the Filtered-query API.	Object	-	-
vertex_filter	No	Filter criteria of vertices at the current layer. For details, see Table 7-273 in the Filtered-query API.	Object	-	-
times	No	Number of layers queried using the same filter criteria	Integer	[1,10]	1

 NOTE

- Filter criteria at the first layer are used to filter source vertices. Therefore, only the **vertex_filter** parameter is valid.
- Filter criteria at the last layer are used to filter source vertices.
- The circle length ranges from 3 to 10. Therefore, the number of filtering layers is 4 to 11.

Table 7-200 response_data parameter descriptions

Parameter	Type	Description
circles	List	Set of circles found. The format is [[circle1], [circle2], ...] . The circle format is as follows: • If output_format is edgeObject, the format is [{"source": sourceId, "target": targetId, "index": edgeIndex}, ...], where sourceId, targetId, and edgeIndex are of the string type. • If output_format is edgId, the format is [sourceId-targetId-edgeIndex,...], where sourceId-targetId-edgeIndex is of the string type. • If output_format is vertexId, the format is [vertexId, ...], where vertexId is of the string type.
runtime	Double	Algorithm running time
n	Integer	Maximum number of enumerated circles
circle_number	Integer	When statistics is set to true , the number of circles that meet filter criteria is displayed.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm
{
  "algorithmName": "filtered_circle_detection",
  "parameters": {
    "n": 10,
    "statistics": true,
    "output_format": "edgId"
  },
  "filters": [
    {
    },
    {
      "operator": "out",
      "edge_filter": {
        "property_filter": {
          "leftvalue": {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "transfer"
          }
        }
      }
    }
  ],
  "times": 5
}
```

7.1.6.2.33 Subgraph Matching (2.2.16)

Table 7-201 Parameter description

Parameter	Mandatory	Description	Type	Value Range
edges	Yes	Edge set of the subgraph to be matched. The vertex ID must be of the size_t type.	String	The value is in standard CSV format. The source and target vertices of an edge are separated by a comma (,), and edges are separated by a newline character (\n). For example, 1,2\n2,3 .
vertices	Yes	Label of each vertex on the subgraph to be matched.	String	The value is in standard CSV format. Vertices and their labels are separated by commas (,), and labels are separated by newline characters (\n). For example, 1,BP\n2,FBP\n3,CP .
directed	No	Whether to consider the direction of the graph	Boolean	The value can be true or false . The default value is true .
n	No	Maximum number of subgraphs to be searched for	Integer	The value range is [1,100000]. The default value is 100 .
batch_number	No	Number of queries processed in batches each time	Integer	The value range is [1,1000000]. The default value is 10000 .
statistics	No	Whether to display the number of all subgraphs that meet the conditions	Boolean	The value can be true or false . The default value is false .

Table 7-202 response_data parameter description

Parameter	Type	Description
subgraphs	List	Subgraphs with the same pattern of the pattern_graph . The value is in the [[subgraph1], [subgraph2], ...] format. Each subgraph is in the [vertex1,vertex2, ...] format, where vertex is of the string type. The vertices of each subgraph correspond to those of pattern_graph .
pattern_graph	List	Graph pattern. The value is in the [vertex1,vertex2, ...] format, where vertex is of the string type.
subgraph_number	Integer	Number of matched graphs. When statistics is set to true , the total number of graphs that meet query conditions is displayed.

7.1.6.2.34 Topicrank (2.2.20)

Table 7-203 parameters parameter descriptions

Parameter	Mandatory	Description	Type	Value Range	Default Value
sources	Yes	Vertex ID. You can specify multiple node IDs in CSV format and separate them with commas (,).	String	Currently, a maximum of 100000 IDs are allowed.	-
actived_p	No	Initial weight of the sources vertex.	Double	The value ranges from 0 to 100000.	1
default_p	No	Initial weight of a non-source vertex	Double	The value ranges from 0 to 100000.	1
filtered	No	Whether to filter results	Boolean	The value can be true or false .	false
only_neighbors	No	Whether to display only the neighboring vertices of the sources	Boolean	The value can be true or false .	false

Parameter	Mandatory	Description	Type	Value Range	Default Value
alpha	No	Weight coefficient	String	The value is a real number between 0 and 1.	0.85
convergence	No	Convergence	String	The value is a real number between 0 and 1.	0.00001
max_iterations	No	Maximum iterations	Integer	An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000].	1000
directed	No	Whether the edges are directed	Boolean	The value can be true or false .	true
num_thread	No	Number of threads	Integer	The value ranges from 1 to 40.	4

Table 7-204 response_data parameter descriptions

Parameter	Type	Description
topicrank	List	TopicRank value of each vertex. The format is as follows: [{vertexId:rankValue},...] where vertexId is of the string type. rankValue is of the double type.

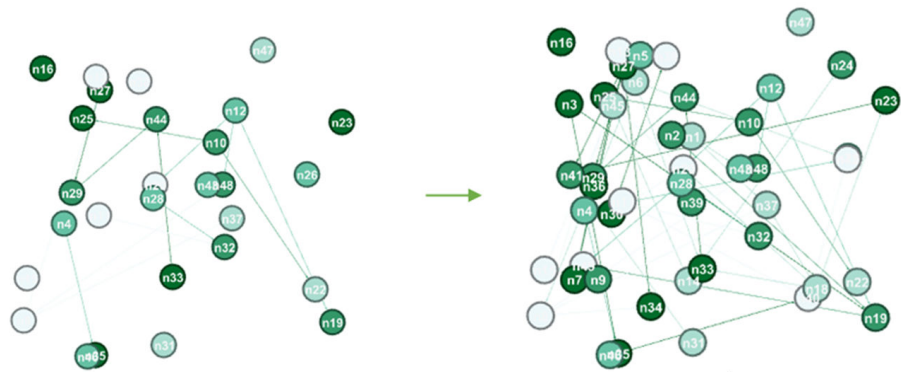
7.1.7 Temporal Graph APIs

7.1.7.1 Community Evolution (temporal_graph)

Function

The community evolution algorithm generates a temporal graph that shows structure changes of a community over time.

Figure 7-1 Principle



URL

POST /ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?
action_id=execute-analysis

Table 7-205 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-206 Request body parameter descriptions

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name. The value is temporal_graph .
dynamicRange	Yes	Object	Temporal parameters
parameters	Yes	String	Algorithm parameters

Table 7-207 dynamicRange parameter descriptions

Parameter	Mandatory	Type	Description
start	Yes	Date or integer	Start time of the temporal analysis. The start time must be earlier than the end time.
end	Yes	Date or integer	End time of the temporal analysis
time_props	Yes	Object	Time properties for the temporal analysis

Table 7-208 time_props parameter descriptions

Parameter	Mandatory	Type	Description
stime	Yes	String	Property name of the start time
etime	Yes	String	Property name of the end time

Table 7-209 parameters parameter descriptions

Parameter	Mandatory	Type	Description
source	Yes	String	Vertices in the community network. You can specify a maximum of 100,000 vertices.
temporal_vertex	No	Boolean	Whether to perform community evolution on a node. The default value is false .

Response Parameters

Table 7-210 Response parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.

Parameter	Type	Description
data	Json	Result details. For details, see Table 7-211 .

Table 7-211 data parameter descriptions

Parameter	Type	Description
vertices	List	Vertex set contained in the result.
edges	List	Edge set contained in the result.

Example Request

Observe the community evolution of some nodes. The algorithm name is **temporal_graph**, the start time of dynamic analysis is **`\${startTime}`**, and the end time is **`\${endTime}`**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?
action_id=execute-analysis
{
  "algorithmName": "temporal_graph",
  "dynamicRange": {
    "start": "${startTime}",
    "end": "${endTime}",
    "time_props": { "stime": "${property(start_time)}", "etime": "${property(start_time)}" }
  },
  "parameters": {
    "sources": [],
    "temporal_vertex": false
  }
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "data": {
    "vertices": [
      {
        "id": "Place00032",
        "label": "Place",
        "properties": {}
      }, {
        "id": "Person00041",
        "label": "Person",
        "properties": {
          "gender": ["Female"],
          "phone": ["P0334529194"],
          "name": ["Wu Guifang"],
          "startTime": [1774368],
          "endTime": [16756560],
          "age": [48]
        }
      }
    ],
    "edges": [

```

```
{
  "index": "0",
  "source": "Person00041",
  "label": "hasVisit",
  "properties": {
    "startTime": [1646092800],
    "visitDate": [20220301],
    "endTime": [1646126769]
  },
  "target": "Place00032"
}, {
  "index": "3",
  "source": "Person00041",
  "label": "hasVisit",
  "properties": {
    "startTime": [1646168289],
    "visitDate": [20220301],
    "endTime": [1646179199]
  },
  "target": "Place00032"
}]
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Dynamic Analysis is not supported in this instance. Please build dynamic graph instance",
  "errorCode": "GES.9001"
}
```

Status Codes

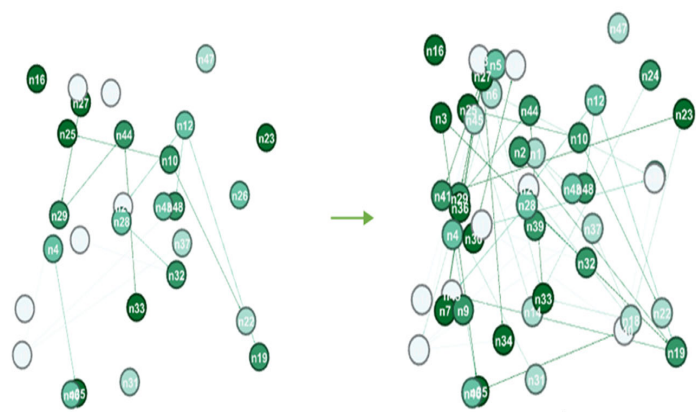
Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

7.1.7.2 Temporal BFS (temporal_bfs)

Function

This algorithm searches for associated vertices based on temporal message passing and temporal BFS algorithms, and outputs the visit time of each vertex and the distance from the vertex to the source start vertex.

Figure 7-2 Principle



URL

POST /ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?
action_id=execute-analysis

Table 7-212 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-213 Request body parameter descriptions

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name. The value is temporal_bfs .
dynamicRange	Yes	String	Temporal parameters
parameters	Yes	String	Algorithm parameters

Table 7-214 dynamicRange parameter descriptions

Parameter	Mandatory	Type	Description
start	Yes	Date or integer	Start time of the temporal analysis

Parameter	Mandatory	Type	Description
end	Yes	Date or integer	End time of the temporal analysis
time_props	Yes	Object	Time properties for the temporal analysis

Table 7-215 time_props parameter descriptions

Parameter	Mandatory	Type	Description
stime	Yes	String	Property name of the start time
etime	Yes	String	Property name of the end time

Table 7-216 parameters parameter descriptions

Parameter	Mandatory	Type	Description
source	Yes	String	Start vertex ID
k	No	Integer	Distance from the target vertices to the start vertex. The value range is [1, 100]. The default value is 3.
directed	No	Boolean	Search direction. The value can be true or false . The default value is true .

Response Parameters

Table 7-217 Response parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
data	Json	Temporal BFS result details.

Example Request

Specify a source vertex ID to search for associated vertices. The algorithm name is **temporal_bfs**, the start time of dynamic analysis is **`\${startTime}`**, and the end time is **`\${endTime}`**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?
action_id=execute-analysis
{
  "algorithmName": "temporal_bfs",
  "dynamicRange": {
    "start": "${startTime}",
    "end": "${endTime}",
    "time_props": { "stime": "${property(start_time)}", "etime": "${property(start_time)}" }
  },
  "parameters": {
    "source": ""
  }
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "data": {
    "outputs": {
      "data_return_size": 2,
      "runtime": 0.000079,
      "data_offset": 0,
      "data_total_size": 2,
      "temporal_bfs": [{
        "Person00041": {
          "arrive": 1646092800,
          "dist": 0,
          "predecessor": ""
        }
      }, {
        "Place00001": {
          "arrive": 1648306984,
          "dist": 1,
          "predecessor": "Person00041"
        }
      }
    ]
  }
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "${errorMessage}",
  "errorCode": "GES.8301"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

7.1.7.3 Temporal Paths

Function

This API is used to execute the temporal paths algorithm based on input parameters.

Note: Only one temporal path that meets the conditions is returned between two vertices.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?
action_id=execute-analysis

Table 7-218 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-219 Request body parameter descriptions

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name. The value is temporal_paths .

Parameter	Mandatory	Type	Description
parameters	Yes	parameters Object	Algorithm parameters. For details, see the parameter descriptions of each algorithm.
dynamicRange	Yes	dynamicRange Object	Temporal parameters

Table 7-220 parameters

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex ID
targets	Yes	String	Target vertex ID set. The value is in CSV format. IDs are separated by commas (,), for example, Alice,Nana . The maximum number of IDs is 100,000.
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is false .
k	No	Integer	Maximum depth. The value ranges from 1 to 100. The default value is 3 .
strategy	No	String	Algorithm policy. The value can be shortest , foremost , or fastest . (Note: fastest is not supported currently.) The default value is shortest . • shortest: Runs the shortest temporal paths algorithm to return the temporal path with the shortest distance. • foremost: Runs the foremost temporal paths algorithm to return the temporal path that reaches the target node as early as possible. • fastest: Runs the fastest temporal paths algorithm to return the temporal path that takes the shortest time.

Table 7-221 dynamicRange

Parameter	Mandatory	Type	Description
start	Yes	Date/ Integer	Start time for temporal analysis
end	Yes	Date/ Integer	End time for temporal analysis
time_props	Yes	time_props Object	Time properties for temporal analysis

Table 7-222 time_props

Parameter	Mandatory	Type	Description
stime	Yes	String	Name of the start time property
etime	Yes	String	Name of the end time property

Response Parameters

Table 7-223 Response parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
data	Json	Details about temporal path analysis results.

Example Request

Specify a source vertex ID to search for associated vertices. The algorithm name is **temporal_paths**, the start time of dynamic analysis is **1646092800**, the end time is **1646170716**, and the source vertex ID is **Person00014**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/dynamicgraphs/action?
action_id=execute-analysis
{
  "algorithmName":"temporal_paths",
  "dynamicRange":{
    "start":1646092800,
    "end":1646170716,
    "time_props":
      {"stime":"startTime","etime":"endTime"}
```



```
    },
    "parameters":{
      "source":" Person00014",

      "targets":"Person00055,Person00058,Person00052,Person00061,Person00060,Place00032,Place00016,Place00026,Place00015,Place00043",
      "strategy":"shortest",
      "directed":true
    }
  }
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "data": {
    "outputs": {
      "data_return_size": 1,
      "direct": 1,
      "runtime": 0.00011,
      "temporal_paths": [
        [{
          "Person00014": {
            "arrive": 1646092800,
            "dist": 0,
            "predecessor": ""
          }
        }, {
          "Place00016": {
            "arrive": 1647169795,
            "dist": 1,
            "predecessor": "Person00014"
          }
        }
      ]
    },
    "data_offset": 0,
    "data_total_size": 1
  }
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.

Return Value	Description
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

7.1.8 Path APIs

7.1.8.1 Querying Path Details (1.1.6)

Function

This API is used to query the path details. All possible paths will be listed.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/paths/action?action_id=query-detail

Table 7-224 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-225 Request body parameter descriptions

Parameter	Mandatory	Type	Description
paths	Yes	List	Set of paths to be queried
directed	No	Boolean	Whether the querying path is directional or non-directional: • true: directed • false: undirected The default value is false .

Response Parameters

Table 7-226 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	Object	Query results. This parameter is left blank when the query fails.

Table 7-227 data parameter descriptions

Parameter	Type	Description
outputs	Object	Query results containing the paths
paths	List	Collection of paths that contain detailed vertex and edge information, in JSONArray format NOTE In the returned paths: - If the vertex does not exist, the corresponding position is {}. - If there is no edge between vertices, the corresponding position is {"edges": []}.

Example Request

Query path details. The paths to be queried are **Ray**, **Lethal Weapon**, and **Alice**. Do not set the query direction.

```
post http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/paths/action?action_id=query-detail
{
  "paths":[
    [
      "Ray",
      "Lethal Weapon",
      "Alice"
    ]
  ]
}
```

```
    ],  
    "directed":false  
  }  
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{  
  "data": {  
    "outputs": {  
      "paths": [  
        {  
          "id": "Ray",  
          "label": "user",  
          "properties": {  
            "Name": ["Ray"],  
            "Gender": ["M"],  
            "Age": ["18-24"],  
            "Occupation": ["college/grad student"],  
            "Zip-code": ["90241"]  
          }  
        },  
        {  
          "edges": [  
            {  
              "source": "Ray",  
              "target": "Lethal Weapon",  
              "index": "1",  
              "label": "rate",  
              "properties": {  
                "Score": [2],  
                "Datetime": ["2000-11-22 19:16:16"]  
              }  
            }  
          ]  
        },  
        {  
          "id": "Alice",  
          "label": "user",  
          "properties": {  
            "Name": ["Alice"],  
            "Gender": ["F"],  
            "Age": ["25-34"],  
            "Occupation": ["academic/educator"],  
            "Zip-code": ["79928"]  
          }  
        }  
      ]  
    }  
  }  
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
```

```
"errorMessage": "Not found. Please check the input parameters.",  
"errorCode": "GES.8000"  
}
```

Status Code

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.9 Graph Statistics APIs

7.1.9.1 Querying General Information About a Graph (1.0.0)

Function

This API is used to query the general information about a graph, such as the numbers of vertices and edges.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/summary?
label_details={labelDetails}

Table 7-228 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Parameter	Mandatory	Type	Description
label_details	No	Boolean	Whether to return the number of vertices and edges under each label. The default value is false . If this parameter is set to true , the numbers of vertices and edges under each label are returned.

Request Parameters

None

Response Parameters

Table 7-229 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	Object	Query results. This parameter is left blank when the request fails.

Table 7-230 data parameter descriptions

Parameter	Type	Description
vertexNum	Integer	Number of vertices in a graph
edgeNum	Integer	Number of edges in a graph

Parameter	Type	Description
labelDetails (2.2.14)	Object	Numbers of vertices and edges under each label. To properly display this parameter, create vertex and edge indexes according to Table 7-231 .

Table 7-231 Description of each element in **labelDetails** when the execution is successful

Parameter	Type	Description
labelInVertex	Object	Number of vertices under each label. If the number of vertices under a label is 0, the label is not displayed. To include this parameter in the response, create an index by referring to Creating an Index (1.1.6). During index creation, set indexType to GlobalCompositeVertexIndex, set hasLabel to true, and leave indexProperty blank.
labelInEdge	Object	Number of edges under different labels. If the number of edges under a label is 0, the label is not displayed. To include this parameter in the response, create an index by referring to Creating an Index (1.1.6). During index creation, set indexType to GlobalCompositeEdgeIndex, set hasLabel to true, and leave indexProperty blank.
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.

Example Request

Query general information about a graph, such as the numbers of vertices and edges. The value **true** indicates that the numbers of vertices and edges of different labels are returned.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/summary?label_details=true
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

- Response example 1 of a successful request (The numbers of vertices and edges under each label are returned.)

Http Status Code: 200

```
{
  "data": {
    "vertexNum": 146,
    "labelDetails": {
      "labelInVertex": {
        "movie": 46,
        "user": 100
      },
      "labelInEdge": {
        "default": 450,
        "rate": 1209
      }
    },
    "edgeNum": 1659
  }
}
```

- Response example 2 of a successful request (The numbers of vertices and edges under each label fail to be returned.)

Http Status Code: 200

```
{
  "data": {
    "vertexNum": 146,
    "labelDetails": {
      "errorMessage": "Label index in vertices is not found.Label index in edges is not found.",
      "errorCode": "GES.8017"
    },
    "edgeNum": 1659
  }
}
```

- Example response 3 of a successful request (Only the number of vertices under each label is returned.)

Http Status Code: 200

```
{
  "data": {
    "vertexNum": 146,
    "labelDetails": {
      "errorMessage": "Label index in edges is not found.",
      "labelInVertex": {
        "movie": 46,
        "user": 100
      },
      "errorCode": "GES.8017"
    },
    "edgeNum": 1659
  }
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Not found. Please check the input parameters.",
}
```



```
}
  "errorCode": "GES.8000"
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.9.2 Querying the Graph Version (2.0.0)

Function

This API is used to query the graph version.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/version

Table 7-232 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-233 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
version	String	Query results. This parameter is left blank when the request fails.

Example Request

Query the graph version.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/version
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "version": "2.0.0"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 404
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Code

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.10 Graph Operation APIs

7.1.10.1 Importing a Graph (2.1.14)

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-graph

Table 7-234 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-235 Request body parameter descriptions

Parameter	Mandatory	Type	Description
edgesetPath	No	String	Set edgesetPath , vertexsetPath , or schemaPath . Edge file directory or name

Parameter	Mandatory	Type	Description
edgesetFormat	No	String	Edge dataset format. Currently, the options are csv and txt . The default value is csv .
vertexsetPath	No	String	Set edgesetPath , vertexsetPath , or schemaPath . Vertex file directory or name
vertexsetFormat	No	String	Vertex dataset format. Currently, the options are csv and txt . The default value is csv .
schemaPath	No	String	Set edgesetPath , vertexsetPath , or schemaPath . Metadata file of the new data
logDir	No	String	Directory for storing logs of imported graphs. This directory stores the data that fails to be imported during graph creation and detailed error causes.
parallelEdge	No	Object	How to process repetitive edges.
action	No	String	Processing mode of repetitive edges. The value can be allow , ignore , or override . The default value is allow . • allow indicates that repetitive edges are allowed. • ignore indicates that subsequent repetitive edges are ignored. • override indicates that the previous repetitive edges are overwritten.

Parameter	Mandatory	Type	Description
ignoreLabel	No	Boolean	Whether to ignore labels on repetitive edges. The value can be true or false. The default value is true. • true: Indicates that the repetitive edge definition does not contain the label. That is, the <source vertex, target vertex> indicates an edge, excluding the label information. • false: Indicates that the repetitive edge definition contains the label. That is, the <source vertex, target vertex, label> indicates an edge.
delimiter	No	Character	Field separator in a CSV file. The default value is comma (,). The default element separator in a field of the list/set type is semicolon (;).
trimQuote	No	Character	Field quote character in a CSV file. The default value is double quotation marks ("). They are used to enclose a field if the field contains separators or line breaks.
offline	No	Boolean	Whether offline import is selected. The value can be true or false. The default value is false. • true: Offline import is selected. The import speed is high, but the graph is locked and cannot be read or written during the import. • false: Online import is selected. Compared with offline import, online import is slower. However, the graph can be read (cannot be written) during the import.
obsParameters	Yes	Object	OBS parameters

Table 7-236 obsParameters parameters

Parameter	Mandatory	Type	Description
accessKey	Yes	String	User AK
secretKey	Yes	String	User SK

Response Parameters

Table 7-237 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	Integer	Job type. This parameter is left blank if the request fails.

Example Request

Import a graph. The edge file directory is **testbucket/demo_movie/edges/**, and the edge dataset format is CSV. The vertex file directory is **testbucket/demo_movie/vertices/**, and the vertex dataset format is CSV. The OBS path of the metadata file of the new data is **testbucket/demo_movie/incremental_data_schema.xml**, and the log storage directory is **testbucket/importlogdir**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-graph
{
  "edgesetPath": "testbucket/demo_movie/edges/",
  "edgesetFormat": "csv",
  "vertexsetPath": "testbucket/demo_movie/vertices/",
  "vertexsetFormat": "csv",
  "schemaPath": "testbucket/demo_movie/incremental_data_schema.xml",
  "logDir": "testbucket/importlogdir",
}
```

```
"parallelEdge": {
  "action": "override",
  "ignoreLabel": true
},
"delimiter": " ",
"trimQuote": "\"",
"offline": true,
"obsParameters": {
  "accessKey": "xxxxxx",
  "secretKey": "xxxxxx"
}
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "b7f3ea8d-8a22-40ea-8751-6bcad5fc38b6000003836",
  "jobType": 4
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.10.2 Exporting a Graph (1.0.5)

 NOTE

If you choose to export CSV files to your local host, the files are opened using the spreadsheet software by default. You are advised to open the files in a text editor. If the data contains special characters such as plus signs (+), minus signs (-), equal signs (=), and at signs (@), the data will be parsed into formulas by the software. To ensure system security, pay attention to the following when opening such files:

1. Do not select **Enable Dynamic Data Exchange Server Launch (not recommended)**.
2. Do not select **Enable** or **Yes** if a dialog box indicating a security issue is displayed.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph

Table 7-238 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-239 Request body parameter descriptions

Parameter	Mandatory	Type	Description
graphExportPath	Yes	String	OBS path to which a graph is exported
edgeSetName	Yes	String	Name of the folder where the edges are exported
vertexSetName	Yes	String	Name of the folder where the vertices are exported
schemaName	Yes	String	Name of the exported metadata file
paginate(2.3.11)	No	Object	Pagination-related parameters. In version 2.3.11 or later, graphs are exported on multiple pages by default.
obsParameters	Yes	String	OBS parameters
accessKey	Yes	String	User AK

Parameter	Mandatory	Type	Description
secretKey	Yes	String	User SK

Table 7-240 Elements of the paginate parameter

Parameter	Mandatory	Type	Description
enable	No	Bool	Whether to enable pagination. The default value is true . If pagination is not required, set this parameter to false .
rowCountPerFile	No	Int	Maximum number of rows in each file when graphs are exported by page. The default maximum value is 10000000 .
numThread	No	Int	Number of concurrent threads when graphs are exported by page. The default value is 8 .
maxSizePerFile	No	Int	Maximum size of each file when graphs are exported by page, in bytes. By default, the size cannot exceed the maximum size of the file that was imported.

Response Parameters

Table 7-241 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.

Parameter	Type	Description
jobId	String	ID of an asynchronous job You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	Integer	Job type. This parameter is left blank if the request fails.

Example Request

- Example request 1: Export a graph. The OBS path for exporting the graph is **demo_movie/**, the name of the folder where the edges are exported is **set_edge**, the name of the folder where the vertices are exported is **set_vertex**, and the name of the exported metadata file is **set_schema.xml**.
POST `http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph`

```
{
  "graphExportPath": "demo_movie/",
  "edgeSetName": "set_edge",
  "vertexSetName": "set_vertex",
  "schemaName": "set_schema.xml",
  "paginate": {
    "numThread": 16,
    "rowCountPerFile": 1000000
  },
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  }
}
```
- Example request 2
POST `http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph`

```
{
  "graphExportPath": "demo_movie/",
  "edgeSetName": "set_edge",
  "vertexSetName": "set_vertex",
  "schemaName": "set_schema.xml",
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  }
}
```

Example Response

Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "9874990c-b2c5-4e8b-96f3-10a9c46fe3a6000003836",
  "jobType": 1
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
```

```
"errorMessage": "Not found. Please check the input parameters.",  
"errorCode": "GES.8000"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.10.3 Clearing a Graph (2.1.1)

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=clear-graph

Table 7-242 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-243 Request body parameter descriptions

Parameter	Mandatory	Type	Description
clearMetadata	No	Boolean	Whether to clear schema data. The default value is false .

Response Parameters

Table 7-244 Response body parameter descriptions

Parameter	Mandatory	Description
errorMessage	No	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	No	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	No	ID of an asynchronous job You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	Integer	Job type. This parameter is left blank if the request fails.

Example Request

Clear a graph by deleting its schema data.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=clear-graph
{
  "clearMetadata": true
}
```

Example Response

Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "c332e1d4-844c-4f75-beeb-e31d2c57f118000003836",
  "jobType": 3
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
}
```

```
"errorCode": "GES.8000"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.11 Subgraph Operation APIs

7.1.11.1 Querying a Subgraph (2.1.13)

Function

This API is used to query the subgraphs formed by the entered vertices and edges between the vertices.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/subgraphs/action?
action_id=query

Table 7-245 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-246 Request body parameter

Parameter	Mandatory	Type	Description
vertices	Yes	String	IDs of the vertices contained in the subgraph NOTE You can enter a maximum of 100,000 vertices. When this limit is exceeded, an error occurs.

Response Parameters

Table 7-247 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
data	Object	The data field is contained when the query is successful, and the data field contains the subgraph query result. NOTE The maximum number of subgraph edges that can be returned is 100,000. If the number of edges exceeds this limit, an error is reported.

Example Request

Query the subgraphs formed by the entered vertices and edges between the vertices. The subgraph vertex IDs are **Ray**, **Ella**, and **Lethal Weapon**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/subgraphs/action??action_id=query
{
  "vertices": [
    "Ray",
    "Ella",
    "Lethal Weapon"
  ]
}
```

Example Response

Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "vertices": [
      {
        "id": "Ray",
        "label": "user",
        "properties": {
          "Name": ["Ray"],
          "Gender": ["M"],
          "Age": ["18-24"],
          "Occupation": ["college/grad student"],
          "Zip-code": ["90241"]
        }
      },
      {
        "id": "Ella",
        "label": "user",
        "properties": {
          "Occupation": ["other or not specified"],
          "Name": ["Era"],
          "Zip-code": ["94402"],
          "Gender": ["F"],
          "Age": ["25-34"]
        }
      }
    ],
    "edges": [
      {
        "source": "Ray",
        "target": "Lethal Weapon",
        "index": "1",
        "label": "rate",
        "properties": {
          "Score": [2],
          "Datetime": ["2000-11-22 19:16:16"]
        }
      },
      {
        "index": "0",
        "source": "Ella",
        "label": "rate",
        "properties": {
          "Score": [5],
          "Datetime": ["2000-11-23 02:30:29"]
        },
        "target": "Lethal Weapon"
      },
      {
        "index": "5",
        "source": "Ella",
        "label": "friends",
        "properties": {},
        "target": "Ray"
      }
    ]
  }
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": " Bad Request, parameter vertices cannot be null",
  "errorCode": "GES.8214"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.11.2 Executing an Algorithm on a Subgraph

Introduction

This API is used to adjust the subgraph creation type based on the input and executes an algorithm on the generated subgraph.

URL

POST /ges/v1.0/{project_id}/graphs/{graph_name}/subgraphs/action?
action_id=execute-algorithm

Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-248 Request body parameter descriptions

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name Options: • connected_component • kcore
parameters	Yes	JSON	Algorithm parameters For details, see: • connected_component • Kcore
subgraphCreator	Yes	Object	Subgraph parameters For details, see Table 7-249 .

Table 7-249 subgraphCreator parameter descriptions

Parameter	Mandatory	Type	Description
name	No	String	Type of the subgraph creator. Currently, only filtered is available.
parameters	Yes	JSON	The parameter format varies according to the name of the subgraph creator.

Table 7-250 Parameter descriptions when name is set to filtered

Parameter	Mandatory	Type	Description
vertex_filter	No	String	Vertex filtering criteria
edge_filter	No	String	Edge filtering criteria

Response Parameters

Table 7-251 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	Integer	Job type. This parameter is left blank when the request fails.

Example Request

Execute a specific algorithm on a generated subgraph. The algorithm name is **connected_component**, and the subgraph generator type is **filtered**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/subgraphs/action?
action_id=execute-algorithm
{
  "algorithmName":"connected_component",
  "subgraphCreator": {
    "name": "filtered",
    "parameters": {
      "edge_filter": {
        "property_filter": {
          "leftvalue": {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "PHYSICAL_LINK"
          }
        }
      }
    }
  },
  "parameters":{
    "num_thread":4
  }
}
```

Example Response

Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "graph [demo] is not found",
  "errorCode": "GES.8402"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.12 Job Management APIs

7.1.12.1 Querying Job Status on the Service Plane (1.0.0)

Function

This API is used to query the execution status of a job. After asynchronous APIs such as those for querying vertices and edges or executing algorithms are used, job IDs are returned. You can use the job ID to query the execution status of a job.

URI

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?
offset=offset&limit=limit
```

Table 7-252 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
job_id	Yes	String	Job ID
offset	No	Integer	Offset of a query. The default value is 0 .
limit	No	Integer	Maximum number of records that can be queried. The default value is 100000 .

Response Parameters

Table 7-253 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
status	String	Job status when the query is successful. The options are: pending running success partiallyFailed failed This parameter is left blank when the query fails.
data	Object	Algorithm execution result. This parameter is left blank when the query fails.

Table 7-254 data parameter descriptions

Parameter	Type	Description
vertices	List	Vertex-associated algorithm result
edges	List	Edge-associated algorithm result
outputs	Object	Other results
data_return_size	Integer	Number of records returned after a query
data_offset	Integer	Result offset of a query
data_total_size	Integer	Total amount of result data generated by asynchronous jobs

Example Request

Query the execution status of a job. The query offset is **0**, and the maximum number of returned results is **2**.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?offset=0&limit=2
```

NOTE

- **SERVER_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- The results obtained from calling the API vary depending on **job_id**.

Example Response

Status code: 200

Example response (successful request)

```
Http Status Code: 200
{
  "data": {
    "outputs": {
      "data_return_size": 2,
      "vertices": [
        {
          "id": "Sarah",
          "label": "user",
          "properties": {
            "Occupation": [
              "other or not specified"
            ],
            "Name": [
              "Sarah"
            ],
            "Zip-code": [
              "55105"
            ],
            "Gender": [
              "F"
            ],
            "Age": [
              "18-24"
            ]
          }
        }
      ]
    }
  }
}
```

```
    ]
  },
  {
    "id": "Sidney",
    "label": "user",
    "properties": {
      "Occupation": [
        "writer"
      ],
      "Name": [
        "Sidney"
      ],
      "Zip-code": [
        "85296"
      ],
      "Gender": [
        "M"
      ],
      "Age": [
        "18-24"
      ]
    }
  },
  "data_offset": 0,
  "data_total_size": 19
},
"status": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.12.2 Canceling a Job (1.0.0)

Function

This API is used to cancel a job that has been submitted.

NOTE

- Only jobs returned by the following APIs can be canceled: exporting a graph, importing a graph, querying vertices that meet filter criteria, querying edges that meet filter criteria, edge filtering query, [Filtered-query V2](#), executing an algorithm, and adding an index.
- Only jobs created using the following algorithms can be canceled: TopicRank, PageRank, PersonalRank, K-core, Connected Component, Label Propagation, Louvain, and Node2vec.

URI

DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}

Table 7-255 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
job_id	Yes	String	Job ID
graph_name	Yes	String	Graph name

Request Parameters

For details, see the URI parameters.

Response Parameters

Table 7-256 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.

Example Request

Cancel a job that has been submitted.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response (successful request)

```
Http Status Code: 200
{ }
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Can not find job to cancel, jobId is 2c90803d94eea6f80194f31256260005",
  "errorCode": "GES.8303"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.1.12.3 Exporting Job Execution Results to Files (2.2.1)

Function

This API is used to export the execution result (**result**) of an asynchronous job (**jobId**) to a file.

- The following algorithms are supported:
 - PageRank, PersonalRank, and Pixie
 - Louvain, Label Propagation, and Connected Component
 - K-Core
 - SSSP, Shortest Path (including Time Window Shortest Path), Shortest Path of Vertex Sets, All Shortest Paths, and n Paths
 - Triangle Count, Cluster Coefficient, Degree Correlation, and Closeness
 - Link Prediction
 - Betweenness, edge_betweenness, and od_betweenness
- The following queries are supported:
 - [Cypher Queries \(2.2.16\)](#)
 - [Querying Vertices That Meet Filter Criteria \(1.0.0\)](#)
 - [Querying Edges That Meet Filter Criteria \(1.0.0\)](#)

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/action?
action_id=export-result

Table 7-257 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain a project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name.
job_id	Yes	String	Job ID corresponding to the response.

Request Parameters

Table 7-258 Request body parameters

Parameter	Mandator y	Type	Description
exportPath	Yes	String	Export path.
fileName	No	String	Name of the exported file.
obsParame ters	Yes	String	OBS authentication parameters. For details, see Table 7-259 .
paginate	No	Object	Pagination parameter. By default, pagination is disabled for the asynchronous task export API. For details, see Table 7-260 .
erase	No	Boolea n	Whether to delete the original job after the export. The value can be true or false . The default value is true , indicating that the job is deleted and resources are released by default.

Table 7-259 obsParameters parameter descriptions

Parameter	Mandator y	Type	Description
accessKey	Yes	String	Access key ID.
secretKey	Yes	String	Secret access key.

Table 7-260 Elements in paginate

Parameter	Mandator y	Type	Description
enable	No	Boolean	Whether to enable pagination. The default value is false . To enable pagination, set this parameter to true .
rowCountP erFile	No	Integer	Maximum number of rows in each file when execution results are exported by page. The default value is 10000000 .
numThread	No	Integer	Number of concurrent threads when execution results are exported by page. The default value is 8 .

Parameter	Mandator y	Type	Description
maxSizePer File	No	Integer	Maximum size of each file when execution results are exported by page, in bytes.

- paginate parameter descriptions
 - a. When pagination is enabled, **fileName** in the request body indicates the directory name, and the directory is used to store pagination files. When pagination is disabled, **fileName** indicates the file name. Before the export, ensure that the path the file name points to is empty so that the existing data on OBS will not be overwritten during the export.
 - b. If the value of **numThread** is greater than the number of vCPUs used by the GES graph instance, the parameter is set to the number of vCPUs.
 - c. The **rowCountPerFile** value affects the number of actually used threads. That is, when the ratio of the result set size to **rowCountPerFile** is less than **numThread**, the ratio is used as the value of **numThread**.
 - d. If the request is canceled by the user, the data uploaded to OBS will not be deleted. For details about the API for canceling jobs, see [Canceling a Job \(1.0.0\)](#).
- When the pagination function is enabled, the file name is named as follows:

If **enable** in the **paginate** parameter is set to **true**, **fileName** indicates a directory. Files in the directory are named using the combination of the thread ID and file number, and are separated by a period (.). For example, for 3.2 million data records, the examples of the first and last file names in different configurations are as follows:

rowCountPerFile	100,000	100,000	1 million	1 million	5 million
numThread	2	48	2	5	2
Threads Actually Used	2	32	2	4	1
Files Generated by a Single Thread	16	1	2	1	1
First File Name	00.000.txt	00.000.txt	00.00.txt	00.00.txt	00.00.txt
Last File Name	01.015.txt	031.000.txt	01.01.txt	03.00.txt	00.00.txt

Response Parameters

Table 7-261 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job. You can check the job execution status and obtain the return result by querying the job ID. For details, see Querying Job Status on the Service Plane (1.0.0) .
jobType	Integer	Job type. This parameter is left blank if the request fails.

Example Request

- Example request 2: Export the execution results of an asynchronous job to an OBS file. The export path is **demo_movie/**. The name of the exported file is **louvain**. After the export, the original job is deleted.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/action?action_id=export-result
{
  "exportPath": "demo_movie/",
  "fileName": "louvain",
  "erase": true,
  "obsParameters": {
    "accessKey": "xxxx",
    "secretKey": "xxxx"
  }
}
```

- Example request 2: Export the execution result of an asynchronous job to an OBS file. The export path is **demo_movie/**. The name of the exported file is **louvain**. By default, pagination is disabled. When data is exported by page, the maximum number of rows in each file is 100,000.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/action?action_id=export-result
{
  "exportPath": "demo_movie/",
  "fileName": "louvain",
  "paginate": {
    "enable": true,
    "numThread": 2,
    "rowCountPerFile": 100000,
  },
  "obsParameters": {
```

```
"accessKey": "xxxx",  
"secretKey": "xxxx"  
}  
}
```

NOTE

Currently, Cypher statements can only be used to export common value types, such as attribute values, numbers, and strings, but not composite value types (such as lists and maps), vertices, or edges. Example:

- The results of the following statements can be exported:
match (n) return id(n) limit 10
match (n) return n.age, n.occupation
match (n)-[r]->(m) return n.Rating limit 10
unwind [1,2,3] as p return p
- The exported TXT file contains null values or blank lines because the results of the following statements contain objects or compound values:
return [1,2,3], {a:1}
match (n) return n limit 10
match (n)-[r]->(m) return r limit 10

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200  
{  
  "jobId": "2aa37d87-fb5a-44c4-81a1-c3876247d1cf000003836",  
  "jobType": 15  
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400  
{  
  "jobId": "2aa37d87-fb5a-44c4-81a1-c3876247d1cf000003836",  
  "jobType": 15  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

Exporting the Result in TXT File

1. The following is an example of the algorithm execution result, for example, content of **Louvain.txt**:

```
# modularity: 0.4269691347613425,  
#community_num: 4,  
#runtime: 0.003784,  
#data_total_size: 34  
#community:  
1,1  
2,1  
...
```

2. The following is an example Cypher query result:

- Example 1

Query statement:

```
match (n:user)-[r]->(m:movie) return id(n),n.Name, n.Occupation, n.Age,r.Score,m.ChineseTitle
```

Result:

```
#data_total_size:1209  
#data_return_size:1209  
#data_offset:0  
#records:  
Vivian, artist, 25-34, 5, Lethal Weapon  
Vivian, Artist, 25-34, 4, Raising Arizona  
Mercedes, K-12 student, Under 18, 3, Lethal Weapon  
Mercedes, K-12 student, Under 18, 3, The Rock  
...
```

- Example 2

Query statement:

```
match (n)-->(m) where id(n)='Vivian' return labels(m),count(*)
```

Result:

```
#data_total_size:2  
#data_return_size:2  
#data_offset:0  
#records:  
user,5  
movie,2
```

7.1.12.4 Listing Jobs (2.2.13)

Function

After the ID of an asynchronous job is returned, if the job ID at the service layer is lost and cannot be obtained through the API, a new API is provided to query all asynchronous jobs stored in the engine. The job ID, job status, and original request of each job are returned.

URI

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status?  
limit={limit}&offset={offset}
```

Table 7-262 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
offset	No	Integer	Offset of a query. The default value is 0 .
limit	No	Integer	Maximum number of records that can be queried. The default value is 100000 .

Request Parameters

For details, see the URI parameters.

Response Parameters

Table 7-263 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
result	String	The value is success if the query is successful; this parameter is not returned if the query fails.
jobs	Object	Job status list stored in the system. If the execution succeeds, this parameter is returned. Table 7-264 describes the structure of a single job.

Table 7-264 Job status structure

Parameter	Type	Description
jobId	String	Job name.

Parameter	Type	Description
rawRequest	String	Original request body
taskType	String	Job type
canStop	Boolean	Whether the job can be stopped
progress	String	Job progress. The value range is [0, 100].
startTime	String	Start time, which is a string of the date type or a timestamp
endTime	String	End time, which is a string of the date type or a timestamp
status	String	Job status when the query is successful. The options are: • pending • running • success • partiallyFailed • failed This parameter is left blank when the query fails.
failReason	String	Failure cause. This parameter is returned only when status is failed .
files	Object	List of imported files. This parameter is returned only when taskType is ImportGraph .

Example Request

Query the job list and return the job ID and status of each job.

```
GET /ges/v1.0/{project_id}/graphs/movie/jobs/status?limit=2&offset=0
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "jobs": [
    {
      "jobId": "b236a002-6acb-40cd-acca-bc3eb96b807d001680206",
      "rawRequest": "{\"algorithmName\":\"shortest_path\",\"parameters\":{\"source\":\"0000\",\"target\":\"38\"}}",
      "taskType": "Algorithm",
      "canStop": false,
      "progress": "100",
      "startTime": "2024-05-23 09:54:00",
      "failReason": "Running algorithm [shortest_path] error: parameter [source] is invalid!"
    }
  ]
}
```



```
    "endTime": "2024-05-23 09:54:00",
    "status": "failed"
  },
  {
    "jobId": "5a39d9f6-e955-4294-8ec6-2fe18eee98c7001680206",
    "rawRequest": "{\"algorithmName\":\"shortest_path\",\"parameters\":{\"source\":\"46\",\"target\":\"38\"}}",
    "taskType": "Algorithm",
    "canStop": false,
    "progress": "100",
    "startTime": "2024-05-23 09:41:40",
    "endTime": "2024-05-23 09:41:40",
    "status": "success"
  },
  {
    "jobId": "cef6ae30-f21b-4a2a-a83c-cf91bee679d9001680206",
    "rawRequest": "",
    "taskType": "ImportGraph",
    "canStop": false,
    "progress": "100",
    "files": [
      {
        "edgeFiles": [
          {
            "fileName": "/root/ges-install/auDatas/ranking_edge.csv",
            "totalLines": 1659,
            "startTime": 1716428001745772,
            "successfulLines": 1659,
            "endTime": 1716428001757920,
            "status": "success",
            "failedLines": 0
          }
        ]
      },
      {
        "vertexFiles": [
          {
            "fileName": "/root/ges-install/auDatas/movies_vertex_new.csv",
            "totalLines": 146,
            "startTime": 1716428001656072,
            "successfulLines": 146,
            "endTime": 1716428001659352,
            "status": "success",
            "failedLines": 0
          }
        ]
      }
    ],
    "schemaFiles": [
      {
        "totalLabels": 49,
        "fileName": "/root/ges-install/auDatas/schema.xml.bak",
        "failedLabels": 0,
        "startTime": 1716428001563921,
        "successfulLabels": 49,
        "endTime": 1716428001655884,
        "status": "success"
      }
    ]
  },
  {
    "startTime": "2024-05-23 09:33:21",
    "endTime": "2024-05-23 09:33:21",
    "status": "success"
  }
],
"jobCount": 3
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.13 Custom Operations APIs

7.1.13.1 Performing Custom Operations

Function

This API is used to perform a specified custom operation.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-operation

Table 7-265 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-266 Request body parameters

Parameter	Mandatory	Type	Description
api	Yes	String	API command used by the custom operation. The options are cypher , gremlin , algorithm , and path_query .
command	Yes	String	Command executed by the custom operation.

Response Parameters

Table 7-267 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	json	Execution result of the custom operation. This parameter is left blank when the request fails.

Example Request

Perform a specified custom operation. The API used for the custom operation is **gremlin**, and the command is `{"command\":"g.V('1')\}"`.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-operation
{
  "api": "gremlin",
  "command": "{\"command\":"g.V('1')\"}"
}
```

Example Response

Status code: 200

Example response (successful request)

```
{
  "data":{
    "vertices":[
      {
        "id": "1",
        "label": "movie",
        "properties":{
          "genres":[
            "Comedy"
          ],
          "movieid":[
            1
          ],
          "title":[
            "Airplane! (1980)"
          ]
        }
      }
    ],
    "runtime":0.126476598
  }
}
```

Status code: 400

Example response for a failed request

```
Internal Server Error
{
  "errorCode": "GES.8814",
  "errorMessage": "Unsupported API."
}
```

Status Code

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Code

See [Error Code](#).

7.1.14 Filtered Query (2.2.13)

Function

This API filters the k-hop process layer by layer, and lists the k hop vertices or edges that meet the filtering criteria.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=filtered-query

Table 7-268 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

NOTE

If **executionMode** is **sync**, the number of returned vertices cannot exceed 100,000.

Table 7-269 Request body parameters

Parameter	Mandatory	Type	Description
executionMode	No	String	sync: synchronous async: asynchronous The default value is sync , indicating synchronous response.
vertices	Yes	Array of Json	List of IDs of source vertices you want to query
query_type	No	String	Possible values are Default , AllVertices , SimpleEdges , Path . Default indicates the default mode, that is, the k hop is returned. AllVertices returns details about all vertices within k hops. SimpleEdges returns all edges within k hops, contain only the ID, index, and label information of the edges. Path returns the path information, that is, the set of paths.

Parameter	Mandatory	Type	Description
by	No	Array of Json	Specified output field. This parameter is valid only when query_type is set to Default or AllVertices . Currently, only one layer is supported. If this parameter is not specified, all content is output by default.
edges	No	Array of Json	List of edges to be queried. Either this parameter or vertices is selected. For details, see Table 7-270 .
filters	Yes	Array of Json	Filter criteria. Each element in the array corresponds to a filter. For details about the formats, see Table 7-271 .
full_path	No	Boolean	Whether to return a complete path. The default value is false . - If the value is true, the paths from the source vertex to all leaf vertices are returned. - If the value is false, the paths from the source vertex to the leaf vertices at layer k are returned.
visualized	No	Boolean	Whether to enable visualization. The default value is false . In asynchronous mode: - When visualized is false, the job query result is returned on multiple pages. - When visualized is true, the job query result is returned on one page.
restricted(2.2.28)	No	Boolean	Whether the input is restricted. The default value is true . true: If vertices contains vertices that do not exist, the query exits and an error is reported. false: The system filters out vertices that do not exist and then performs the query task.

Table 7-270 edges element formats

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex ID
target	Yes	String	Target vertex ID

Parameter	Mandatory	Type	Description
index	No	String	Indexes of edges in the source edge set

Table 7-271 Filters element formats

Parameter	Mandatory	Type	Description
operator	Yes	String	Query type. Possible values: • inV: incoming vertex • outV: outgoing vertex • bothV: incoming and outgoing vertices • vertex: all vertices. Filtering is available only at the first layer. If vertices are input in the beginning, the first-layer output is the input vertices. If no vertices are input in the beginning, all vertices are output at the first layer. • in: incoming edge • out: outgoing edge • both: incoming and outgoing edges • edge: all edges. Filtering is available only at the first layer. The usage is similar to that of vertices The query result of the previous layer is the input of the next layer. • If the result of the previous layer is a vertex, the corresponding operations can be inV, outV, bothV, in, out, and both. • If the result of the previous layer is an edge, the corresponding operation can be inV, outV, and bothV.
vertex_filter	No	String	This parameter is optional when operator is set to inV , outV , or bothV . For details about the formats, see Table 7-273 .
edge_filter	No	String	This parameter is optional when operator is set to in , out , or both . For details about the formats, see Table 7-273 .

Table 7-272 by element formats

Parameter	Mandatory	Type	Description
id	No	Boolean	Whether to output the ID. The default value is false .
label	No	Boolean	Whether to output the label. The default value is false .
properties	No	Boolean	Whether to output properties. The default value is false .
selectedProperties	No	Array of String	When properties is set to true , you can select the properties to be output. If this parameter is left blank, all properties are output. By default, this parameter is left blank.

Table 7-273 property_filter elements

Parameter	Mandatory	Type	Description
leftvalue	Yes	String	Left value. For details about the formats, see Table 7-274 .

Parameter	Mandatory	Type	Description
predicate	Yes	String	Filtering type. The supported operations include: Relational operators: • =: equal to • !=: not equal to • <: less than • ≤: Less than or equal to • >: greater than • ≥: greater than or equal to Logical operations: • &: and • : or Set operations: • IN/NOTIN: whether the left value and right value have an intersection • CONTAIN/NOTCONTAIN: whether the property value contains the right value. • SUBSET: The right value is a subset of the property value. Match operators: • PREFIX: The right value is the prefix of the left value. • NOTPREFIX: The right value is not the prefix of the left value. • SUFFIX: The right value is the suffix of the left value. • NOTSUFFIX: The right value is not the suffix of the left value. • SUBSTRING: The right value is a sub-string of the left value. • NOTSUBSTRING: The right value is not a sub-string of the left value. • FUZZY: fuzzy match • REGEX: expression match • CISUBSTRING: sub-string that ignores cases HAS/HASNOT: whether this property exists. Only property filtering is supported. That is, the left value can only be property_name.
rightvalue	Yes	String	Right value. For details about the formats, see Table 7-275 .

Table 7-274 leftvalue elements

Parameter	Mandatory	Type	Description
label_name	No	String	If label is used as the filter criterion, label_name can be selected and the value is labelName . Set the value field of rightvalue to the label name.
property_name	No	String	If property is used as the filter criterion, set this parameter to the property name and set value of rightvalue to the property value.
id	No	String	If the vertex ID is filtered, this parameter is optional.
property_filter	No	String	If predicate is set to & or , property_filter can be nested in leftvalue and rightvalue .

Table 7-275 rightvalue elements

Parameter	Mandatory	Type	Description
value	Yes	String	- If label is used as the filter criterion, the value is the label name. - If property is used as the filter criterion, the value is the property name.
property_filter	No	String	If predicate is set to & or , property_filter can be nested in leftvalue and rightvalue .

Table 7-276 predicate supported left values

predicate	label_name	id	property_name	Nested Filters
&	No	No	No	Yes
	No	No	No	Yes
HAS/HASNOT	No	No	Yes	No
CONTAIN/NOTCONTAIN	No	No	Yes	No

predicate	label_name	id	property_name	Nested Filters
SUBSET	No	No	Yes	Yes (Only the right value set is supported. If the right value is single, no filtering function is available.)
IN/NOTIN	Yes	Yes	Yes	Yes (Only the right value set is supported. If the right value is single, no match is available.)
PREFIX	Yes	Yes	Yes	No
FUZZY	Yes	Yes	Yes	No
REGEX	Yes	Yes	Yes	No
SUBSTRING	Yes	Yes	Yes	No
CISUBSTRING	Yes	Yes	Yes	No
=/! =/<</<=>/>=	Yes	Yes	Yes	No

 NOTE

- The left value set is supported. The left value in the body is a string.
- The right value set is supported. If you select **No**, only the first character string in the set is matched even if the right value set is supported.
- Boolean value matching. When the right value is **true**, the value is identified as true for matching. Otherwise, the value is identified as false for matching.

Response Parameters

- Synchronous response

Table 7-277 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	Object	Query results. This parameter is left blank when the query fails.

Table 7-278 data parameter description

Parameter	Type	Description
vertices	List	Vertex result set. If the last layer of filters is vertex filtering, the data contains vertices.
edges	List	Edge result set. If the last layer of filters is edge filtering, the data contains edges.
paths	List	Path set. The format is [[path1],[path2]] . For details, see Table 7-279 .

Table 7-279 path parameter description

Parameter	Type	Description
source	String	Source vertex ID
target	String	Target vertex ID
index	String	Edge index
label	String	Edge label

- Asynchronous response

Table 7-280 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails.
jobType	Integer	Job type. This parameter is left blank when the request fails.

Example Request

- (Synchronous mode) List the kth-hop vertices or edges that meet filter criteria. The execution mode is synchronous. Visualization is not performed. That is, job query results are displayed on multiple pages.

POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=filtered-query

```
{
  "executionMode": "sync",
  "visualized": "false",
  "filters": [
    {
      "operator": "outV"
    },
    {
      "operator": "out",
      "edge_filter": {
        "property_filter": {
          "leftvalue": {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "rate"
          }
        }
      }
    }
  ],
  "full_path": false,
  "vertices": [
    "tr_10"
  ]
}
```

- (Asynchronous mode) List the kth-hop vertices or edges that meet filter criteria. The execution mode is asynchronous. Visualization is not performed. That is, job query results are displayed on multiple pages.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=filtered-query
{
  "executionMode": "async",
  "visualized": "false",
  "filters": [
    {
      "operator": "outV"
    },
    {
      "operator": "out",
      "edge_filter": {
        "property_filter": {
          "leftvalue": {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "rate"
          }
        }
      }
    }
  ],
  "full_path": false,
  "vertices": [
    "tr_10"
  ]
}
```

- property_filter is nested. List the kth-hop vertices or edges that meet filter criteria. The execution mode is synchronous. Visualization is not performed. That is, job query results are displayed on multiple pages.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=filtered-query
{
  "executionMode": "sync",
  "filters": [
    {
      "operator": "outV",
      "vertex_filter": {
        "property_filter": {
          "leftvalue": {
            "property_filter": {
              "leftvalue": {
                "property_name": "genres"
              },
              "predicate": "PREFIX",
              "rightvalue": {
                "value": "A|"
              }
            }
          }
        }
      },
      "predicate": "&",
      "rightvalue": {
        "property_filter": {
          "leftvalue": {
            "label_name": "labelName"
          },
          "predicate": "=",
          "rightvalue": {
            "value": "movie"
          }
        }
      }
    }
  ],
  "vertices": [
    "tr_3"
  ]
}
```

```
]
}
```

Example Response

- Synchronous response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "data": {
    "edges": [
      {
        "index": "1",
        "source": "tr_1",
        "label": "rate",
        "properties": {
          "Rating": [
            0
          ],
          "Datetime": [
            ""
          ]
        },
        "target": "tr_3"
      },
      .....
      {
        "index": "199998",
        "source": "tr_1",
        "label": "rate",
        "properties": {
          "Rating": [
            0
          ],
          "Datetime": [
            ""
          ]
        },
        "target": "tr_200000"
      }
    ]
  }
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "graph [tesdt_117] is not found",
  "errorCode": "GES.8806"
}
```

- Asynchronous response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "jobId": "6622f13c-4b88-45f5-89a9-eea096647c4a",
  "jobType": 1
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.15 Filtered Query V2 (2.3.6)

Function

This is a new version of the Filtered Query API. This API supports both Filtered Query and Repeat Query functions.

 NOTE

You can use this API to accelerate multi-hop filtered query and looped traversal query. For example, a Gremlin statement is as follows:
g.V('node1').repeat(out('label_2')).emit()

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=path-query

Table 7-281 URI parameters

Parameter	Mandator y	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

NOTE

The number of elements in each traversal cannot exceed 100 million.

Table 7-282 Request body parameter descriptions

Parameter	Mandatory	Type	Default Value	Description
executionMode (2.2.22)	No	String	sync	Execution mode of the query task. The value can be sync (synchronously) or async (asynchronously).
vertices	Yes	Array of Json	None	List of IDs of source vertices you want to query
repeat	Yes	Array of Json	None	Filter criteria for repeat queries. Each element in the array corresponds to a filter. For details about the format, see Table 7-284 .
until (2.2.22)	No	Array of Json	None	while/do looping mode. Used to terminate traversing the filter criteria list. For the specific format, refer to Table 7-285 . For APIs of the 2.2.22 version, until supports only one stop condition.
times	No	Integer	2	Maximum number of steps. The default value is 2 , and the maximum value is 20 .
emit	No	Boolean	false	Whether all elements will be returned. The default value is false . If select and as are configured, or queryType is set to Tree , this parameter determines whether vertices that are not on the final complete path will be returned.
limit	No	Integer	10000	Number of vertices, edges, or paths.
queryType(2.2.22)	No	String	Default	Type of the query. The value can be Default or Tree . Default means that the path query result will be returned. Tree means that the path information of the path query will be returned in the tree structure.

Parameter	Mandatory	Type	Default Value	Description
select (2.2.21)	No	Array of String	None	Fields you want to be displayed in the result. The values can be the fields set in the as parameter, or you can set the by parameter together with this parameter to control output content. If you set the by parameters together, you can set select to v0, v1, v2, ..., and vtimes: • v0: layer 0 of the K hops • v1: the first layer of the K hops • v2: the second layer of the K hops The path selected by this parameter is deduplicated by default.
by(2.2.21)	No	Array of Json	Output all	Content of output fields. 1. If this parameter is not specified, all content is output by default. 2. If the select parameter is configured, the number of by must be the same as the number of select. Each by can output only one element. 3. When select is not used to choose output elements, by operates on the final result set. Refer to Table 7-283 for more details.
statistics	No	Boolean	false	Whether only the number of hit records will be returned. The default value is false .
mode	No	Boolean	None	Traversal mode that will be set forcibly. The value can be Dense or Sparse . The default mode is automatically switched based on the graph structure.
strategy	No	String	ShortestPath	Traversal policy. The value can be ShortestPath or Walk .

Parameter	Mandatory	Type	Default Value	Description
restricted(2.2.28)	No	Boolean	true	Whether the input is restricted. The default value is true . • true: If vertices contains vertices that do not exist, the query exits and an error is reported. • false: The system filters out vertices that do not exist and then performs the query task.

Table 7-283 by elements

Parameter	Mandatory	Type	Description
id	No	Boolean	Whether to output the ID. The default value is false .
label	No	Boolean	Whether to output the label. The default value is false .
properties	No	Boolean	Whether to output properties. The default value is false .
selectedProperties	No	Array of String	When properties is set to true , you can select the properties to be output. If this parameter is left blank, all properties are output.

Table 7-284 repeat elements

Parameter	Mandatory	Type	Description
operator	Yes	String	Type of the query. The value can be inV (incoming vertex), outV (outgoing vertex), and bothV (incoming and outgoing vertices).
vertex_filter	No	JSON String	Search criteria for the next hop. For details about the format, see Table 7-286 .
edge_filter	No	JSON String	Edge search criteria. For details about the format, see Table 7-286 .
as	No	JSON String	Alias of elements at this layer. The value can be used to select output fields.

Table 7-285 until elements

Parameter	Mandatory	Type	Description
vertex_filter	No	JSON String	This parameter is optional when operator in repeat is set to inV , outV , or bothV . For details about the format, see Table 7-286 .

Table 7-286 property_filter elements

Parameter	Mandatory	Type	Description
leftvalue	Yes	String	Left value of a search criterion. For details about the format, see Table 7-287 .

Parameter	Mandatory	Type	Description
predicate	Yes	String	Filtering type. The options are: Relational operators: • =: equal to • !=: not equal to • <: less than • <=: less than or equal to • >: greater than • >=: greater than or equal to Logical operations: • &: and • : or Set operations: • IN/NOTIN: whether the left value and right value have an intersection • CONTAIN/NOTCONTAIN: whether the property value contains the right value • SUBSET: The right value is a subset of the property value. Match operators: • PREFIX: The right value is the prefix of the left value. • NOTPREFIX: The right value is not the prefix of the left value. • SUFFIX: The right value is the suffix of the left value. • NOTSUFFIX: The right value is not the suffix of the left value. • SUBSTRING: The right value is a sub-string of the left value. • NOTSUBSTRING: The right value is not a sub-string of the left value. • FUZZY: fuzzy match • REGEX: regular expression match • CISUBSTRING: sub-string that ignores cases HAS/HASNOT: whether this property exists. Only property filtering is supported. That is, the left value can only be property_name.
rightvalue	Yes	String	Right value of a search criterion. For details about the format, see Table 7-288 .

Table 7-287 leftvalue elements

Parameter	Mandatory	Type	Description
label_name	No	String	If label is used as the filter criterion, label_name can be selected and the value is labelName . Set the value field of rightvalue to the label name.
property_name	No	String	If property is used as the filter criterion, set this parameter to the property name and set value of rightvalue to the property value.
id	No	String	If the vertex ID is filtered, this parameter is optional.
property_filter	No	JSON String	If predicate is set to & or , property_filter can be nested in leftvalue and rightvalue .
degree	No	String	Direction of vertex degree filtering statistics. This parameter is optional. The value can be both , in , or out .

Table 7-288 rightvalue elements

Parameter	Mandatory	Type	Description
value	Yes	String	If label is used as the search criterion, the value is the label name. If property is used as the search criterion, the value is the property name.
property_filter	No	JSON String	If predicate is set to & or , property_filter can be nested in leftvalue and rightvalue .

Table 7-289 predicate supported left values

predicate	label_name	id	property_name	Nested Filters
&	No	No	No	Yes
 	No	No	No	Yes
HAS/HASNOT	No	No	Yes	No
=/!=/</>/>=	Yes	Yes	Yes	No

Response Parameters

- Synchronous response

Table 7-290 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.
data	Object	Query results. This parameter is left blank when the query fails.

Table 7-291 data parameter descriptions

Parameter	Type	Description
vertices	List	Vertex result set. If the last layer of filters is vertex filtering, the data contains vertices.
edges	List	Edge result set. If the last layer of filters is edge filtering, the data contains edges.

- Asynchronous response

Table 7-292 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails.
jobType	Integer	Job type. This parameter is left blank when the request fails.

Example Request

- Example request 1: List the kth-hop vertices or edges that meet filter criteria. The query type is outgoing vertex, and the query is performed on the next-hop vertex.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=path-query
{
  "repeat":[
    {
      "operator":"outV",
      "vertex_filter":{
        "property_filter":{
          "leftvalue":{
            "label_name":"labelName"
          },
          "predicate":"=",
          "rightvalue":{
            "value":"rate"
          }
        }
      }
    }
  ],
  "times":2,
  "vertices":[
    "1","2"
  ]
}
```

NOTE

The preceding request is equivalent to this Gremlin statement:
`g.V('1','2').repeat(out().hasLabel('rate').times(2).dedup()).`

- Example request 2: List the kth-hop vertices or edges that meet filter criteria. The query type is outgoing vertex, and the query is performed on the next-hop vertex.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=path-query
{
  "repeat":[
    {
      "operator":"outV",
      "vertex_filter":{
        "property_filter":{
          "leftvalue":{
            "label_name":"labelName"
          }
        }
      }
    }
  ]
}
```



```
    },
    "predicate": "=",
    "rightvalue": {
      "value": "rate"
    }
  }
}
],
"until": [
  {
    "vertex_filter": {
      "property_filter": {
        "leftvalue": {
          "property_name": "movieid"
        },
        "predicate": "=",
        "rightvalue": {
          "value": "1"
        }
      }
    }
  }
],
"vertices": [
  "v1", "v2"
]
}
```

NOTE

The preceding request is equivalent to this Gremlin statement:

```
g.V('v1','v2').repeat(out().hasLabel('rate')).until(has('movieid','1')).dedup()
```

Example Response

- Synchronous response

Status code: 200

Example response for a successful request

```
{
  "data": {
    "vertices": [
      {
        "id": "51",
        "label": "user",
        "properties": {
          "occupation": [
            "homemaker"
          ],
          "gender": [
            "F"
          ],
          "Zip-code": [
            "46911"
          ],
          "userid": [
            5
          ],
          "age": [
            "56+"
          ]
        }
      }
    ]
  }
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "graph [tesdt_117] is not found",
  "errorCode": "GES.8806"
}
```

- Asynchronous response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "6622f13c-4b88-45f5-89a9-eaa096647c4a",
  "jobType": 1
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.16 Cypher Queries (2.2.16)

7.1.16.1 Executing Cypher Queries

Function

Cypher is a widely used declarative graph database query language. It can be used to query data in GES and returns results. Graph statistics are used in Cypher implementation. Currently, the label-based vertex and edge indexes are used

during Cypher query and compilation. To use Cypher normally, create indexes by referring to [Cypher Prerequisites](#).

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query

Table 7-293 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-294 Request body parameter descriptions

Parameter	Mandatory	Type	Description
statements	Yes	List	Statement group that contains one or more statements. Table 7-295 describes the format of each element.

Table 7-295 statements parameter descriptions

Parameter	Mandatory	Type	Description
statement	Yes	String	Cypher statement
parameters	No	Object	Cypher statement parameters, which are used for parameterized queries. By default, this field is left blank. For details, see parameterized queries .
resultDataContents	No	String or List	Format of the returned result. You can set one or more formats. The options are row , graph , and raw (new in version 2.2.27).

Parameter	Mandatory	Type	Description
includeStats	No	Boolean	Whether the returned result contains addition, deletion, and modification statistics. If this parameter is not set, the returned result does not contain the information by default.
runtime	No	String	Executor type. The value can be map , slotted , or block . The default value is map . NOTE 1. The slotted executor is supported since version 2.3.14. 2. The block executor is supported since version 2.4.1. 3. Compared with the map executor, the slotted and block executors complete more statement data flow analysis in the plan generation phase of statements. In most cases, they execute faster while requiring less memory.
executionMode (2.2.23)	No	String	Execution mode. Set this parameter to sync for synchronous execution and to async for asynchronous execution. If this parameter is not set, the execution is synchronous by default. For details about how to obtain the query result in asynchronous mode, see Querying Job Status on the Service Plane .
limit (2.2.23)	No	Int	Maximum number of asynchronous results. This parameter applies only when executionMode is async . The default value is 100000 .

 NOTE

- You can add the **explain** or **profile** prefix before the statement to display the query plan.
 - **explain** displays only the query plan but does not execute the statement. The explain prefix is supported since version 2.2.20.
 - **profile** displays the query plan and executes the statement. The profile prefix is supported since version 2.3.12.
- In asynchronous mode (**executionMode** is **async**), Cypher query results can be exported to CSV files (GES 2.3.4 or later supports this function). For details, see [Exporting Job Execution Results to Files \(2.2.1\)](#). Currently, the following values can be returned:
 1. Vertex and edge single-value properties, vertex and edge IDs, and group counts.
 2. The current version does not support exporting object types. Objects are converted to null values in the CSV file.

Response Parameters

Table 7-296 Response body parameter descriptions

Parameter	Type	Description
results	List	Each element is the return result of a Cypher statement.
errors	List	Each element contains the code and message information in string form.

Table 7-297 Elements of the results parameter

Parameter	Type	Description
columns	List	Name of a returned field
data	List	Returned data value. Each element indicates a record.
stats	Object	Addition, deletion, and modification statistics
plan	Object	If the Cypher statement contains the explain or profile prefix, this field contains the query plan. Otherwise, this field is not displayed. The profile feature is supported since version 2.3.12.
jobId(2.3.10)	String	Asynchronous job ID if the request is executed asynchronously
jobType(2.3.10)	Integer	Type of the asynchronous job if the request is executed asynchronously

Table 7-298 Elements of the data parameter

Parameter	Type	Description
row	List	Content of a specific row. Each element corresponds to a field in the row. This parameter is displayed only when resultDataContents is empty or contains row .
meta	List	Type of each field in a row. This parameter is displayed only when resultDataContents is empty or contains row .
graph	Object	Information returned in graph format. This parameter is displayed only when resultDataContents contains graph .
raw(2.2.27)	List	Information returned in raw format. This parameter is displayed only when resultDataContents contains raw .

Table 7-299 stats elements in a response

Parameter	Type	Description
contains_updates	Boolean	Whether data is modified during the query
edges_created	Integer	Number of created edges
edges_deleted	Integer	Number of deleted edges
labels_set	Integer	Number of labels that have been set
properties_set	Integer	Number of properties that have been set
vertices_created	Integer	Number of created vertices
vertices_deleted	Integer	Number of deleted vertices

Example Request

Run the Cypher statement **match (n) return n limit 1**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query
{
  "statements": [{
    "statement": "match (n) return n limit 1",
    "parameters": {},
```

```
        "resultDataContents": ["row"],  
        "includeStats": false  
    }  
}
```

Example Response

Status code: 200

Example response for a successful request (synchronous call)

Http Status Code: 200

```
{  
  "results": [  
    {  
      "columns": ["n"],  
      "data": [  
        {  
          "row": [  
            {  
              "occupation": "artist",  
              "gender": "F",  
              "Zip-code": "98133",  
              "userid": 0,  
              "age": "25-34"  
            }  
          ],  
          "meta": [  
            {  
              "id": "46",  
              "type": "node",  
              "labels": [  
                "user"  
              ]  
            }  
          ]  
        }  
      ],  
      "stats": {  
        "contains_updates": false,  
        "edges_created": 0,  
        "edges_deleted": 0,  
        "labels_set": 0,  
        "properties_set": 0,  
        "vertices_created": 0,  
        "vertices_deleted": 0  
      }  
    }  
  ],  
  "errors": []  
}
```

Status code: 200

Example response for a successful request (asynchronous call)

Http Status Code: 200

```
{  
  "results": [  
    {  
      "columns": [  
        "jobId",  
        "jobType"  
      ],  
      "jobId": "b64a5846-e306-4f87-b0f1-d595ee2a9990",  
      "jobType": 1,  
      "data": [  
        {  
          "row": [  
            {  
              "occupation": "artist",  
              "gender": "F",  
              "Zip-code": "98133",  
              "userid": 0,  
              "age": "25-34"  
            }  
          ],  
          "meta": [  
            {  
              "id": "46",  
              "type": "node",  
              "labels": [  
                "user"  
              ]  
            }  
          ]  
        }  
      ],  
      "stats": {  
        "contains_updates": false,  
        "edges_created": 0,  
        "edges_deleted": 0,  
        "labels_set": 0,  
        "properties_set": 0,  
        "vertices_created": 0,  
        "vertices_deleted": 0  
      }  
    }  
  ],  
  "errors": []  
}
```

```
{
  "row": [
    "b64a5846-e306-4f87-b0f1-d595ee2a9990",
    1
  ],
  "meta": [
    null,
    null
  ]
}
],
"errors": []
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "results": [],
  "errors": [
    {
      "code": "GES.8901",
      "message": "No statement provided."
    }
  ]
}
```

Status Codes

Return Value	Description
202 Accepted	The request has been received but has not been processed.
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.16.2 Cypher Prerequisites

The current Cypher query compilation process uses the label-based vertex and edge indexes. To use Cypher normally, use the [index creation API](#) to create indexes.

- Example command for adding a vertex label index. The index name is **cypher_vertex_index**, and the index type is global vertex index.


```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
{
  "indexName": "cypher_vertex_index",
  "indexType": "GlobalCompositeVertexIndex",
  "hasLabel": "true",
  "indexProperty": []
}
```

- Example command for adding an edge label index. The index name is **cypher_edge_index**, and the index type is global edge index.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
{
  "indexName": "cypher_edge_index",
  "indexType": "GlobalCompositeEdgeIndex",
  "hasLabel": "true",
  "indexProperty": []
}
```

NOTE

1. You do not need to create the indexes if the graph is of the Hundred-million-edge, Billion-edge, or Ten-billion-edge types. (The GES version must be 2.3.6 or later.)
2. If you need to create indexes, you must create both two indexes (vertex label index and edge label index) at the same time to use Cypher for query.
3. If your graph already has a vertex index or an edge index whose **hasLabel** is **true** and **indexProperty** is empty, you do not need to create the vertex index or edge index again.
4. The API for creating an index is asynchronous. To check whether the index is successfully created, use the [API for querying job status](#).
5. If fine-grained permissions are used, a member account must have the schema permission and the read permission on all labels (including the default label **__DEFAULT__**) to create indexes. Otherwise, after an index is created, you need to use an account with the required permissions to send **call db.schema()** to update the compiler cache of Cypher queries.

7.1.16.3 Basic Operations and Compatibility

Basic Operations

Operation	Cypher Statement
Querying vertices	match (n) return n
Querying edges	match (n)-[r]->(m) return n, r, m
Querying paths	match (n:user)-[r]->(m:movie)-->(s:series) return n,r,m,s
Querying information by specifying filtering criteria	match(n:user) where n.userid>=5 return n
Grouping and aggregating	match(n:movie) return n.genres, count(*)
Deduplicating	match(n:movie) return distinct n.genres
Sorting	match(n:movie) return n order by n.movieid
Creating a vertex	create (n:user{userid:1}) return n

Operation	Cypher Statement
Creating an edge	match (n:user{userid:15}),(m:movie{movieid:10}) create (n)-[r:rate]->(m)
Deleting a vertex	match (n:user{userid:1}) delete n
Modifying labels	match (n:user{userid:1}) set n:movie return n
Modifying properties	match (n:user{userid:1}) set n.userid=2 return n

Compatibility to Cypher

1. Cypher clauses

Cypher implements a couple of clauses. You can combine clauses to implement different query semantics, including vertex and edge filtering, multi-hop query, sorting and deduplication, and grouping and aggregation. Currently, GES supports the Cypher clauses listed in the following table.

Table 7-300 Supported Cypher clauses

Clause	Support	Example
match	Partially supported	match (n:movie) return n
optional match	Partially supported	optional match (n)-->(m) where id(n)='1' return m
return	Supported	return [1,2,3] as p
with	Supported	match (n) with labels(n) as label, count(*) as count where count > 10 return *
where	Supported	match (n:movie) where n.movieid > 10 return n
order by	Supported	match (n:movie) return n order by n.genres
skip	Supported	match (n:movie) return n order by n.genres skip 5
limit	Supported	match (n:movie) return n order by n.genres skip 5 limit 10
create	Supported	create (n:user{_ID_: 'Jack' }) return n
delete	Supported	match (n:movie)->[r]-(m:user) delete r
set	Supported	match (n:user{userid:0}) set n.gender='M' return n

Clause	Support	Example
call procedures	Supported	call db.schema()
unwind	Supported	unwind [1, 2, 3] as p return p
union	Supported	match (n:movie) return id(n) union match (n:user) return id(n)

 NOTE

1. Currently, operations such as merge and foreach are not supported. Cypher statements cannot be used to add or delete indexes.
 2. Since GES metadata is not schema-free, there are strict restrictions on vertex and edge label properties, hence the Remove operation is not supported.
 3. The ORDER BY clause does not support sorting of the List type. When Cardinality of the property value is not single, the sorting result is unknown.
 4. The CREATE statement does not support referencing the properties of the vertex to be created. For example, in the following statement, when the prop property of vertex b is set to **a.id**, it will not be assigned the value 0.
CREATE (a:End {prop: 42, id: 0}), (b:Begin {prop: a.id})
- Available items for the match clause

Item	Description	Example Clauses	Earliest Version Required
Vertex pattern	Patterns for matching vertex with specified labels, properties, and IDs.	match (n:movie{title:'hello'}) match (n) where id(n)='xx'	2.2.16
Edge pattern	Patterns for matching directional and non-directional edges with specified labels and properties. Specified IDs of both start and end vertices are supported.	match (n)-[r]->(m) match (n)-[r]-(m) match (n)-[r:rate{Rating:1}]- (m) match (n)-[r]-(m) where id(n)='x'and id(m)='y'	2.2.16
Path	Anonymous paths	match (n)-[r]->(m)-->(s)	2.2.16
	Named paths	match p=(n)-[r]->(m)-->(s)	2.2.19

Multiple patterns	You can enter multiple patterns following match and separate them with commas (.). match (n)-[r]->(m), (m)-->(s)	2.2.16
Multi-match	You can enter multiple match clauses. You can use with to connect multiple clauses. match (n)-[r]->(m) with m match (m)-->(s)	2.2.16
Variable-length path pattern	Patterns for matching variable-length paths starting with a specified vertex. match p=(n)-[r*1..3]->(m) where id(n)='xx' return p match p=(n{title:'name'})-[r*1..3]->(m) return p	2.2.19
	Traversal conditions for matching variable-length paths. match p=(n)-[r*1..3]->(m) where id(n)='xx' and all (x in nodes(p) where x.prop='value1') return p	2.2.28
	Both start vertex and end vertex of a variable-length path can be specified. match p=(n)-[r*1..3]->(m) where id(n)='xx' and id(m)='y' return p	2.3.9
	Deduplication by end vertex is supported: match p=(n)-[r*1..3]->(m) where id(n)='xx' return distinct m	2.3.17

2. Parameterized queries

Cypher supports parameterized queries. Numeric and string values in a query statement are extracted and converted to parameters for faster compilation, improving the query speed.

There are some examples of parameterized queries:

– Example 1

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?
action_id=execute-cypher-query
{
  "statements": [{
    "statement": " match (n:user) where n.occupation = $occupation return n",
    "parameters": {
      "occupation" : "artist"
    },
    "resultDataContents": ["row"]
  }]
}
```

– Example 2

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?
action_id=execute-cypher-query
{
  "statements": [{
    "statement": " match (n:user {`Zip-code`:98133'}) set n = $props return n",
    "parameters": {
```

```
    "props": {  
      "gender": "M",  
      "age": "56+"  
    },  
    "resultDataContents": ["row"]  
  }  
}
```

NOTE

There are some scenarios where parameterized queries are not supported. The following syntax is not valid:

1. Using **\$param** to search by property key and value. For example, **match (n) where n.\$param = 'something'**
2. Using **\$code** for vertex and edge labels. For example, **match (n:user) set n:\$code**
3. Supported data types

Currently, GES supports 10 data types: char, char_array, float, double, Boolean, long, Integer, date, enum, and string. Both Boolean and numeric types are supported in the Cypher syntax. The following table lists the mapping between other types and Cypher data types.

Table 7-301 Mapping between data types of GES and Cypher

GES	Cypher	Description
char	String	-
char_array	String	-
string	String	-
enum	String	The Cypher syntax does not provide enum-related syntax. During Cypher query, an enum is converted to a string. When Cypher is used to set properties, values that are not in the enumeration list fail to be set.
date	Temporal	Currently, Cypher dates can be converted into GES dates, but Cypher date functions cannot be used for inputting a date.

Table 7-302 Special types supported by Cypher

Type	Supported	Example
Node	Yes	match (n) return n limit 10
Relationship	Yes	match (n)-[r]->(m) return r limit 10
List	Yes	return [1,2,3] as li
Map	Yes	match (n)-->(m) return {start:id(n), end:id(m)}

Type	Supported	Example
Path	Yes	match p=(n1)-[:friends*1..2]-(n2) return p limit 10
Point, Spatial	No	-

 NOTE

For the special types listed above, only the **List** can be used to match multi-value properties in GES. Other types cannot be set as the value of a property on vertices or edges using the set statement.

4. Vertex ID compatibility

- Cypher does not provide the syntax for setting the ID when a vertex is added. In GES, however, an ID of the string type is required to uniquely identify a vertex. To use the Cypher syntax in GES, add **_ID_** to specify the ID of a vertex in the create statement. For example, the **create(n{_ID_:'123456'})** statement creates a vertex whose ID is 123456.
- If the ID is not specified, a random ID is generated for the vertex.

 NOTE

The **_ID_** identifier is supported only in the create statement. The match and set clauses do not support the **_ID_** identifier. In the match clause, you can use the **id()** function to obtain the vertex ID.

7.1.16.4 Supported Expressions, Functions, and Procedures

Expression

Cypher queries support multiple expressions and can be used in combination to form various filter criteria. Currently, the following expressions are supported:

Operation Type	Expression	Example
Logical operations	and	match (n:user) where n.age='Under 18' and n.gender='F' return n
	or	match(n:user) where n.`Zip-code`='22181' or n.userid=6 return n
	not	match(n:movie) where not n.genres contains 'Drama' return n
Null value judgment	is null	match (n) where n.userid is null return n
	is not null	match (n) where n.userid is not null return n
Comparison calculation	>,>=,<,<=,=,<>	match(n:user) where n.userid>=5 return n

Operation Type	Expression	Example
Arithmetic operators (2.3.10)	+, -, *, /, %, ^	return (1+3)%3
String comparisons	starts with	match(n:movie) where n.genres starts with 'Comedy' return n
	ends with	match(n:movie) where n.genres ends with 'Drama' return n
	contains	match(n:movie) where n.genres contains 'Drama' return n
List-related operation	in	match(n:student) where 'math' in n.courses return n
	[]	match(n:user) return n['userid'] with [1, 2, 3, 4] as list return list[0] with [1, 2, 3, 4] as list return list[0..1] match p=(n)-->(m) return [x in nodes(p) where x.gender='F' id(x)]
Date expressions (2.3.10)	.year, .month, .day, .hour, .minute, .second, .dayOfWeek	Year, month, and day of a specific date: with '2000-12-27 23:44:41' as strVal with datetime(strVal) as d2 return d2.year, d2.month, d2.day, d2.hour, d2.minute, d2.second, d2.dayOfWeek, d2.ordinalDay

NOTE

The where clause in Cypher queries does not support regular expressions.

Function

Cypher supports the following functions for grouping, aggregation, and vertex and edge operations:

NOTE

Cypher queries are case-insensitive when it comes to input functions, meaning they do not distinguish between uppercase and lowercase letters.

1. Aggregate

Aggregate functions **count** and **collect** are supported.

Function	Earliest Version Supported	Description	Example
count	2.2.17	Returns the total number of results.	match (n) return count(*) match (n) return count(n.userid)
collect	2.2.17	Collects results into a list.	match (n:movie) return n.genres, collect(n) as movieList
sum	2.3.3	Returns the sum of values.	unwind [1, 2.0, 3] as p return sum(p)
avg	2.3.3	Returns the average of values.	unwind [1, 2.0, 3] as p return avg(p)
min	2.3.3	Returns the minimum value.	unwind [1, 2.0, 3] as p return min(p)
max	2.3.3	Returns the maximum value.	unwind [1, 2.0, 3] as p return max(p)

2. Regular functions

Based on the types of input parameters, regular functions are classified into vertex and edge functions, path functions, list functions, and value functions.

Table 7-303 Vertex and edge functions

Function	Earliest Version Supported	Description	Example
id	2.2.16	Obtains the ID of a vertex.	match (n) return id(n)
labels	2.2.16	Obtains labels of a vertex.	match (n) return labels(n)
type	2.2.16	Obtains the label of an edge.	match(n)-[r]->(m) return type(r)
degree	2.2.26	Obtains the degree of a vertex.	match (n) where id(n)='Vivian' return degree(n)
inDegree	2.2.26	Obtains the indegree of a vertex.	match (n) where id(n)='Vivian' return inDegree(n)

Function	Earliest Version Supported	Description	Example
outDegree	2.2.26	Obtains the outdegree of a vertex.	match (n) where id(n)='Vivian' return outDegree(n)
startNode	2.3.10	Obtains the start vertex of an edge.	match (n)-[r]->(m) return startNode(r)
endNode	2.3.10	Obtains the end vertex of an edge.	match (n)-[r]->(m) return endNode(r)

Table 7-304 Path functions

Function	Earliest Version Supported	Description	Example
nodes	2.2.19	Obtains the list of vertices on a path.	match p=(n)-[:friends*1..2]->(m) return nodes(p)
relationships	2.2.19	Obtains the list of edges on a path.	match p=(n)-[:friends*1..2]->(m) return relationships(p)
length	2.2.19	Obtains the path length.	match p=(n)-[:friends*1..2]->(m) return length(p)

Table 7-305 List functions

Function	Earliest Version Supported	Description	Example
head	2.3.10	Obtains the first element of a list.	with [1,2,3,4] as list return head(list)
last	2.3.10	Obtains the last element of a list.	with [1,2,3,4] as list return last(list)
size	2.3.10	Obtains the list length.	with [1,2,3,4] as list return size(list)

Function	Earliest Version Supported	Description	Example
range	2.3.10	Generates a list.	return range(1,5), range(1,5,2)

Table 7-306 Value functions

Function	Earliest Version Supported	Description	Example
toString	2.2.21	Converts a value to a string.	match (n) where toString(labels(n)) contains 'movi' return n
toUpper	2.2.26	Converts a string into uppercase letters.	match (n:movie) return toUpper(n.title)
toLowerCase	2.2.26	Converts a string into lowercase letters.	match (n:movie) return toLower(n.title)
toInteger	2.2.29	Converts a string to an int number.	with '123' as p return toInteger(p)
toLong	2.2.29	Converts a string to a long number.	with '123' as p return toLong(p)
toFloat	2.2.29	Converts a string to a float number.	with '123.4' as p return toFloat(p)
toDouble	2.2.29	Converts a string to a double number.	with '123.4' as p return toDouble(p)
toBoolean	2.2.29	Converts a string to a bool value.	with 'true' as p return toBoolean(p)
size	2.2.29	Obtains the string length.	with 'GES' as p return size(p)
subString	2.3.10	Truncates a part of a string.	return subString('abc', 1), subString('abcde', 1,2)

coalesce	2.3.10	Obtains the first non-null value of the parameters.	return coalesce(null, '123')
trim	2.3.11	Removes whitespace characters on both sides of a string.	return trim(' hello ')
lTrim	2.3.11	Removes whitespace characters on the left side of a string.	return lTrim(' hello')
rTrim	2.3.11	Removes whitespace characters on the right side of a string.	return rTrim('hello ')
reverse	2.3.11	Returns a string with the characters in reverse order.	return reverse('hello')
left	2.3.11	Obtains several characters from the left side of a string.	with 'hello' as p return left(p, 3)
right	2.3.11	Obtains several characters from the right side of a string.	with 'hello' as p return right(p, 3)
replace	2.3.11	Replaces a string.	with 'hello' as p return replace(p, 'll', 'o')
split	2.3.11	Splits a string.	with 'hello' as p return split(p, 'e')

Table 7-307 Mathematical functions

Function	Earliest Version Supported	Description	Example
floor	2.3.10	Rounds a number down to the nearest integer.	return floor(4.1)
ceil	2.3.10	Rounds a number up to the nearest integer.	return ceil(4.1)
round	2.3.14	Round	return round(3.4), round(3.5)
abs	2.3.14	Absolute value function	return abs(-3),abs(-3.5)
sin	2.3.14	Sine function	return sin(pi()/2)
cos	2.3.14	Cosine function	return cos(0),cos(pi()/2)
tan	2.3.14	Tangent function	return tan(pi()/4)
acos	2.3.14	Inverse cosine function	return acos(1)
asin	2.3.14	Inverse sine function	return asin(0)
atan	2.3.14	Inverse tangent function	return atan(1)
cot	2.3.14	Cotangent function	return cot(pi()/4)
radians	2.3.14	Converts degree to radian	return radians(180)
degrees	2.3.14	Converts radian to degree	return degrees(pi())
pi	2.3.14	Returns the approximate value of Pi (π).	return pi()

Table 7-308 Date and time functions

Function	Earliest Version Supported	Description	Example
<code>datetime(val)</code>	2.3.10	Returns the time based on the timestamp.	<code>return datetime(1688696395)</code>
<code>datetime()</code>	2.3.14	Obtains the current time (valid only for read statements).	<code>return datetime()</code>
<code>timestamp(val)</code>	2.3.10	Returns the timestamp based on the time string.	<code>return timestamp('2023-07-07 02:20:42')</code>
<code>timestamp()</code>	2.3.14	Obtains the current timestamp (valid only for read statements).	<code>return timestamp()</code>
<code>localDatetime</code>	2.3.14	Converts a time or timestamp to a local time string.	<code>return localDatetime(timestamp())</code>

Table 7-309 Predicate functions

Function	Earliest Version Supported	Description	Example
<code>all</code>	2.2.19	If all elements meet the expression, true is returned.	<code>all (x in p where x>1)</code>
<code>any</code>	2.2.19	If any element meets the expression, true is returned.	<code>any (x in p where x>1)</code>

Function	Earliest Version Supported	Description	Example
none	2.2.19	If all elements cannot meet the expression, true is returned.	none (x in p where x>1)
single	2.2.19	If only one element meets the expression, true is returned.	single (x in p where x>1)

Table 7-310 Algorithm expressions

Function	Earliest Version Supported	Description	Example
shortestPath	2.3.2	Returns the shortest path between two vertices.	The following statement returns the shortest path between the given vertices n and m . The direction is from m to n, and the edge label is rate : with n,m, shortestPath((n)<-[rate*]-(m)) as p return p
allShortestPaths	2.3.2	Returns all shortest paths between two vertices.	The following statement returns all shortest paths between the given vertices n and m : with n,m, allShortestPaths((n)-[*]-(m)) as p return p

Procedure

Currently, GES supports the following procedures.

Procedure	Statement
Obtaining graph pattern information	call db.schema()
Obtaining vertex labels	call db.labels()

Procedure	Statement
Obtaining edge labels	call db.relationshipTypes()
Querying the Cypher statements that are being executed	call dbms.listQueries()
Terminating a Cypher statement based on queryId	call dbms.killQuery('queryId')
Querying indexes	call db.indexes()
Full-text indexing for querying vertices that meet the search criteria	call db.index.fulltext.queryNodes()
Full-text indexing for querying edges that meet the conditions	call db.index.fulltext.queryRelationships()

 NOTE

Full-text indexes support six types of queries: prefix, wildcard, regexp, fuzzy, match, and combine. To use full-text indexes, you need to call the API for creating a full-text index.

 NOTE

Function and procedure names are case sensitive and must be in lower camel case.

- **Example of a full-text index query request**
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query
{
 "statements": [
 {
 "statement": "call db.index.fulltext.queryNodes('combine', {title:'1977'}) yield node, score
return node, score skip 1 limit 10",
 "resultDataContents": [
 "row"
],
 "parameters": {}
 }
]
}
- **Parallel edge processing policy**
When using Cypher to add edges, you can add duplicate edges. Duplicate edges are two edges with the same source vertex and target vertex.
- **How to add an edge without a label**
When you use a Cypher statement to add an edge, set the label of the edge to the default value **__DEFAULT__**. For example, **create ()-[r:__DEFAULT__]->() return r**.

Querying the Schema Structure Using Cypher

- **Function**
Query the generated schema structure.

- Query statement
 - Name: Schema structure query
 - Statement: **call db.schema()**
 - Note:

If you did not call the API for generating the schema structure, the returned schema file contains all labels.

If you have called the API for generating the schema structure, this API returns the labels as the vertices and the relationships between the labels as edges.

7.1.17 Domain-Specific Language (DSL) Query APIs (2.3.14)

7.1.17.1 Executing DSL Algorithms

Function

This API is used to provide flexible and controllable DSLs to help users design and run algorithms at low costs. For details about the DSL algorithm, see the syntax introduction in [DSL Syntax](#).

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=algorithm-query

Table 7-311 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For how to obtain a project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-312 Body parameter

Parameter	Mandatory	Type	Description
commands	Yes	String	Command executed by the custom operation. For details about the syntax, see the syntax introduction.

Response Parameters

Table 7-313 Response parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	Object	Query results. This parameter is left blank when the query fails.

Example Request

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=algorithm-query
{
  "commands": "Match<Vertex> v(['1']); v.repeat(bothV()).limit(2); return v;"
}
```

Example Response

Example response for a successful request

Status code: 200

```
{
  "data": {
    "vertices": [
      {
        "id": "1",
        "label": "movie",
        "properties": {
          "genres": [
            "Comedy"
          ],
          "movieid": [
            1
          ],
          "title": [
            "Airplane! (1980)"
          ]
        }
      }
    ],
    "runtime": 0.126476598
  }
}
```

Status code: 400

Example response for a failed request

```
{
  "errorCode": "GES.8814",
  "errorMessage": "Unsupported API."
}
```

Status Codes

Table 7-314 Return code for failed requests

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

7.1.17.2 DSL Syntax

Syntax Introduction

The Algorithm query API is a graph DSL provided by GES. You can use it to query and calculate graphs. During service planning, various query operators and fine-grained basic computing pattern operators are added to DSL so that DSL can support user-defined graph traversal, multi-hop filtering query, pattern matching, similarity algorithms, community algorithms, recommendation algorithms, path analysis, and custom service algorithms.

For example, to query the neighborhood vertex set of the second hop from vertices 1 and 2, run the following command:

```
Match<Vertex> v(['1','2']); v.repeat(bothV()).times(2).limit(3); return v;
```

DSL Statement Construction

Statements are DSL program fragments that are executed in sequence. To enrich DSL's capability of expressing user-defined algorithms, DSL supports the following statements: selection statement, loop statement, expression statement, operator operand, declaration statement, and jump statement. Typically, the DSL structure consists of a declaration statement, an expression statement, and a jump statement. For example:

```
GlobalAcc<Sum,int> g=0; // Declaration statement, which declares an aggregator variable g
g+=1*3+2;               // Expression statement, which is used to perform the aggregation operation on the
                        aggregator g. g = g + 1 x 3 + 2
return g;               // Jump statement, which ends DSL and returns g
```

 NOTE

Currently, only the following selection and loop statements are supported, which are written in selection, loop, and lambda expressions:

1. Assignment/aggregation of various expressions
2. Update operator of the Vertex matcher
3. Gather operator of the Vertex matcher

Table 7-315 DSL syntax description

Syntax	Keyword	Description
Select statement	if	Executes a statement conditionally.
Loop statement	while	Executes statements repeatedly.
Jump statement	return	Returns different types of results.
Declaration statement	Match< ? >	Declares a matcher Match to quickly match graph data for more operations.
	VertexAcc< ?, ?>	Declares a vertex aggregator to define additional properties/variables on a vertex. Match can be used to perform operations on additional variables on vertices in batches.
	GlobalAcc< ?, ?>	Declares a global aggregator to define globally operable variables.
Expression statement	operator=	Assigns values to variables.
	operator+=	Variable aggregation operation
	Arithmetic operations +, -, *, /	Arithmetic operations
	Comparison operations >, >=, ==, <, <=, !=	Comparative operations
	lambda expression	Anonymous expressions
Operator operation statement	repeat	Allows users to perform multi-hop filtering query. The syntax includes emit, times, and limit.
	update	Performs a set of batch operations defined on the matcher
	gather	Performs a set of batch operations defined on the matcher
	pick	Quickly obtains <i>N</i> random vertices

Syntax	Keyword	Description
	init	Reinitializes the vertex set of the vertex matcher
	insert	Adds a matching vertex set to the vertex matcher
	move	Quickly moves the vertex set in other_match_vertex to the match_vertex matcher
	subgraph	It is mainly used in the return statement. It can return guidance subgraphs of the match_vertex vertex matcher.
	intersection	Takes the intersection of two vertex matchers
	pattern	Executes a complete Cypher statement and places the result in the vertex matcher

1. Select statement: if

```
if(expression) {  
    true Branching statement  
}
```

Executes a statement conditionally.

```
GlobalAcc<Sum, int> threshold = 10;  
Match<Vertex> v(['1', '2']);  
if(threshold < 20) {  
    v.repeat(outV().has('name', 'peter')).times(2).emit();  
}
```

2. Loop statement: while

```
while(expression) {  
    true Branching statement  
}
```

Executes some code repeatedly and conditionally

```
GlobalAcc<Sum, int> loop = 0;  
VertexAcc<Sum, int> score = 1;  
Match<Vertex> v(['1', '2']);  
while(loop < 10) {  
    loop += 1;  
    v.update((v)->{v.score = 1 + 2 * v.score;});  
}
```

3. Jump statement: return

Returns JSON files of different types and formats

Type	Return Type	Description
Match	Vertex set	-

Type	Return Type	Description
VertexAcc	map	By default, values that are not changed after initialization are not output.
GlobalAcc	Single value	-
subgraph	Vertex set and edge set	-

If some vertices are matched, they can be directly returned:

- commands statement:

```
Match<Vertex> v(['1','2']);  
return v;
```

Response:

```
{  
  "vertices": [  
    {  
      "id": "1",  
      "label": "movie",  
      "properties": {  
        "genres": [  
          "Comedy"  
        ]  
      }  
    },  
    ...  
  ]  
}
```

- Alternatively, ACC values can be directly returned.

```
GlobalAcc<Max,int> g1=10;  
g1+=2;  
return g1;
```

Response:

```
{  
  "data": {  
    "value": 10  
  }  
}
```

- Assume that you want to obtain the guidance subgraph of a match vertex set for subsequent task execution (for example, drawing on the canvas).

```
Match<Vertex> v(['1','2']);  
return v.subgraph();
```

Response:

```
{  
  "data": {  
    "vertices": [  
      {  
        "id": "1",  
        "label": "user",  
        "properties": {  
          //balabala  
        }  
      },  
      {  
        "id": "2",  
        "label": "user",  
        "properties": {  
          //balabala  
        }  
      }  
    ]  
  }  
}
```

```
    "label": "movie",
    "properties": {
      //balabala
    }
  },
  "edges": [
    {
      "index": "0",
      "source": "1",
      "label": "rate",
      "properties": {
        //balabala
      },
      "target": "2"
    }
  ]
}
```

4. Expression statement

An expression is a sequence of operators and operands, which specifies a calculation. Different statements may use expressions. For example, loop statements and selection statements may use expressions to express branches or loop conditions. Expressions can also be used to assign values to variables and perform aggregation operations.

- The following table lists common operators:

Common Operator	Form	Symbol
Assignment	a=b;	=
Aggregation	a+=b	+=
Mathmetical	a+b;a-b;a*b;a/b	+, -, *, /
Comparison	a>b; a=b;a<=b;a==b;a!=b	>, >=, ==, <, <=, !=

NOTE

The calculation of aggregation operations is determined by the definition of [Aggregator](#).

- The following table lists the priorities among expressions:

Priori ty	Operator	Combination
1	() : Function call [] : Subscript .: Member access a++ : Suffix increment a-- : Suffix decrement	Left to right

Priori ty	Operator	Combination
2	a*b a/b Multiplication and division	Left to right
3	a+b a-b Addition and subtraction	Left to right
4	< <= > >= Relational operators	Left to right
5	== != Equality operators	Left to right
6	and Logical AND	Left to right
7	or Logical OR	Left to right
8	= += Assignment, aggregation operations	Right to left

– Expression types

Due to the particularity of variables (different types of aggregators), DSL classifies expressions into the following types:

- i. numeric expression - Numerical expression
- ii. vertexacc expression - Expression containing the vertexacc variable
- iii. globalacc expression - Expression containing the globalacc variable

```
1+2*3 // Numerical expression
```

```
GlobalAcc<Sum, int> diff = 0;  
VertexAcc<Sum, int> score = 0;  
Match<Vertex> v(['Tom']);
```

```
1+diff*2 // Expression with globalAcc, which belongs to globalacc expression  
1+v.score*2 // Expression with vertexAcc, which belongs to vertexacc expression
```

– Expression upgrade table

There are different constraints on assignment and calculation for different types of expressions. That is, whether the assignment and calculation between different types are valid is restricted.

The following expressions are used to indicate whether assignment/aggregation operations support different expressions.

Left Value/ Right Value	Numeric Expression	VertexAcc Expression	GlobalAcc Expression
vertexacc variable	vertexacc expression	Partially supported, the same matcher required	vertexacc expression
globalacc variable	globalacc expression	Not supported	globalacc expression

- **Value promotion strategy**

During the calculation of an expression, pure numerical values can be converted into other types of values. The following lists only the supported value promotion strategies:

- Integer value promotion: Boolean values can be converted to integer values. The value **false** changes to **0**, and the value **true** changes to **1**.
- Floating-point value promotion: Pure numerical values of the floating-point type can be converted to values of the double type without changing the values.

- **Value conversion**

Value conversion is currently not supported. Unlike promotion, value conversions can change values and have potential precision losses.

- **Lambda expressions**

```
($parameters)->{statements;}
```

It is mainly used to receive some steps of a function. For example, both the gather and update steps support transferring lambda expressions as parameters. Variables such as **GlobalAcc** and **VertexAcc** are used in lambda expressions to assign and aggregate values. Note the following:

- The parameter type does not need to be declared and can be identified in a unified manner.
- GlobalAcc in the context can be accessed by lambda expressions.
- Lambda expressions can be executed as independent function bodies. The input parameters must be empty.

```
GlobalAcc<Sum,int> g1=0;
g1+=2*10+1;
()->{g1+=2*10+1;} // Same result as the previous statement

// update receives lambda as a parameter:
v.update((v1)->{g1+=v1.acc1*2+g2+v1.acc2;});
```

Variable Declaration

Match matcher

1. Match matcher description

DSL allows you to define objects such as vertices, paths, and submaps that can be matched and on which related operations can be performed. After declaring and initializing the matcher, you can use the matcher operator to perform operations on vertex sets in batches. For details, see [Match matcher operators](#). Run the following statement to declare and initialize the matcher:

```
Match<[Vertex|Path|Subgraph]> $variable;
```

DSL allows you to define vertices, paths, and submaps that can be matched and on which related operations can be performed.

```
Match<Vertex> Quickly match and perform operations on vertices
```

2. Match<Vertex> matcher: Match matcher that can be used to quickly match and perform operations on vertex sets

```
Match<[ Vertex ]> $match_vertex_variable;
Match<[ Vertex ]> $match_vertex_variable($VertexList);
```

You can use Match<Vertex> to match or initialize vertex sets.


```
Match<Vertex> v(['1', '2']); // Directly match by vertex ID.  
Match<Vertex> v();          // Only Match variables are defined. The vertices to be matched are empty.  
Match<Vertex> v;  
v.pattern('match (n:user) where n.age>30 return n limit 10'); // Uses the results of the Cypher  
statements to match the filtered vertices.
```

DSL provides a large number of operators for Match<Vertex>, such as **pattern**, **init**, and **pick**. For details, see [Match matcher operators](#).

3. Relationship between match matcher and aggregators

You can define an aggregator to hold values and calculations. Each type of matcher can perform operations on its corresponding aggregator in batches.

For example, you can use Match<Vertex> to perform aggregation calculation on a specified vertex set.

Aggregator

1. Aggregator description

Aggregators are used to simplify the expression of values during calculation. Different types of aggregators define different scopes of batch aggregation operations.

NOTE

DSL does not support direct definition of variables of various numeric types to carry data generated during calculation. Instead, DSL provides aggregators to simplify operations in various application scenarios.

1. VertexAcc can be used to define additional "properties (variables)" on vertices, and Match<Vertex> can be used to perform operations on additional variables on vertices in batches.
2. GlobalAcc can be used to define globally operable variables.

Aggregator declaration syntax:

```
Accumulator<Aggregator Operator, NumericType> v;
```

Parameter descriptions:

NumericType: int, (float and double types are not supported currently)

Aggregator Operator: Sum,Max,Min

DSL allows you to define different aggregators to simplify algorithm operations. Currently, two types of aggregate variables are supported:

- a. VertexAcc<Aggregator Operator, type>
- b. GlobalAcc<Aggregator Operator, type>

You can perform the following operations on an aggregator:

- a. Initialization: Define an aggregator and assign initial values.
- b. Assignment: Reset the aggregator values.
- c. Aggregation: Perform aggregation operations based on defined aggregator operators.

2. Aggregator operator description

Different aggregation operations are provided, such as Sum, Max, and Min. Variables are updated using operators + and =.

```
$match_vertex.$vertex_accumulator += $value;  
$global_accumulator += $value;
```

3. Vertex aggregator VertexAcc<Aggregator Operator, type>

VertexAcc can be used to quickly define additional "property (variables)" on vertices. and Match<Vertex> can be used to perform operations on additional

variables on vertices in batches. This significantly facilitates the calculation process.

```
VertexAcc<Sum, float> score = 0.5;
```

Each time a VertexAcc is defined, DSL allocates a variable of the type type to each vertex in the full graph. You can use Match<Vertex> to perform operations on the defined VertexAcc. For example:

```
VertexAcc<Sum, int> score = 0;
// Tom.score = 0, Jack.score = 0
Match<Vertex> v(['Tom', 'Jack']);
// Tom.score = 0, Jack.score = 0
v.score += 1; // This operation aggregates the scores of Tom and Jack. That is, their scores are
              // increased by 1.
// Tom.score = 1, Jack.score = 1
v.score = 10; // This operation assigns values to the scores of Tom and Jack. That is, their scores are
              // updated to 10.
// Tom.score = 10, Jack.score = 10
v.score += 5; // This operation aggregates the scores of Tom and Jack. That is, their scores are
              // increased by 5.
// Tom.score = 15, Jack.score = 15
```

VertexAcc can be operated by expressions. Currently, numeric, GlobalAcc, and VertexAcc expressions can be used for aggregation and assignment.

```
VertexAcc<Sum, int> score = 0;
VertexAcc<Sum, int> factor = 1;
GlobalAcc<Sum, int> alpha = 10;
Match<Vertex> v(['Tom']);
// Tom.score = 0.
v.score = alpha x 2 + 3; // This operation assigns a value to Tom's score.
// Tom.score = alpha x 2 + 3 = 10 x 2 + 3 = 23

v.score += v.factor*2; // This operation assigns a value to Jack's score.
// Tom.score = Tom.score + Tom.factor x 2 = 23 + 1 x 2 = 25
```

4. Global aggregator GlobalAcc<Aggregator Operator, type>

Each time a GlobalAcc is defined, DSL creates a variable of the type type in the DSL scope. Then, perform operations on GlobalAcc directly. For example:

```
GlobalAcc<Sum, int> diff = 0; // Define a GlobalAcc.
diff += 1; // Perform an aggregation operation, that is, diff = Sum (0,1)
// diff = 1
diff = 2 x 3; // Assign a value, that is, diff = 2 x 3
// diff = 6
GlobalAcc<Sum, int> g2 = 6; // Define a GlobalAcc.
diff += g2; // Perform an aggregation operation, that is, diff = Sum(6,g2)
// diff = 12
```

GlobalAcc can be operated by expressions. Currently, numeric and GlobalAcc expressions can be used for aggregation and assignment.

```
GlobalAcc<Sum, int> alpha = 0;
GlobalAcc<Sum, int> beta = 10;
alpha = beta*2+3;
// alpha = beta x 2 + 3 = 10 x 2 + 3 = 23
```

Operator Introduction

Match matcher operators

After a **Match matcher** is defined, different operation operators, such as repeat, gather, update, and pattern, can be used to assist graph calculation and query.

The operation operator varies with the match type.

Operator	Description	Matcher	Anonymous Expression	Select/Loop Statement
repeat	Performs multi-hop filtering like Gremlin.	Vertex	×	×
pick	Randomly selects vertices.	Vertex	×	×
pattern(2.3.11)	Runs Cypher statements.	Vertex	×	×
update	Performs a set of batch operations defined on the matcher.	Vertex	√	√
gather	Performs a set of batch operations defined on the matcher.	Vertex	√	√
init(2.3.12)	Initializes the matching vertex set based on a specified ID.	Vertex	×	×
insert(2.3.12)	Add new vertices to the matcher.	Vertex	×	×
move(2.3.12)	Quickly moves other matcher vertex sets.	Vertex	×	×
intersection(2.3.12)	Calculates the intersection of vertices.	Vertex	×	×
subgraph(2.3.12)	Obtains guidance subgraphs.	Vertex	×	×

- **repeat**

It allows you to perform multi-hop filtering query. The syntax of repeat is similar to that of repeat in Gremlin. Its semantic expression ability is rich, especially suitable for relational link query.

For example, if a two-hop query of filtering name=peter is executed from vertex (1,2) to the outgoing direction, it may be expressed in Gremlin as:

```
g.V('1','2').repeat(out().has('name','peter')).times(2).emit().dedup()
```

In DSL, the preceding functions can be written as follows:

```
Match<Vertex> v(['1','2']);  
v.repeat(outV().has('name','peter')).times(2).emit();
```

- The repeat step contains some unique associated steps:

Parameter	Mandatory	Type	Default Value	Description
repeat	Yes	Traversal step	None	Executes the repeat rule.
times	No	Integer	2	Maximum number of steps. The default value is 2 , and the maximum value is 20 .
emit	No	Boolean	false	Whether all elements will be returned. The default value is false . If select-as or path is set for output, this parameter determines whether vertices that are not on the final complete path will be returned.
limit	No	Integer	10000	Number of vertices, edges, or paths.

- Rules in **repeat** consist of traversal and filter criteria. They can be multiple rules separated by commas (,).

Table 7-316 Traversal process description

Step	Description
outV	Neighbor vertex in the outgoing direction
inV	Neighbor vertex in the ingoing direction
bothV	Neighbor vertex in both directions
outE	Edge in the outgoing direction
inE	Edge in the ingoing direction

Step	Description
bothE	Edge in both directions
otherV	Neighborhood vertex

Table 7-317 Filter criteria description

Filter Criterion	Description
has(key)	Whether the property name key exists
has(key, value)	Whether the value of the property name key is value
hasLabel(values) (V2.3.5)	Whether the label value is one of values
and(filter operator A, filter operator B)	Logical operator of filter criteria. Criteria A and B must be both met. They can be nested.
or(filter operator A, filter operator B)	Logical operator of filter criteria. Either criterion A or B needs to be met. They can be nested.

```
and(has('person'), or(has('name', 'peter'), has('age', '30')))
```

```
has('person') // The property name person exists.  
has('name', 'peter') // The value of the property name name is peter.  
hasLabel('movie','user') // The label value is movie or user.  
and( has('name', 'peter'), has('age', '30')) // The property name is peter and age is 30.
```

- **update**

```
$match_vertex.update($lambda_func);
```

The update operator is used to define a group of batch operations on the matcher. Currently, only Match<Vertex> is supported.

The Update operation on Match<Vertex> applies all operations defined in the input lambda function to the vertices matched by Match.

The vertex matcher Match<Vertex> receives only lambda expressions that contain one input parameter.

```
Match<Vertex> v(['1','2']);  
VertexAcc<Max,int> acc=1;  
GlobalAcc<Sum,int> g=0;  
v.update( (v1)->{g+=v1.acc*2;} ); //g='1'.acc*2+'2'.acc*2=1*2+1*2=4  
return g; // Return g = 4;
```

- **gather**

```
$match_vertex.gather($lambda_func);
```

The gather operator is used to define a group of batch operations on the matcher. Currently, only Match<Vertex> is supported.

- **gather of Match<Vertex>**

The Gather operation on Match<Vertex> applies all operations defined in the input lambda function to the edges of the vertices matched by Match.

The vertex matcher `Match` receives only lambda expressions that contain two input parameters. The first parameter indicates the source vertex on the edge, and the second parameter indicates the target vertex on the edge.

```
Match<Vertex> v(['1','2']);
VertexAcc<Max,int> acc=1;
GlobalAcc<Sum,int> g=0;
v.gather( (s,t)->{g+=s.acc*2+t.acc;} ); // This operation takes effect on the edges of vertices 1
and 2. That is, 1-2, 1-3, 2-3, 2-1.
//g=g + 1.acc*2+2.acc + 1.acc*2+3.acc + 2.acc*2+1.acc + 2.acc*2+3.acc
//g=0 + 3 + 3 + 3 + 3=12
return g;
```

- **pick**

```
$match_vertex.pick(n);
```

pick allows you to randomly select *N* vertices from `Match<Vertex>`. This function helps you quickly obtain *N* random vertices.

```
Match<Vertex> v();
v.pick(10); // Select 10 vertices randomly from v.
return v; // Return information about 10 random vertices.
```

- **init(2.3.12)**

```
$match_vertex.init([vertex_list]); // Specify the vertex ID list.
$match_vertex.init(vertexid); // Specify a single ID.
```

Resets the vertex set of the vertex matcher.

```
Match<Vertex> v(['1','2']);
return v; // Return information about vertices 1 and 2.
Match<Vertex> v(['1','2']);
v.init(['3','4']);
return v; // Return information about vertices 3 and 4.
```

- **insert(2.3.12)**

```
$match_vertex.insert([vertex_list]); // Specify a list of vertex IDs.
$match_vertex.insert(vertexid); // Specify a single ID.
```

Adds a matching vertex set to the vertex matcher.

```
Match<Vertex> v(['1','2']);
return v; // Return information about vertices 1 and 2.

Match<Vertex> v(['1','2']);
v.insert(['3','4']);
return v; // Return information about vertices 1, 2, 3, and 4.
```

- **move(2.3.12)**

```
$match_vertex.move(other_match_vertex);
```

Quickly moves the vertex set in **other_match_vertex** to the **match_vertex** matcher. This operator can assign a value to another matcher by one matcher, but does not produce a replication result. Instead, it is similar to the move semantics in C++, that is, moving the original matcher vertex set to the new matcher. After the operation, the original matcher vertex set is cleared.

```
Match<Vertex> v1(['1','2']);
Match<Vertex> v2(['3','4']);
v1.move(v2); // v1=[3,4], v2=[]
return v1; // Return information about vertices 3 and 4.
```

- **subgraph(2.3.12)**

```
$match_vertex.subgraph(); // Return guidance subgraphs of the matcher vertex set.
$match_vertex.subgraph(filter_step); // Return guidance subgraphs of the matcher vertex set with
edge filtering.
```

It is mainly used in the return statement. It can return the induced subgraph of the **match_vertex** vertex matcher.

When obtaining a subgraph, you can set the filter criterion **filter_step** on the edge. For details about the syntax, see the filter operator in **repeat**.

```
Match<Vertex> v(['1','2']);  
return v.subgraph();// Return vertices 1 and 2 as well as the edge set between vertices 1 and 2.
```

```
Match<Vertex> v(['1','2','3']);  
return v.subgraph(has(year, 2022));// Return vertices 1 and 2 as well as the edge set between vertices 1 and 2.
```

- **Set basic operations (2.3.12)**

Set operations can be performed on objects in each matcher, such as union and intersection.

intersection (2.3.12)

```
Match<Vertex> start(['1','2']);  
Match<Vertex> target(['2']);  
Match<Vertex> set;  
start.intersection(target);  
return start;// Return vertex 2.
```

- **Pattern Matching: pattern (2.3.11)**

DSL uses the Cypher syntax to express pattern matching, such as vertex set, path, and subgraph matching.

Statements that support Cypher:

Statement	Supported or Not
match(2.3.11)	Partially supported
limit(2.3.11)	Supported
return(2.3.11)	Supported

DSL uses the step-pattern of the Match variable to implement pattern matching.

For example, you need to use a certain rule to match vertices.

```
Match<Vertex> v;  
v.pattern('match (n:user) where n.age>30 return n limit 10');
```

The pattern is a complete Cypher statement, which filters 10 vertices and returns them to **Match<Vertex>**.

String in Cypher

The pattern syntax contains a complete Cypher statement in single quotation marks. When a string to be expressed, such as an ID or property value, appears in the Cypher statement, double quotation marks are required.

When using REST APIs to call DSL, you need to add the escape character \ to the double quotation marks. For example:

```
Match<Vertex> v;  
v.pattern('match (n) where id(n)=\"12\" return n');  
return v;
```

7.1.18 Graph Query Language (GQL) APIs (2.4.15)

7.1.18.1 Executing GQL Queries

Function

GQL is a declarative query language with its core syntax resembling Cypher. Use GQL to query data in GES and retrieve results. For detailed information on GQL queries, refer to [GQL Syntax](#).

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-gql-query

Table 7-318 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For how to obtain a project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-319 Body parameters

Parameter	Mandatory	Type	Description
statements	Yes	List	Statement group that contains one or more statements.

Table 7-320 statements parameters

Parameter	Mandatory	Type	Description
statement	Yes	String	GQL statement
parameters	No	Json	GQL statement parameters, which are used for parameterized queries. This parameter is left blank by default. For details, see parameterized queries .
resultDataContents	No	String or List	Format of the execution results. You can set one or more formats. The options are row , graph , and raw .

Parameter	Mandatory	Type	Description
includeStats	No	Boolean	Whether the returned result contains addition, deletion, and modification statistics. If this parameter is not set, the returned result does not contain the information by default.
runtime	No	String	Executor type. The default value is block .
executionMode	No	String	Execution mode. Set this parameter to sync for synchronous execution and to async for asynchronous execution. If this parameter is not set, the execution is synchronous by default. For how to obtain the query results in asynchronous mode, refer to section 5.10.1.
limit	No	Integer	Maximum number of asynchronous results. This parameter applies only when executionMode is async . The default value is 100000 .

 NOTE

1. You can add the **explain** or **profile** prefix before a statement to display the query plan. The **explain** prefix displays only the query plan but does not execute the statement. The **profile** prefix displays the query plan and executes the statement.
2. In asynchronous mode (**executionMode** is **async**), GQL query results of the memory edition can be exported to CSV files. This edition currently allows returning the following value types: vertex and edge single-value properties, vertex and edge IDs, and group count results. Object types cannot be exported in the current version. They are treated as null values in CSV files. For more details, refer to [Exporting Job Execution Results to Files \(2.2.1\)](#).

Response Parameters

Table 7-321 Response parameter descriptions

Parameter	Type	Description
results	List	Each element is the return result of a GQL statement. For details about the parameters, see Table 7-322 .
errors	List	Each element contains the code and message information in string form.

Table 7-322 Elements of the results parameter

Parameter	Type	Description
columns	List	Name of a returned field
data	List	Returned data value. Each element indicates a record. For detailed parameter descriptions, refer to Table 7-323 .
stats	Json	Returned statistics about addition, deletion, and modification operations. For detailed descriptions, refer to Table 7-324 .
plan	Json	If the GQL statement contains the explain or profile prefix, this field contains the query plan. Otherwise, this field is not displayed.
jobId	String	Asynchronous job ID if the request is executed asynchronously
jobType	Integer	Type of the asynchronous job if the request is executed asynchronously

Table 7-323 Elements of the data parameter

Parameter	Type	Description
row	List	Content of a specific row. Each element corresponds to a field in the row. This parameter is displayed only when resultDataContents is empty or contains row .
meta	List	Type of each field in a row. This parameter is displayed only when resultDataContents is empty or contains row .
graph	Json	Information returned in graph format. This parameter is displayed only when resultDataContents contains graph .
raw	List	Content of a specific row. Each element corresponds to a field in the row. This parameter is displayed only when resultDataContents contains raw .

Table 7-324 Elements of the stats parameter

Parameter	Type	Description
contains_updates	Boolean	Whether data is modified during the query
edges_created	Integer	Number of created edges
edges_deleted	Integer	Number of deleted edges
labels_set	Integer	Number of labels that have been set
properties_set	Integer	Number of properties that have been set
vertices_created	Integer	Number of created vertices
vertices_deleted	Integer	Number of deleted vertices

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-gql-query
{
  "statements": [{
    "statement": "match (n) return n limit 1",
    "runtime": "block",
    "parameters": {},
    "resultDataContents": ["raw"]
  }],
  "includeStats": true
}
```

Example Response

- Example response for a successful synchronous request

Status code: 200

Http Status Code: 200

```
{
  "results": [
    {
      "columns": ["n"],
      "data": [
        {
          "row": [
            {
              "occupation": "artist",
              "gender": "F",
              "Zip-code": "98133",
              "userid": 0,
              "age": "25-34"
            }
          ]
        }
      ],
      "meta": [
        {
          "id": "46",
          "type": "node",
          "labels": [
            "user"
          ]
        }
      ]
    }
  ]
}
```

```
    ],  
    "stats": {  
      "contains_updates": false,  
      "edges_created": 0,  
      "edges_deleted": 0,  
      "labels_set": 0,  
      "properties_set": 0,  
      "vertices_created": 0,  
      "vertices_deleted": 0  
    }  
  },  
  "errors": []  
}
```

- Example response for a successful asynchronous request

Status code: 200

Http Status Code: 200

```
{  
  "results": [  
    {  
      "columns": ["n"],  
      "data": [  
        {  
          "row": [  
            {  
              "occupation": "artist",  
              "gender": "F",  
              "Zip-code": "98133",  
              "userid": 0,  
              "age": "25-34"  
            }  
          ],  
          "meta": [  
            {  
              "id": "46",  
              "type": "node",  
              "labels": [  
                "user"  
              ]  
            }  
          ]  
        }  
      ],  
      "stats": {  
        "contains_updates": false,  
        "edges_created": 0,  
        "edges_deleted": 0,  
        "labels_set": 0,  
        "properties_set": 0,  
        "vertices_created": 0,  
        "vertices_deleted": 0  
      }  
    }  
  ],  
  "errors": []  
}
```

Status code: 400

Example response for a failed request

Http Status Code: 200

```
{  
  "results": [],  
  "errors": [  
    {  
      "code": "GES.8904",  
      "message": "Label index in vertices is not found."  
    }  
  ]  
}
```

```
    ]  
  }
```

Status Codes

Table 7-325 Return code for failed requests

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

7.1.18.2 GQL Syntax

GQL is a standard query language designed for graph databases. For details about the GQL standard, see the [GQL document](#) released by ISO.

Supported GQL Clauses

GES supports the following GQL clauses:

Table 7-326 Supported GQL clauses

Keyword	Support	Example	Remarks
insert	Only supported by the database edition	insert (n:movie{_ID_: '1a', genres: 'Comedy Drama'}) return n	-
set	Only supported by the database edition	match (n) where element_id(n)= '1a' set n.movieid=1 return n	-
remove	Only supported by the database edition	match (n) where element_id(n)= '1a' remove n.movieid return n	-

Keyword	Support	Example	Remarks
delete	Only supported by the database edition	match (n) where element_id(n)='1a' delete n	-
match	Partially supported	match (n where element_id(n)='1a')-->(m) return count(*)	For details, see Pattern Matching .
optional match	Partially supported	match (n) where id(n)='1a' optional match (n)-->(m) return count(*)	For details, see Pattern Matching .
let	Supported	let a=1, value b::long=2 return a,b	When type matching is not considered, it can be equivalently represented as the WITH semantics in Cypher.
for	Supported	for a in [1,2,3] return a	The offset/ordinality parameter is currently not supported. It can be equivalently represented as the UNWIND semantics in Cypher.
filter/ where	Supported	match (n) where element_id(n)='1a' return n	-
order by	Supported	match (n:movie) return n order by n.genres asc	-
limit/ offset	Supported	match (n:movie) return n skip 10 limit 10	-
union& union all	Supported	return 1 as a union return 2 as a	-
next yield	Supported	match (n:user) return n union all match (n:movie) return n next yield n as b match (b)-->(m) return m	Currently, the yield keyword must be included to declare variables used in later stages.

Keyword	Support	Example	Remarks
call	Partially supported	call db.indexes()	-

 NOTE

1. Currently, multi-graph joint query and the **USE GRAPH** semantics are not supported.
2. The **SELECT... FROM... GROUP BY... HAVING** clause is not supported.
3. The **at schema** operation is not supported.
4. Currently, only the **union** operation is supported for the result set. The set operation semantics such as **except**, **intersect**, and **otherwise** are not supported.
5. The keyword **call** can only be followed by a procedure name. Subqueries of the call style are not supported.
6. The memory edition does not support modification operations.

Pattern Matching

In GQL, pattern matching generally refers to Cypher. Patterns described in the following table that do not include the **match** keyword are graph patterns within the **match** clause.

Table 7-327 Pattern matching

Item	Description	Example	Supported Version
Vertex pattern	Indicates a group of vertices sharing the same feature.	(n:movie{title:'hello'})	2.4.15
		(n:movie where n.title='hello')	
		(n:movie user)	
Edge pattern	Supports edge patterns with direction, without direction, and with label/property filtering. Allows edge queries by specifying the IDs of both start and end vertices.	(n)-[r]-> (m), (n)<-[r]-(m), (n)-[r]-(m)	
		(n)-[r:rate{Rating:1}]- (m)	
		(n)-[r:rate where r.Rating=1]- (m)	
Path	Supports anonymous paths.	match (n)-[r]->(m)-->(s)	
	Supports named paths.	match p=(n)-[r]->(m)-->(s)	
	Variable-length paths	match p=trail (n)-[r]->{1,3}(m) where element_id(n)='46' return p	

Item	Description	Example	Supported Version
	Deduplicated variable-length traversals	match p=trail (n)-[r]->{1,3}(m) where element_id(n)='46' return distinct m	Not supported
	Nested paths	match (d:Stop) ((s:Stop)-[:NEXT]->(f:Stop) where s.time <> f.time){1,3} (a:Stop) RETURN d.departs AS departureTime, a.arrives AS arrivalTime	
Path mode	Supports walk, trail, simple, and acyclic path modes. See Path Mode for details.	match trail (n)-[r]-{1,3}(m), trail (n)--(m1) where element_id(n)='0' return count(*)	2.4.15
Other	Multiple patterns in the match clause	match (n)-[r]->(m), (m)-->(s) return count(*)	
	Multiple match statements	match (n)-[r]->(m) with m match (m)-->(s) return count(*)	

 NOTE

1. The **optional** keyword cannot be followed by a subquery enclosed in curly braces.
2. The **SHORTEST** and **GROUP** keywords cannot follow the **match** keyword.
3. The **keep** keyword is not currently supported.

Path Mode

In the **match** statement, you can input several graph patterns representing multiple paths, with each path separated by commas. GQL supports the following four path modes, which are configured as optional keywords before the graph pattern. If no keyword is configured, the default value **walk** is used.

Table 7-328 Path modes

Keyword	Description
walk	In the same path, both vertices and edges can be repeated.
trail	In the same path, edges cannot be duplicated.
simple	In the same path, intermediate vertices except the end vertex cannot be the same.

Keyword	Description
acyclic	In the same path, each vertex must be unique.

Supported Types

1. The mapping between basic data types in GQL and GES types is as follows:

Table 7-329 GQL basic types

GQL Type	GES Type	Remarks
bool	Boolean	-
string	string, char array, and enum	-
int32	Integer	-
int64	long	-
float	float	-
double	double	-
datetime	date	The smallest time unit supported in GES is second.

2. Additionally, there are some complex types in GQL, with their construction methods and GES support detailed as follows:

Table 7-330 GQL complex types

GQL Type	Support	Example
vertex/node	Supported	match (n) return n
edge/relationship	Supported	match (n)-[r]->(m) return r
array/list	Supported	return [1,2,3] as li
record	Supported	match (n)-->(m) return {start:id(n), end:id(m)}
path	Supported	match p=(n1)-[:friends]-{1,2}(n2) return p limit 10

 NOTE

1. Among the complex types mentioned in [GQL complex types](#), only **List** can be used to correspond to multi-valued properties in GES. Other types cannot be assigned as values to properties on vertices or edges using the **SET** statement.
2. Currently, the **path** type can be created using the **match** statement, but connecting multiple paths or constructing a path from vertices and edges is not yet supported.
3. Display construction and assignment for graph and table objects are currently unsupported.
4. Support for the **Duration** type compliant with the GQL standard is also unavailable.

Supported Expressions and Functions

1. GQL supports the expressions and functions in [Supported Expressions, Functions, and Procedures](#). GQL also supports the following functions.

Table 7-331 Functions supported by GQL

Function	Corresponding Cypher Function	Description	Example
collect_list	collect	Aggregate function, which collects results into a list.	match (n:movie) return n.genres, collect_list(n) as movieList
element_id	id	Obtains the vertex or edge ID.	match (n) where element_id(n)= '0' return n
character_length	size	Obtains the length of a string.	return character_length('abc')
path_length	length	Obtains the path length.	match p=TRAIL (n)--{1,2}(m) where id(n)='0' return path_length(p)

2. The following expressions are supported:

Table 7-332 Expressions supported by GQL

Expression	Description	Example
	Supports concatenation between lists and between strings.	return 'abc' 'de', [1,2] [3,4]

3. Supports GQL-style value type predicate expressions.

Table 7-333 Judgment expressions supported by GQL

Category	Type Keyword	Example
Basic type determination	Boolean, string, int, long, float, double, datetime, node, edge, path, list, record	let a=1 return a is ::int
Nested type determination	list	let a=[1,2] return a is ::list<int>
Union type	any Note: any<int string long> indicates that a type is one of int, string, and long.	for a in [1,2,'a1'] return a is ::any<int string long>

 NOTE

Currently, mod and trigonometric functions are not supported.

Index Support

Currently, GQL supports vertex indexes with **hasLabel** set to **true** and supports single-value matching and multi-value matching. For details about how to create and edit indexes, refer to [Index Operation APIs](#).

Example:

Index Type	Label	Index Attribute	Remarks
GlobalComposite-VertexIndex	true	movieid	Single-attribute index
GlobalComposite-VertexIndex	true	movieid, title	Multi-attribute index

The indexes in the above table serve as an example. The following queries can utilize the indexes:

Type	Matching Mode	Example
Single-attribute index	Exact match - single value	match (n:movie) where n.movieid = 0 return n
	Exact match - multiple values	match (n:movie) where n.movieid in [0, 1, 2] return n
	Range match	match (n:movie) where n.movieid > 0 return n

Type	Matching Mode	Example
		match (n:movie) where n.movieid >= 0 return n
		match (n:movie) where n.movieid < 0 return n
		match (n:movie) where n.movieid <= 0 return n
Multi-value index	Exact match	match (n:movie) where n.movieid = 0 and n.title=" return n

When performing **EXPLAIN** or **PROFILE** on a query statement that utilizes an index, the `NodeIndexSeek` operator is printed, indicating that the index was used for the query.

7.1.18.3 GQL Compatibility

For details about the GQL standard, refer to the [GQL document](#) published by ISO. GES provides minimal support for GQL (GQL minimum conformance) as follows:

1. Supports features GG02 "Graph with a closed graph type" and GG20 "Explicit element type names".
2. Supports Unicode 13.
3. Supports basic data types, including: STRING, BOOLEAN, INT, and FLOAT.
4. Supports a range of features. See the [Supported Features](#) for details.

Supported Features

The following table lists the GQL features supported by GES:

Chapter	Section	Syntax/Feature	Associated ID	Example
9 Procedure specification	9.1	<next statement>	GQ 20	match (n:user) return n union all match (n:movie) return n next yield n as b match (b)-->(m) return m
13 Data-modifying statements (supported only by the database edition)	13.2	<insert statement>	-	insert (n:movie{_ID_: '1a', genres: 'Comedy Drama'}) return n
	13.3	<set statement>	GD0 2	match (n) where element_id(n) = '1a' set n.movieid=1 return n

Chapter	Section	Syntax/Feature	Associated ID	Example
	13.4	<remove statement>	GD02	match (n) where element_id(n) = '1a' remove n.movieid return n
	13.5	<delete statement>	-	match (n) where element_id(n) = '1a' delete n
14 Query statement	14.2	union	GQ03	return 1 as a union return 2 as a
	14.4	<simple match statement>	-	match (n) return n
		<optional match statement>	-	match (n) where id(n)= '1a' optional match (n)-->(m) return count(*)
	14.6	<filter statement>	GQ08	match (n) where element_id(n) = '1a' return n
	14.7	<let statement>	GQ09	let a=1, value b::long=2 return a,b
	14.8	<for statement>	GQ10	for a in [1,2,3] return a
	14.11	<return statement>	-	match (n) return n.name as name, n.age
15 Procedure calling	15.3	<named procedure call>	GP04	call db.schema()
16 Comment elements	16.4	<graph pattern>	G004	For details, see Pattern Matching .
	16.5	<insert graph pattern>	-	insert (n:movie{ _ID_: '1a', genres: 'Comedy Drama'}) return n
	16.7	<quantified path primary>	G036	match p=(n)-[r]->{1,3}(m) where element_id(n)= 'xx' return p
		Non-local element pattern predicates	G041	let a = 'Jaws (1975) ' match (n:movie where n.title=a) return n

Chapter	Section	Syntax/Feature	Associated ID	Example
		<full edge pattern>	G043	match (n),(m) where element_id(n)= '100' and element_id(m)= '120' optional match (n)-[r]->(m) with n,r,m optional match (n)-[r1]-(m) with n,r,r1,m optional match (n)<-[r2]-(m) return r,r1,r2
	16.8	<label expression>	-	match (n:movie) return count(*)
		<label disjunction>	-	match (n:movie user) return n limit 10
		<label negation>	-	match (n:!movie) return count(*)
		<parenthesized label expression>	-	match (n:(!movie&!default) user) return count(*)
	16.11	<fixed quantifier>	-	For details, see "Variable-length path" in Pattern Matching .
		<general quantifier>	-	match p=(n)-[r]->{,3}(m) where element_id(n)= 'xx' return p
	16.13	<where clause>	-	Refer to sections 14.4 and 14.6.
	16.16	<order by clause>	-	match (n:movie) return n order by n.genres asc
	16.18	<limit clause>	GQ13	match (n:movie) return n skip 10 limit 10
	16.19	<offset clause>	GQ12	
19 Predicate	19.3	<comparison predicate>	-	let a=1,b=2 return a=b,a<>b,a<b,a>b,a<=b,a>=b
	19.5	<null predicate>	-	let value a::long=1 return a is null
	19.9	<label predicate>	G111	match (n) where element_id(n)= '0' return n:movie

Chapter	Section	Syntax/Feature	Associated ID	Example
20 Value expressions and specifications	20.4	<dynamic parameter specification>	-	See Basic Operations and Compatibility and 2.
	20.7	<simple case>	-	let a =2 return case a when 1 then a+1 else a+2 end
		<searched case>	-	let a =2 return case when a=1 then a+1 else a+2 end
	20.9	Horizontal Aggregation	GE09	match (n) return labels(n),count(distinct n),avg(n.age),max(n.age),min(n.age),sum(n.age)
	20.10	<element_id function>	G100	match (n) where element_id(n) = '1a' return n
	20.11	<property reference>	-	match (n) where element_id(n) = '1a' return n.movieid
		record property	-	return {name: 'Jay',age:1}.name
	20.15	<list concatenation>	-	return [1,2,3,4] [5,6,7,8]
	20.17	<list value constructor>	-	return [1,2,3,4]
	20.18	<record constructor>	GV45	return {name: 'Jay', age:10, details:{salary:1}}
	20.20	<boolean value expression>	-	return (true and false) or false
		Boolean Xor	GE07	return true xor false
	20.21	<numeric value expression>	-	return 1+2,1-2,1*2,1/2,-2
	20.22	<char length expression>	-	let value s::STRING = 'hello_world' return character_length(s)
		<path length expression>	GF04	match p=TRAIL (n)--{1,2}(m) where id(n)= '0' return path_length(p)

Chapter	Section	Syntax/Feature	Associated ID	Example
		size function	GF13	let list = [1,2,3,4] return size(list)

Supported GQL Optional Features

For optional GQL features, the GQL standard provides feature IDs and names. Currently, GES supports the following features:

Feature ID	Feature Name
G004	Path variables
G011	Advanced path modes: TRAIL
G036	Quantified edges
G041	Non-local element pattern predicates
G043	Complete full edge patterns
G060	Bounded graph pattern quantifiers
G100	ELEMENT_ID function
G111	IS LABELED predicate
GD01	Updatable graphs
GD02	Graph label set changes
GE07	Boolean XOR
GE09	Horizontal aggregation
GF01	Enhanced numeric functions
GF02	Trigonometric functions
GF11	Advanced aggregate functions: binary set functions
GF13	SIZE function
GG02	Graph with a closed graph type
GG20	Explicit element type names
GP04	Named procedure calls
GQ03	Composite query: UNION
GQ08	Filter statement

GQ09	LET statement
GQ10	FOR statement: list value support
GQ12	ORDER BY and page statement: OFFSET clause
GQ13	ORDER BY and page statement: LIMIT clause
GQ20	Advanced linear composition with NEXT
GV45	Record types
GV47	Open record type
GV48	Nested record types
GV50	List value types
GV55	Path value types

7.1.19 Updating Specified Properties of Vertices and Edges by Importing a File (2.2.13)

Function

This API is used to update specified properties of vertices and edges by importing a file.

 NOTE

To prevent failures in restoring the updated graph data during system restarting, do not delete the data stored on OBS when the graph is in use.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-properties

Table 7-334 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain a project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name.

Request Parameters

NOTE

The format of the CSV file for updating properties is as follows:

- Vertex file (including **label**): vertex ID,**label**,**property_1...property_n**
- Vertex file (excluding **label**): vertex ID,**property_1...property_n**
- Edge file: source vertex ID, target vertex ID, label, edge ID, **property_1...property_n**

Table 7-335 Request body parameters

Parameter	Mandatory	Type	Description
vertexsetPath	At least one of the parameters vertexsetPath and edgesetPath is mandatory.	String	Vertex file directory or name.
vertexsetFormat	No	String	Vertex dataset format. Currently, the options are csv and txt . The default value is csv .
vertexProperties	Mandatory if vertexsetPath exists	Object	Label of a vertex and list of properties to be updated in a vertex file, in JSONArray format. For details, see Table 7-336 .
edgesetPath	At least one of the parameters vertexsetPath and edgesetPath is mandatory.	String	Edge file directory or name.
edgesetFormat	No	String	Edge dataset format. Currently, the options are csv and txt . The default value is csv .
edgeProperties	Mandatory if edgesetPath is set.	Object	Label of an edge and list of properties to be updated in an edge file, in JSONArray format. For details, see Table 7-337 .
targetProperties	Mandatory if edgesetPath is set.	Object	Property information used to distinguish duplicate edges in the edge file, in JSONArray format. For details, see Table 7-338 .

Parameter	Mandatory	Type	Description
delimiter	No	Character	Field separator in a CSV file. The default value is comma (,). The default element separator in a field of the list/set type is semicolon (;).
trimQuote	No	Character	Field quote character in a CSV file. The default value is double quotation marks ("). They are used to enclose a field if the field contains separators or line breaks.
obsParameters	Yes	String	OBS authentication parameters. For details, see Table 7-339 .
vertexFileContainsLabel	No	Boolean	Whether the vertex file contains label information. This parameter is optional. The default value is true .

Table 7-336 vertexProperties parameter descriptions

Parameter	Mandatory	Type	Description
label	Yes	String	Label name.
properties	Yes	Object	Properties to be updated, in JSONArray format. The sequence of the properties must be the same as that in the vertex file.

Table 7-337 edgeProperties parameter descriptions

Parameter	Mandatory	Type	Description
label	Yes	String	Label name.
properties	Yes	Object	Properties to be updated, in JSONArray format. The sequence of the properties must be the same as that in the edge file.

Table 7-338 targetProperties parameter descriptions

Parameter	Mandatory	Type	Description
label	Yes	String	Label name.

Parameter	Mandatory	Type	Description
properties	Yes	Object	Edge ID properties, in JSONArray format. Currently, only one property is supported.

Table 7-339 obsParameters parameter descriptions

Parameter	Mandatory	Type	Description
accessKey	Yes	String	Access key ID.
secretKey	Yes	String	Secret access key.

Response Parameters

Table 7-340 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job. NOTE You can check the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	Integer	Job type. This parameter is left blank if the request fails.

Example Request

Import a file to update specified properties of vertices and edges. The vertex file directory is **datasets/movie/movie.csv**, and the vertex dataset format is CSV. The edge file directory is **datasets/movie/ranking_edge.csv**, and the edge dataset format is also CSV.

```
POST http://{Endpoint}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-properties
{
  "vertexsetPath": "datasets/movie/movie.csv",
  "vertexsetFormat": "csv",
  "vertexProperties": [
    {
      "label": "movie",
      "properties": [
        "genres"
      ]
    }
  ],
  "edgesetPath": "datasets/movie/ranking_edge.csv",
  "edgesetFormat": "csv",
  "edgeProperties": [
    {
      "label": "rate",
      "properties": [
        "Datetime"
      ]
    }
  ],
  "targetProperties": [
    {
      "label": "rate",
      "properties": [
        "Rating"
      ]
    }
  ],
  "delimiter": ",",
  "trimQuote": "\"",
  "obsParameters": {
    "accessKey": "XXXXXXX",
    "secretKey": "XXXXXXX"
  },
  "vertexFileContainLabel": true
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "72b268f7-fdb6-467d-a41f-b1a09f26e593000003836",
  "jobType": 3
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.

Return Value	Description
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.20 Deleting Vertices and Edges by Files (2.2.15)

Function

This API is used to delete vertices and edges by reading the files.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=delete-by-file

Table 7-341 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-342 Request body parameters

Parameter	Mandatory	Type	Description
vertexsetPath	Either vertexsetPath or edgesetPath is mandatory.	String	Vertex file directory or name

Parameter	Mandatory	Type	Description
vertexsetFormat	No	String	Vertex dataset format. Currently, the options are csv and txt . The CSV format is used by default.
edgesetPath	Either vertexsetPath or edgesetPath is mandatory.	String	Edge file directory or name
edgesetFormat	No	String	Edge dataset format. Currently, the options are csv and txt . The CSV format is used by default.
targetProperties	No	Object	Indicates property information used to distinguish duplicate edges in the edge file, in JSONArray format. For details, see Table 7-343 .
delimiter	No	Character	Field separator in a CSV file. The default value is comma (,). The default element separator in a field of the list/set type is semicolon (;).
trimQuote	No	Character	Field quote character in a CSV file. The default value is double quotation marks ("). They are used to enclose a field if the field contains separators or line breaks.
obsParameters	Yes	String	OBS authentication parameters. For details, see Table 7-259 .

Table 7-343 targetProperties parameter description

Parameter	Mandatory	Type	Description
label	Yes	String	Name of a label
properties	Yes	Object	Edge ID properties, in JSONArray format. Currently, only one property is supported.

Response Parameters

Table 7-344 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job You can view the job execution status and obtain the return result by querying the job ID. For details, see Querying Job Status on the Service Plane (1.0.0) .
jobType	Integer	Job type. This parameter is left blank if the request fails.

Example Request

Delete vertices and edges by reading files. The vertex file directory is **datasets/movie/movie.csv**, and the vertex data set format is CSV. The edge file directory is **datasets/movie/ranking_edge.csv**, and the edge data set format is CSV.

```
POST http://{Endpoint}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=delete-by-file
{
  "vertexsetPath": "datasets/movie/movie.csv",
  "vertexsetFormat": "csv",
  "edgesetPath": "datasets/movie/ranking_edge.csv",
  "edgesetFormat": "csv",
  "targetProperties": [
    {
      "label": "rate",
      "properties": [
        "Rating"
      ]
    }
  ],
  "delimiter": ",",
  "trimQuote": "\"",
  "obsParameters": {
    "accessKey": "XXXXXXX",
    "secretKey": "XXXXXXX"
  }
}
```


Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "72b268f7-fdb6-467d-a41f-b1a09f26e593000003836",
  "jobType": 3
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.1.21 O&M Monitoring APIs

7.1.21.1 Viewing Monitoring Metrics

Function

This API is used to view monitoring metrics, including node metrics and graph instance performance monitoring metrics.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/om/metrics?
real_time=&with_performance_metrics=

Table 7-345 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Table 7-346 Query parameters

Parameter	Mandatory	Type	Description
real_time	No	Boolean	Whether to query real-time monitoring metrics. The value can be true or false . The default value is false . false: Metrics for a graph instance within 2 minutes are queried. true: Real-time monitoring metrics are queried, and the query is responded to in 3 to 5 seconds.
with_performance_metrics	No	Boolean	Whether to query performance metrics. The graph instance performance metrics and metrics of each node are returned. The value can be true or false . The default value is true . If set to false , only node metrics are returned, and the response time is reduced by 1 to 2 seconds.

Request Parameters

None

Response Parameters

Table 7-347 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
project_id	String	Schema structure query result For details, see Table 5-117.
id	String	Graph ID
name	String	Graph name
timestamp	long	Current timestamp
node_metrics	Array of objects	Node metrics
performance_metrics	Object	Performance metrics

Table 7-348 node_metrics

Parameter	Type	Description
overview	String	Node overview
disk_details	Array of objects	Disk details of the node
network_details	Array of objects	Network details of the node

Table 7-349 overview parameter descriptions

Parameter	Type	Description
ges_instance_name	String	Node name
instance_id	String	Node ID
work_ip	String	Node IP address
role	String	Node role
cpu_usage	Double	CPU usage
cpu_usage_usr	Double	CPU usage in user mode
cpu_usage_sys	Double	CPU usage in kernel mode

Parameter	Type	Description
cpu_iowait	Double	CPU I/O wait rate
cpu_idle	Double	CPU idle rate
mem_total	Double	Total memory size, in GB
mem_usage	Double	Used memory size, in GB
mem_free	Double	Available memory size, in GB
mem_cached	Double	Memory cache size, in GB
mem_buffer	Double	Memory buffer size, in GB
disk_total	Double	Total disk space, in GB
disk_usage_avg	Double	Average disk usage
disk_used	Double	Used disk space, in GB
disk_available	Double	Total available disk space, in GB
disk_io_read	Double	Disk read rate of a node, in KB/s
disk_io_write	Double	Disk read rate of a node, in KB/s
disk_io	Double	Total read and write speed of a node's disk
swap_total	Double	Total swap disk space of a node
swap_free	Double	Remaining swap disk space of a node
network_io_rate	Double	Network I/O rate of a node, in KB/s
host_stat	String	Node status
host_name	String	Node name

Table 7-350 disk_details parameter descriptions

Parameter	Type	Description
disk_name	String	Disk name
disk_type	String	Disk type
total	Double	Total disk space
available	Double	Available disk space
used	Double	Used disk space
used_percentage	Double	Disk usage
svctm	Long	Disk I/O service time, in milliseconds

Parameter	Type	Description
await	Double	Disk I/O wait time, in milliseconds
util	Double	Disk I/O usage
write_rate	Double	Disk read rate
read_rate	Double	Disk write rate

Table 7-351 network_details parameter descriptions

Parameter	Type	Description
status	Integer	NIC status
interface_name	String	NIC name
packets_rcv	Long	Number of received packets
packets_send	Long	Number of sent packets
packets_drop	Long	Number of lost packets
send_rate	Double	Sending rate, in KB/s
rcv_rate	Double	Receiving rate, in KB/s

Table 7-352 performance_metrics parameter descriptions

Parameter	Type	Description
cpu_usage	Double	CPU usage
memory_usage	Double	Memory usage
disk_usage	Double	Average disk usage
disk_io_rate	Double	Disk I/O rate
network_io_rate	Double	Network I/O rate
swap_disk_usage	Double	Swap disk usage
tomcat_connections_usage	Double	Tomcat connection usage
qps	Long	Number of requests per second
vertex_number	Long	Number of vertices
vertex_capacity	Long	Vertex capacity
vertex_usage	Double	Vertex usage

Parameter	Type	Description
edge_number	Long	Number of edges
edge_capacity	Long	Edge capacity
edge_usage	Double	Edge usage
read_waiting_queue_length	Long	Length of the read waiting queue
read_running_queue_length	Long	Length of the read running queue
write_waiting_queue_length	Long	Length of the write waiting queue
write_running_queue_length	Long	Length of the write running queue
graph_name	String	Graph instance name
jvm_heap_usage	Double	Java heap memory usage

Example Request

GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/om/metrics?real_time=true

Example Response

Status code: 200

Example response for a successful request

```
{
  "project_id": "xxx",
  "id": "xxxxx",
  "name": "baiwan_demo",
  "timestamp": 1699506387592,
  "node_metrics": [
    {
      "overview": {
        "network_io_rate": 2.99,
        "role": "slave",
        "disk_io_write": 107.23,
        "mem_cached": 2.57,
        "cpu_usage_usr": 7.15,
        "cpu_usage_sys": 3.07,
        "disk_io_read": 3.03,
        "ges_instance_name": "baiwan_demo-ges-cn-cn-2-1",
        "disk_used": 5.48,
        "swap_total": 0,
        "mem_buffer": 192.5,
        "disk_available": 144.47,
        "cpu_iowait": 0.17,
        "cpu_idle": 89.61,
        "mem_total": 15.15,
        "instance_id": "xxxxxxxxx",
        "mem_usage": 7.22,
        "disk_total": 149.95,
```

```
"host_stat": 200,
"mem_free": 11.29,
"swap_free": 0,
"cpu_usage": 10.22,
"disk_usage_avg": 3.65,
"work_ip": "172.16.25.224",
"host_name": "baiwan_demo-ges-cn-cn-2-1"
},
"disk_details": [
{
"svctm": 0,
"total": 50,
"util": 0.61,
"write_rate": 96.48,
"disk_name": "vda",
"disk_type": "system",
"used_percentage": 0.09,
"available": 45.68,
"await": 18.16,
"read_rate": 2.99,
"used": 4.32
}
],
"network_details": [
{
"send_rate": 0.05,
"packets_drop": 0,
"packets_rcv": 1001419,
"packets_send": 342518,
"interface_name": "eth0",
"rcv_rate": 0.06,
"status": 1
}
]
},
"performance_metrics": {
"tomcat_connections_usage": 0,
"network_io_rate": 3.11,
"swap_disk_usage": 0,
"vertex_capacity": 1200000,
"memory_usage": 23.83,
"vertex_number": 1071803,
"jvm_heap_usage": 0.23,
"edge_capacity": 1200000,
"read_waiting_queue_length": 0,
"disk_io_rate": 0,
"qps": 0,
"write_running_queue_length": 0,
"write_waiting_queue_length": 0,
"disk_usage": 1.77,
"edge_number": 1200000,
"edge_usage": 100,
"cpu_usage": 9.23,
"read_running_queue_length": 0,
"vertex_usage": 89.32
}
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Error Codes

See [Error Codes for Service Plane APIs](#).

7.1.21.2 Viewing Real-Time Requests

Function

This API is used to view the real-time requests on the current primary node.

URI

GET

/ges/v1.0/{project_id}/graphs/{graph_name}/om/real-time-queries?summary=

Table 7-353 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Table 7-354 Query parameters

Parameter	Mandatory	Type	Description
summary	No	Boolean	Whether to query only the summary information about real-time requests. The default value is false . If set to true , only the summary information is returned.

Request Parameters

None

Response Parameters

Table 7-355 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
instance_name	String	Instance name
edge_capacity	Long	Edge capacity
edge_number	Long	Number of edges
edge_usage	Double	Edge usage
vertex_capacity	String	Vertex capacity
vertex_number	Long	Number of vertices
vertex_usage	Long	Vertex usage
read_waiting_queue_length	Long	Length of the read waiting queue
read_running_queue_length	Long	Length of the read running queue
write_waiting_queue_length	Long	Length of the write waiting queue
write_running_queue_length	Long	Length of the write running queue
current_queries	List<CurrentQuery>	Details of the current query list

Table 7-356 CurrentQuery parameter descriptions

Parameter	Type	Description
task_name	String	Task name
request_id	String	Request ID

Parameter	Type	Description
status	String	Request execution status
request	String	Request parameter
running_duration	Double	Request execution duration, in seconds
pending_duration	Double	Request blocking duration, in seconds
begin_time	String	Request start time
progress	Double	Request execution progress

Example Request

View real-time requests.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/om/real-time-queries
```

Example Response

Status code: 200

Example response for a successful request

```
{
  "instance_name": "ges_demo-ges-dn-2-1",
  "edge_capacity": 0,
  "edge_number": 0,
  "edge_usage": 0,
  "vertex_capacity": 0,
  "vertex_number": 0,
  "vertex_usage": 0,
  "read_waiting_queue_length": 2,
  "read_running_queue_length": 1,
  "write_waiting_queue_length": 0,
  "write_running_queue_length": 0,
  "current_queries": [
    {
      "task_name": "incremental_load_graph",
      "request": {
        "vertex_file_format": "csv",
        "parallel_edge": "allow",
        "graph_name": "ges_demo",
        "vertex_file_path": "wepbucket/movie1/sit_ges_data_uat/nodes/",
        "edge_file_path": "wepbucket/movie1/sit_ges_data_uat/edges/",
        "schema_file_path": "wepbucket/movie1/sit_ges_data_uat/sit_ges_metadata_v6.xml",
        "vidSerialize": true,
        "offline": false,
        "trim_quote": "\\\"",
        "ignore_label": true,
        "delimiter": ";",
        "edge_file_format": "csv",
        "parameters": {
          "secret_key": "xxxxxxx",
          "access_key": "xxxxxxx",
          "region": "cn-north-7"
        }
      }
    }
  ],
}
```

```
{
  "pending_duration": 9.91311,
  "request_id": "0c56e2d14369586da38d7fe3b81bb1bd",
  "status": "pending",
  "begin_time": "",
  "running_duration": 0,
  "progress": 0
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2 Database Edition

7.2.1 Specification Description

As a new graph database product, the database edition provides graph management, graph data addition, deletion, modification, query, and analysis capabilities, supports Cypher, and provides graph data persistence to flush data written by users to disks. In this way, multiple copies and hardware redundancy are used to achieve high availability and fast fault recovery. Currently, the database edition has completed storage and query of hundreds of billions and trillions of graphs.

The following sections describe the APIs supported by database graphs.

Data Types

Type	Description
char	Character
float	Float type (32-bit float)
double	Double type (64-bit float)
bool	Boolean type. Available values are 0/1 and true/false .
long	Long integer (value range: -2^{63} to $2^{63}-1$)
int	Integer (value range: -2^{31} to $2^{31}-1$)

Type	Description
date	Date. Currently, the following formats are supported: • YYYY-MM-DD HH:MM:SS • YYYY-MM-DD NOTE The value of <i>MM</i> or <i>DD</i> must consist of two digits. If the value contains only one digit, add 0 before it, for example, 05-01 .
string	Variable-length string
enum	Enumeration type. The maximum number is 65535. Each enumeration type is a string.

Restrictions on Composite Types

The value can be **single**, **list** (with elements repeatable), or **set** (with elements unrepeatable).

Data Import Restrictions

For details about how to import data, see [Importing Data](#).

There are some restrictions when you import data to a database edition graph:

- Importing data concurrently
Multiple data files can be imported at the same time. To accelerate the import, split a large file into multiple OBS files that each does not exceed 5 GB.
- Uploading import logs to OBS
Set the **logDir** parameter to save import logs where you can find the error cause.
- Importing edge data
If you import only edges of a database edition graph, DLI cannot automatically generate vertices for the graph. If only edges are imported, you cannot query vertices or access the graph from vertices. You are advised to import vertex data too.

7.2.2 Vertex Operation APIs

7.2.2.1 Querying Vertex Details

Function

This API is used to query the vertex information (such as the label and property) based on the vertex ID.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/detail?
vertexIds={vertex_ids}

Table 7-357 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
vertexIds	Yes	String	IDs of the vertices to be queried. You can specify only one vertex ID for a database edition graph.

Request Parameters

None

Response Parameters

Table 7-358 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	Object	Vertex details you want to query. For details, see Table 7-359 .
result	String	Query results. The value is success for a successful request and failed for a failed request.

Table 7-359 data parameter descriptions

Parameter	Type	Description
vertices	List	Vertex result set. If no edge is found, this parameter will be left blank.

Example Request

Query node information by node ID and return node details.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/detail?vertexIds=46
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "data": {
    "vertices": [
      {
        "id": "46",
        "labels": [
          "user"
        ],
        "properties": {
          "user": {
            "userid": [
              0
            ],
            "gender": [
              "F"
            ],
            "age": [
              "25-34"
            ],
            "occupation": [
              "artist"
            ],
            "Zip-code": [
              "98133"
            ]
          }
        }
      }
    ]
  },
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
```

```
"errorMessage": "Graph [31dc8f201e6246768fde20492d4a9875-demo] does not exist, please check  
projectId and graphName.",  
"errorCode": "GES.8000",  
"result": "failed"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.2.2 Batch Querying Vertices

Function

This API is used to query the vertex data (such as the labels and properties) in batches based on the vertex IDs.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?
action_id=batch-query

Table 7-360 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-361 Request body parameter

Parameter	Mandatory	Type	Description
vertices	Yes	String	IDs of the vertices to query

Response Parameters

Table 7-362 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
data	Object	This parameter is returned when the query is successful, which contains the query results of vertices
result	String	Query results. The value is success for a successful request and failed for a failed request.

Example Request

Retrieve information about a group of nodes using their IDs.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-query
{
  "vertices":
    ["0","1"]
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "data": {
    "vertices": [
      {
        "id": "0",
        "labels": [
          "movie"
        ],
        "properties": {
          "movie": {
            "movieid": [
              0
            ],
            "title": [
              "American Beauty (1999)"
            ],
            "genres": [
              "Comedy|Drama"
            ]
          }
        }
      },
      {
        "id": "51",
        "labels": [
          "user"
        ],
        "properties": {
          "user": {
            "userid": [
              5
            ],
            "gender": [
              "F"
            ],
            "age": [
              "56+"
            ],
            "occupation": [
              "homemaker"
            ],
            "Zip-code": [
              "46911"
            ]
          }
        }
      }
    ]
  },
  "result": "success"
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Bad Request, parameter vertices cannot be null",
  "errorCode": "GES.8214",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.2.3 Batch Adding Vertices

Function

This API is used to add vertices in batches.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?
action_id=batch-add

Table 7-363 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-364 Request body parameters

Parameter	Mandatory	Type	Description
vertices	Yes	List	Vertices to add. You can add a maximum of 10,000 vertices at a time. For details about this array, see the vertices parameters .
overrideExists	No	Boolean	Checks whether the vertex to be inserted exists. The default value is false . - If this parameter is set to false, an error is reported as long as there is one vertex, and all vertices fail to be written. - If this parameter is set to true, the existing vertices are overwritten.

Table 7-365 vertices parameter description

Parameter	Mandatory	Type	Description
vertex	Yes	String	Vertex ID
label	Yes	String	Vertex label
properties	No	Json	Value of each property

Response Parameters

Table 7-366 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.

Parameter	Type	Description
result	String	Request result. If the request is successful, the value is success . If the request fails, the value is failed .

Example Request

Add vertices in batches. The names of the vertices to add are **150** and **6**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-add
{
  "vertices": [
    {
      "vertex": "150",
      "label": "movie",
      "properties": {
        "movieid": [
          "150"
        ],
        "title": [
          "testmoive"
        ],
        "genres": [
          "Comedy"
        ]
      }
    },
    {
      "vertex": "6",
      "label": "movie",
      "properties": {
        "movieid": [
          "6"
        ],
        "title": [
          "testmoive_exist_id"
        ],
        "genres": [
          "Comedy"
        ]
      }
    }
  ],
  "overrideExists": true
}
```

NOTE

- **SERVER_URL**: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).
- In the example, if vertex **6** already exists in the graph, properties of vertex **6** are overwritten.

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
```

```
"result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "vertex [6, movie] already exist",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.2.4 Batch Deleting Vertices

Function

This API is used to delete vertices in batches based on the vertex IDs.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?
action_id=batch-delete

Table 7-367 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-368 Request body parameter descriptions

Parameter	Mandatory	Type	Description
vertices	Yes	List	IDs of the vertices to delete

Response Parameters

Table 7-369 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
result	String	Response result. If the request is successful, the value is success . If the request fails, the value is failed .

Example Request

Delete vertices in batches by vertex ID. The IDs of the vertices to delete are **Vivian** and **46**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-delete
{
  "vertices": [
    "Vivian",
    "46"
  ]
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Bad Request, parameter vertices cannot be null",
  "errorCode": "GES.8214",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.2.5 Batch Updating Vertex Properties

Function

This API is used to update vertex properties in batches.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/properties/action?
action_id={actionId}

Table 7-370 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
action_id	Yes	String	Operator. The options are: • batch-update: Update the value of a property. • batch-add: Add the value to a property. When the property's cardinality is single, the operation is the same as that of batch-update. When cardinality is list or set, the operator adds a value to a set. • batch-del: Delete a property value.

Request Parameters

Table 7-371 Request body parameter description

Parameter	Mandatory	Type	Description
vertices	Yes	List	Vertices to update. For details about this array, see the Table 7-372 .

Table 7-372 vertices parameter descriptions

Parameter	Mandatory	Type	Description
vertex	Yes	String	Vertex ID
label	No	String	Vertex label
properties	Yes	Json	Value of each property to be updated

Response Parameters

Table 7-373 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
result	String	Response result. If the request is successful, the value is success . If the request fails, the value is failed .

Example Request

Update vertex properties in batches. The names of the vertices to update are **150** and **6**.

POST http://{{SERVER_URL}}/ges/v1.0/{{project_id}}/graphs/{{graph_name}}/vertices/properties/action?action_id=batch-update

```
{
  "vertices": [
    {
      "vertex": "150",
      "label": "movie",
      "properties": {
        "movieid": [
          "150"
        ],
        "title": [
          "testmoive"
        ],
        "genres": [
          "Comedy"
        ]
      }
    },
    {
      "vertex": "6",
      "properties": {
        "title": [
          "testmoive_exist_id"
        ],
        "genres": [
          "Comedy"
        ]
      }
    }
  ]
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Vertex [150, movie] not exist.",
  "errorCode": "GES.8220",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.3 Edge Operation APIs

7.2.3.1 Querying Edge Details

Function

This API is used to query the detailed information about an edge based on the source vertex, target vertex, and index of the edge. Information about edges and properties is returned.

URI

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?  
source={sourceVertex}&target={targetVertex}&label={label}&sortKey={sortKey}&sortKeyType={sortKeyType}
```

Table 7-374 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge
label	No	String	Label of an edge
sortKey	No	String	Sort key value of duplicate edges.
sortKeyType	No	String	Sort key type of duplicate edges. The options are Integer , String , and varString . This parameter is mandatory when sortKey is set. It must be the same as that of sortKey set during graph creation, except when null is assigned.

Request Parameters

None

Response Parameters

Table 7-375 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.

Parameter	Type	Description
data	Object	Query results. If the query is successful, the query result will be returned. If the query fails, this parameter will be left blank.
result	String	Query result. If the query is successful, the value is success . If the query fails, the value is failed .

Table 7-376 data parameter descriptions

Parameter	Mandatory	Type	Description
edges	Yes	List	Edge result set. If no edge is found, this parameter will be left blank.

Example Request

Query details about an edge.

```
GET /ges/v1.0/{project_id}/graphs/{graph_name}/edges/detail?
source=46&&target=39&&label=rate&&sortKey=5&&sortKeyType=int
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
{
  "data": {
    "edges": [
      {
        "source": "46",
        "target": "39",
        "label": "rate",
        "sortKey": 5,
        "properties": {
          "Rating": [
            5
          ],
          "Datetime": [
            "2018-01-0120:30:05"
          ]
        }
      }
    ]
  },
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Graph [31dc8f201e6246768fde20492d4a9875-demo] does not exist, please check  
projectId and graphName.",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.3.2 Batch Querying Edges

Function

This API is used to query the detailed information about edges in batches based on the source vertices, target vertices, and indexes of the edges. Information about edges and properties is returned.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-query

Table 7-377 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-378 Request body parameter description

Parameter	Mandatory	Type	Description
edges	Yes	List	Edge array to be queried

Table 7-379 edges parameter descriptions

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge
label	No	String	Edge label
sortKey	No	String	Sort key of duplicate edges. The options are Integer , String , and varString . If this parameter is set, label must also be set. The type must be the same as that of sortKey set during graph creation, except when null is assigned.

Response Parameters

Table 7-380 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	String	The data field is contained when the query is successful, and the data field contains the edges query result.

Parameter	Type	Description
result	String	Query result. If the query is successful, the value is success . If the query fails, the value is failed .

Example Request

Query details about edges in batches based on the source vertex, target vertex, and index. The source vertex of the edges to be queried is **Vivian**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-query
{
  "edges": [
    {
      "source": "Vivian",
      "target": "Lethal Weapon",
      "label": "rate"
    },
    {
      "source": "Vivian",
      "target": "Raising Arizona"
    }
  ]
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
{
  "data": {
    "edges": [
      {
        "source": "Vivian",
        "target": "Raising Arizona",
        "label": "rate",
        "sortKeyType": "abc",
        "properties": {
          "Score": [
            4
          ],
          "Datetime": [
            "2000-12-27 23:51:42"
          ]
        }
      },
      {
        "source": "Vivian",
        "target": "Lethal Weapon",
        "label": "rate",
        "sortKeyType": "bbb",
        "properties": {
          "Score": [
            5
          ],
          "Datetime": [

```

```
    "2000-12-27 23:44:41"
  }
}
},
"result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "parameter [source] does not exist",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.3.3 Batch Adding Edges

Function

This API is used to add edges in batches.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add

Table 7-381 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-382 Request body parameter descriptions

Parameter	Mandatory	Type	Description
edges	Yes	edges List	Edge array to be added
parallelEdge	No	parallelEdge Object	How to process repetitive edges
createNotExists	No	Boolean	Whether to add source or target vertices that do not exist in the edges parameter before adding edges. The value can only be false . false : Edges can be added regardless of whether the source or target vertex exists.

Table 7-383 edges parameter descriptions

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge
label	Yes	String	Edge label
sortKey	No	String	Sort key of duplicate edges. The options are Integer , String , and varString . If this parameter is set, label must also be set. The type must be the same as that of sortKey set during graph creation, except when null is assigned. If the sort key is empty, the value is null .

Parameter	Mandatory	Type	Description
properties	No	Object	Value of each property

Table 7-384 parallelEdge parameter descriptions

Parameter	Mandatory	Type	Description
action	No	String	Processing mode. The value can be override or ignore . - If the value is override, the previous repetitive edges are overwritten. - If the value is ignore and an edge already exists, the previous edge will not be overwritten. If no edge exists, the system adds an edge.
ignoreLabel	No	Boolean	Whether to ignore labels on repetitive edges. The value can only be false . false : Edges with the same source vertex, target vertex, label, and sort key are duplicate edges.

Response Parameters

Table 7-385 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the request is successful, the value is success . If the request fails, the value is failed .

Example Request

Add edges in batches. The source vertex is **46**, the target vertices are **39** and **38**, the edge label is **rate**, and **sortKey** is **3** or **4**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add
{
  "edges": [
    {
      "source": "46",
      "target": "39",
      "label": "rate",
      "sortKey": 3,
      "properties": {
        "Rating": [
          5
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    },
    {
      "source": "46",
      "target": "38",
      "label": "rate",
      "sortKey": 4,
      "properties": {
        "Rating": [
          4
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    }
  ],
  "parallelEdge": {
    "action": "override",
    "ignoreLabel": false
  },
  "createNotExists": false
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success",
  "data": {
    "edges": [
      {
        "index": "7",
        "source": "46",
        "target": "39"
      },
      {
        "index": "0",
```

```
    "source": "46",  
    "target": "38"  
  }  
]  
}  
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400  
{  
  "errorMessage": "Parameter [source] cannot be empty.",  
  "errorCode": "GES.8000",  
  "result": "failed"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.3.4 Batch Deleting Edges

Function

This API is used to delete edges in batches based on the source vertices, target vertices, and indexes of the edges.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-delete

Table 7-386 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-387 Request body parameter descriptions

Parameter	Mandatory	Type	Description
edges	Yes	List	Edge array to be deleted
executionMode	No	String	sync indicates the synchronous mode, and async indicates the asynchronous mode. The default value is sync .
ignoreError	No	Boolean	Whether to ignore errors, for example, the edge to delete does not exist. The default value is false , indicating that errors will not be ignored. When set to true , errors in JSON format are not ignored.

Table 7-388 edges parameter descriptions

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge
label	No	String	Label of an edge. If the index parameter is set, this parameter is ignored. If the index parameter is not set, an edge that meets the source , target , and label conditions is deleted. If the specified label value does not exist in the schema or the edge with the same label does not exist, no edge will be deleted.

Parameter	Mandatory	Type	Description
sortKey	No	String	Sort key of duplicate edges. The options are Integer , String , and varString . If this parameter is set, label must also be set. The type must be the same as that of sortKey set during graph creation, except when null is assigned.

Response Parameters

- Synchronous call

Table 7-389 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
result	String	Response result. If the request is successful, the value is success . If the request fails, the value is failed .

- Asynchronous call (2.2.14)

Table 7-390 Response body parameter descriptions

Parameter	Mandatory	Type	Description
errorMessage	No	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Parameter	Mandatory	Type	Description
errorCode	No	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	No	String	ID of the deletion job. This parameter is left blank when the request fails. This ID can be used as a request parameter to obtain the deletion result through the API for querying the job status.
jobType	No	Integer	Job type. This parameter is left blank when the request fails.

Example Request

Delete edges in batches. The source vertices of the edges are **39631050_Landscape** and **27803870_Landmark building**, and the target vertices of the edges are **27803870_Landmark building** and **27661363_Villa hot spring**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-delete
{
  "edges": [
    {
      "source": "39631050_Landscape",
      "target": "27803870_Landmark building"
    },
    {
      "label": "road",
      "source": "27803870_Landmark building",
      "target": "27661363_Villa hot spring"
    }
  ],
  "ignoreError": true
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

- Synchronization mode

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "parameter does not contain source",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

- Asynchronous mode

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "500dea8f-9651-41fe-8299-c20f13a032ea",
  "jobType": 3
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.3.5 Batch Updating Edge Properties

Function

This API is used to update edge properties in batches.

URI

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?
action_id={actionId}
```


Table 7-391 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
action_id	Yes	String	Operator. The options are: • batch-update: Update the value of a property. • batch-add: Add the value to a property. When the property's cardinality is single, the operation is the same as that of batch-update. When cardinality is list or set, the operator adds a value to a set. • batch-del: Delete a property value.

Request Parameters

Table 7-392 Request body parameter descriptions

Parameter	Mandatory	Type	Description
edges	Yes	List	Edge array to be updated

Table 7-393 edges parameter descriptions

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge
label	No	String	Edge label
sortKey	No	String	Sort key of duplicate edges. The options are Integer , String , and varString . If this parameter is set, label must also be set. The type must be the same as that of sortKey set during graph creation, except when null is assigned.
properties	Yes	Object	Value of each property

Response Parameters

Table 7-394 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
result	String	Response result. If the request is successful, the value is success . If the request fails, the value is failed .

Example Request

Update edge properties in batches. The source vertex of the edge is **46**, and the target vertices of the edge are **39** and **38**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/properties/action?
action_id=batch-update
{
  "edges": [
    {
      "source": "46",
      "target": "39",
      "properties": {
        "Rating": [
          5
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    },
    {
      "source": "46",
      "target": "38",
      "index": "0",
      "properties": {
        "Rating": [
          4
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    }
  ]
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Edge [46] not exist.",
  "errorCode": "GES.8221",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.4 Metadata Operation APIs

7.2.4.1 Adding a Label

Function

This API is used to add a label.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels

Table 7-395 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

- Request parameter descriptions

Table 7-396 Request body parameter descriptions

Parameter	Mandatory	Type	Description
name	Yes	String	Name of a label A label name can contain a maximum of 256 characters. Only letters, numbers, spaces, and special characters (%@#\$.?*.+_) are allowed.
type	No	String	Label type, indicating that the label is used for vertices or edges. The options are: vertex: The label is used for vertices. edge: The label is used for edges. all: The label is used for vertices and edges. The default value is all .
properties	Yes	Object	Properties you want to add to the label. For details about the parameters, see Table 7-397 .

Table 7-397 properties parameter descriptions

Parameter	Mandatory	Type	Description
property	No	Object	Label properties. For details about the parameters, see Table 7-398 .

Table 7-398 property parameter descriptions

Parameter	Mandatory	Type	Description
name	Yes	String	Property name 1. A property name can contain a maximum of 256 characters. 2. A property name cannot contain <, >, &, ASCII 14, 15, or 30. 3. The property under a label must be unique.
cardinality	Yes	String	Composite type of a property. Currently, only single is supported.
dataType	Yes	String	Data type of a property. For details, see the metadata types in Specification Description .

Response Parameters

Table 7-399 Parameter description

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the request is successful, the value is success . If the request fails, the value is failed .
cause	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.

Example Request

Add a label. The label name is **book**. The label has one property to add.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/labels
{
  "name": "book",
  "type": "vertex",
  "properties": [
    {
      "property": {
        "name": "Title",
        "cardinality": "single",
        "dataType": "string"
      }
    },
    {
      "property": {
        "name": "Version",
        "cardinality": "single",
        "dataType": "string"
      }
    }
  ]
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "label : book has exist",
  "errorCode": "GES.8701",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.4.2 Updating a Label

Function

This API is used to modify a label in overwriting mode. Modifying a label affects only metadata and does not affect existing vertices and edges. The existing vertex and edge properties remain in the old label format, while newly added (or overwritten) vertex and edge properties are written in the latest label format.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema?label={labelName}

Table 7-400 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
label	Yes	String	Label name

Request Parameters

Table 7-401 Request body parameter descriptions

Parameter	Mandatory	Type	Description
type	No	String	Label type, indicating that the label is used for vertices or edges. The options are: • vertex: The label is used for vertices. • edge: The label is used for edges. • all: The label is used for vertices and edges. The default value is all .
properties	Yes	Object	Property array to be updated. Table 7-402 describes the parameters in an array.

Table 7-402 properties parameter descriptions

Parameter	Mandatory	Type	Description
property	No	Object	Label properties. For details about the parameters, see Table 7-403. Note: If modifying the label disrupts the order of the original properties, such as inserting a new property in the middle of the original property list or deleting a property, then after data export, the old vertices and edges under this label will still be exported in the format of the old label, which does not match the exported schema and cannot be re-imported. If adding a property at the end of the original label, all exported data can be imported without being affected.

Table 7-403 property parameter descriptions

Parameter	Mandatory	Type	Description
name	Yes	String	Property name 1. A property name can contain a maximum of 256 characters. 2. A property name cannot contain <, >, &, ASCII 14,15 and 30. 3. The property under a label must be unique.
cardinality	Yes	String	Composite type of a property. Currently, only single is supported.
dataType	Yes	String	Data type of a property. For details, see the metadata types in Specification Description .

Response Parameters

Table 7-404 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Response result. If the request is successful, the value is success . If the request fails, the value is failed .

Example Request

Update the label named **book**.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema?label={labelName}
{
  "type": "vertex",
  "properties": [
    {
      "property": {
        "name": "Title",
        "cardinality": "single",
        "dataType": "string"
      }
    },
    {
      "property": {
        "name": "Version",
        "cardinality": "single",
        "dataType": "string"
      }
    }
  ]
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "label : book not exist",
  "errorCode": "GES.8703",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.4.3 Querying Labels

Function

This API is used to query labels.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema?label={labelName}

Table 7-405 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Parameter	Mandatory	Type	Description
graph_name	Yes	String	Graph name
label	Yes	String	Label name

Response Parameters

Table 7-406 Response parameters

Parameter	Type	Description
data	data Object	Query result. This parameter is left blank when the request fails.
result	String	Response result. The value is success for a successful request and failed for a failed request.
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.

Table 7-407 data

Parameter	Type	Description
properties	Object	Property array.
type	String	Label type, indicating that the label is used for vertices or edges.

Example Request

Query labels.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema?label={labelName}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "data": {
    "type": "vertex",
    "properties": [
      {
        "name": "Rating",
        "type": "int",
        "cardinality": "single"
      },
      {
        "name": "Datetime",
        "type": "string",
        "cardinality": "single"
      }
    ]
  },
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Label [book1] does not exist.",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.4.4 Querying Graph Metadata Details

Function

This API is used to query graph metadata details.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema

Table 7-408 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Response Parameters

Table 7-409 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	Object	Query results. This parameter is left blank when the request fails.

Table 7-410 data parameter description

Parameter	Type	Description
schema	List	Definition of each label and their associated property fields

Table 7-411 schema parameter descriptions

Parameter	Type	Description
label	String	label name
properties	Object	Property array. For details, see Table 7-412 .
type	String	Label type, indicating that the label is used for vertices or edges.

Table 7-412 properties parameter descriptions

Parameter	Type	Description
name	String	Property name 1. A property name can contain a maximum of 256 characters. 2. A property name cannot contain <, >, &, ASCII 14, 15, or 30. 3. The property under a label must be unique.
cardinality	String	Cardinality type of a property. The options are: • single • list • set
dataType	String	Data type of a property. For details, see the metadata types in Specification Description .

Example Request

Query metadata details of a graph.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "data": {
    "schema": [
      {
        "label": "__DEFAULT__",
```

```
    "type": "all"
  },
  {
    "label": "friends",
    "type": "vertex"
  },
  {
    "label": "movie",
    "type": "vertex",
    "properties": [
      {
        "name": "ChineseTitle",
        "type": "string",
        "cardinality": "single"
      },
      {
        "name": "Year",
        "type": "int",
        "cardinality": "single"
      }
    ]
  },
  {
    "label": "user",
    "type": "vertex",
    "properties": [
      {
        "name": "Name",
        "type": "string",
        "cardinality": "single"
      },
      {
        "name": "Occupation",
        "type": "string",
        "cardinality": "single"
      },
      {
        "name": "Zip-code",
        "type": "string",
        "cardinality": "single"
      }
    ]
  },
  {
    "label": "rate",
    "type": "edge",
    "properties": [
      {
        "name": "Score",
        "type": "int",
        "cardinality": "single"
      },
      {
        "name": "Datetime",
        "type": "date",
        "cardinality": "single"
      }
    ]
  }
]
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
```

```
"errorMessage": "Graph [854d9316e71d4dc599c5f27ea323d445-ges_fdb1] does not exist, please check  
projectId and graphName.",  
"errorCode": "GES.8000",  
"result": "failed"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.4.5 Generating Data Assets

Function

This API is used to generate data assets.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets

Table 7-413 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-414 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the results by querying the job ID. For details, see Job Management APIs .

Example Request

Generate data assets.

```
POST /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets
{ }
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

7.2.4.6 Obtaining Data Assets

Function

This API is used to obtain data assets.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets

Table 7-415 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-416 Parameter descriptions

Parameter	Type	Description
generating	Bool	Whether data assets are being generated
progress	String	Progress of generating data assets
last_generate_time	String	Last time when data assets were generated
data_assets	Object	Number of vertices and edges under different labels. For details, see Table 7-417 .
errorCode	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.

Table 7-417 data_assets parameter description

Parameter	Type	Description
vertex	Object	Number of vertices under different labels
edge	Object	Number of edges under different labels

Example Request

Obtain data assets.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/schema/data-assets
```

Example Response

Status code: 200

There are several scenarios for example response for successful requests, including:

- A data asset has never been generated.
Http Status Code: 200

```
{  
  "result": "success"  
}
```
- A data asset has never been generated and is being generated.
Http Status Code: 200

```
{  
  "progress": "10.05%",  
  "generating": true  
}
```
- A data asset has been generated and a latest one is being generated.
Http Status Code: 200

```
{  
  "last_generate_time": "2022-1-3 12:34:12",  
  "data_assets": {  
    "vertex": {  
      "label1": 3,  
      "label2": 14  
    },  
    "edge": {  
      "label3": 123,  
      "label4": 435  
    }  
  },  
  "progress": "10.05%",  
  "generating": true  
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.5 Index Operation APIs**7.2.5.1 Creating an Index****Function**

This API is used to create indexes based on the specified information such as `indexName` and `IndexType`. Currently, composite indexes and full-text indexes are supported.

- Composite indexes include **GlobalCompositeVertexIndex**, **GlobalCompositeEdgeIndex**, **CompositeVertexIndex**, and **CompositeEdgeIndex**. A composite index is created for a fixed combination of properties. A local index can be created on a specified label. You do not need to specify a label for a global composite index. As long as a label contains a specified property, an index is automatically created on the label. Indexes can be used to accelerate queries.
- Full-text indexes (`FullTextIndex`) can implement functions such as full-text search and fuzzy search. To ensure you get the most up-to-date data, you are advised to wait for at least 60 seconds after making an update before searching. This is because of the refresh mechanism of full-text indexing. For details about how to use full-text indexes, see [Querying Vertices That Meet Filter Criteria](#) and [Querying Edges That Meet Filter Criteria](#). You can also

use full-text indexes in Cypher statements. For details, see [Performing Cypher Queries](#).

 NOTE

- Full-text indexes are currently supported only by the database edition.
- After an index is created, wait for 30 seconds for index synchronization. After the synchronization is complete, Cypher queries can be accelerated using the index.

Indexes

Index	Fuzzy Search	Speed	Flexibility
Composite index	Not supported	Fast	Fixed composite property keys only
Full-text index	Supported	Slower than composite indexes	Randomly combined property keys

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/indices

Table 7-418 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-419 Request body parameter descriptions

Parameter	Mandatory	Type	Description
indexName	Yes	String	Index name. The name can contain a maximum of 63 characters. Only letters, numbers, hyphens (-), and underscores (_) are allowed. NOTE

Parameter	Mandatory	Type	Description
indexType	Yes	String	Index type, which is case sensitive • GlobalCompositeVertexIndex is a global composite vertex index. • GlobalCompositeEdgeIndex is a global composite edge index. • CompositeVertexIndex is a local composite vertex index. • CompositeEdgeIndex is a local composite edge index. • FullTextIndex is a full-text index.
hasLabel	No	Boolean	Whether labels exist. The default value is false . • true • false
indexLabel	No	List	Labels on which indexes are created. This parameter is available only when indexType is set to CompositeVertexIndex or CompositeEdgeIndex . This parameter is mandatory.
indexProperty	No (If hasLabel is false or null , this parameter is mandatory.)	List	Index property list Indexes can be created for the following property types: integer , float , double , long , enum , string , and date . If the index type is composite, the property type can only be single .
strLength	No	Integer	Properties of the string type are truncated to a fixed length during index encoding. This parameter indicates the truncation length. If not specified, the default value is 40 bytes. The value range is [1, 256]. Typically, you do not need to manually specify this parameter. Even if some properties exceed this length and are truncated during index encoding. It usually does not cause significant performance degradation. Unless most properties exceed the default length, you are advised to consider increasing strLength , but this will lead to a decrease in indexing performance.

 NOTE

- If a property is of the string type, the value must be no more than 40 bytes. The excess part will be deleted.
- Cypher queries can be accelerated with indexes whose **hasLabel** is **True**.
 - If **indexProperty** is left blank, the created index is a label index, which accelerates label filtering.
 - If **indexProperty** is specified, the created index is a property index, which accelerates property filtering.

Response Parameters

Table 7-420 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .
jobType	String	Type of an asynchronous job
result	String	If the execution is successful, the value of result is success .

Example Request

Create a composite index. The index name is **ageIndex** and the index type is global vertex index.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
{
  "indexName": "ageIndex",
  "indexType": "GlobalCompositeVertexIndex",
  "hasLabel": true,
  "indexProperty": ["age"]
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232",
  "jobType": 8
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Graph [854d9316e71d4dc599c5f27ea323d445-ges_fdb1] does not exist, please check  
projectId and graphName.",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.5.2 Deleting an Index

Function

This API is used to delete an index based on the specified indexName.

URI

DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/indices/{indexName}

Table 7-421 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
indexName	Yes	String	Index name

Response Parameters

Table 7-422 Parameter description

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobType	String	Type of an asynchronous job

Example Request

Delete an index by name. The index name is **ageIndex**.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices/ageIndex
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
```

```
"jobId": "fb74314e-a82d-41b2-8900-96e2559fa0d9000168232",  
"jobType": 9  
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400  
{  
  "errorMessage": "Graph [854d9316e71d4dc599c5f27ea323d445-ges_fdb1] does not exist, please check  
projectId and graphName.",  
  "errorCode": "GES.8000",  
  "result": "failed"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.5.3 Querying Indexes

Function

This API is used to query all indexes created on a graph.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/indices

Table 7-423 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Response Parameters

Table 7-424 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
data	Object	Index data you want to query
result	String	Query results. If the query is successful, success is displayed.
indices	List	Indexes of the query results
indexType	String	Index types of the query results
indexName	String	Index names of the query results
indexLabel	List	Labels of local indexes
indexProperty	List	Index properties of the query results
hasLabel	Boolean	Whether the indexes of the query results contain labels NOTE For full-text indexes, the default value is false .
strLength	Integer	Query result, which is the truncated length of properties of the string type during encoding and is not displayed if not specified

Example Request

Query all indexes created on a graph.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "data": {
    "indices": [
      {
        "indexType": "GlobalCompositeVertexIndex",
        "indexName": "ageIndx",
        "indexProperty": [
          "age"
        ],
        "hasLabel": true
      }
    ]
  },
  "result": "success"
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Graph [854d9316e71d4dc599c5f27ea323d445-ges_fdb1] does not exist, please check  
projectId and graphName.",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.5.4 Batch Creating Indexes

Function

This API is used to create indexes in batches. By doing so, the number of data scans is reduced and the overall time required is shorten. The types of indexes that

can be created are the same as those of the index creation API. For details, see [Creating an Index](#).

NOTE

After an index is created, wait for 30 seconds for index synchronization. After the synchronization is complete, Cypher queries can be accelerated using the index.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/indices/action?action_id=batch-build

Table 7-425 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-426 Request body parameter descriptions

Parameter	Mandatory	Type	Description
indices	Yes	Array	Index array. A single graph currently supports a maximum of 500 indexes. For details about index parameters, see Table 7-427 .

Table 7-427 indices parameter descriptions

Parameter	Mandatory	Type	Description
indexName	Yes	String	Index name. The name can contain a maximum of 63 characters. Only letters, numbers, hyphens (-), and underscores (_) are allowed.

Parameter	Mandatory	Type	Description
indexType	Yes	String	Index type, which is case sensitive • GlobalCompositeVertexIndex is a global composite vertex index. • GlobalCompositeEdgeIndex is a global composite edge index. • CompositeVertexIndex is a local composite vertex index. • CompositeEdgeIndex is a local composite edge index. • FullTextIndex is a full-text index.
hasLabel	No	Boolean	Whether there are labels. The default value is false . • true • false
indexLabel	No	List	Label list. This parameter is available and mandatory only when indexType is set to CompositeVertexIndex or CompositeEdgeIndex . It is used to specify the labels on which indexes are created.
indexProperty	No (If hasLabel is false or null , this parameter is mandatory.)	String	Index property list Indexes can be created for the following property types: integer , float , double , long , enum , string , and date .
strLength	No	Integer	Properties of the string type are truncated to a fixed length during index encoding. This parameter indicates the truncation length. If not specified, the default value is 40 bytes. The value range is [1, 256]. Typically, you do not need to manually specify this parameter. Even if some properties exceed this length and are truncated during index encoding. It usually does not cause significant performance degradation. Unless most properties exceed the default length, you are advised to consider increasing strLength , but this will lead to a decrease in indexing performance.

 NOTE

- If a property is of type string or char array, the value must not exceed 40 bytes. Any extra characters will be removed.
- Cypher queries can be accelerated with indexes whose **hasLabel** is **True**.
 - If **indexProperty** is left blank, the created index is a label index, which accelerates label filtering.
 - If **indexProperty** is specified, the created index is a property index, which accelerates property filtering.

Response Parameters

Table 7-428 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job NOTE • You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs.

Example Request

Create multiple composite indexes at a time. The index names are **vertexIndex**, **edgeIndex**, and **useridIndex**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/indices/action?action_id=batch-build
{
  "indices": [
    {
      "indexName": "vertexIndex",
      "indexType": "GlobalCompositeVertexIndex",
      "hasLabel": true,
      "indexProperty": []
    },
    {
      "indexName": "edgeIndex",
      "indexType": "GlobalCompositeEdgeIndex",
      "hasLabel": true,
      "indexProperty": []
    }
  ]
}
```

```
"indexName": "useridIndex",  
"indexType": "GlobalCompositeEdgeIndex",  
"hasLabel": true,  
"indexProperty": ["userid"]  
}  
]  
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200  
{  
  "jobId": "ccffb1141c909708a5a700d763dc93450000314299090"  
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400  
{  
  "errorMessage": "Parameter error!",  
  "errorCode": "GES.8005"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.6 HyG Graph Management APIs

7.2.6.1 Creating a HyG Graph

Function

This API is used to create a HyG graph.

NOTE

- Computing of graphs of the database edition relies on the HyG engine. Before executing an algorithm, you need to create a HyG graph and synchronize data from the graph database to the HyG engine.
- To apply for access to the HyG component, [submit a service ticket](#) as it is currently only available through a whitelist.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}

Table 7-429 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name. The graph name must exist in the graph database, and currently only graphs with idType of fixedLengthString are supported.

Request Parameters

Table 7-430 Request body parameters

Parameter	Mandatory	Type	Description
policy	No	String	Graph partitioning policy. Currently, the oec policy is supported. The default policy is oec .

Parameter	Mandatory	Type	Description
inEdge	No	Boolean	Whether the graph contains incoming edges. The default value is false. If this parameter is set to true, the data synchronization performance will be affected. For some algorithms, such as <code>shortest_path</code>, <code>sssp</code>, and <code>k_hop</code>, if the incoming edges are not contained, the algorithm performance may deteriorate or an error may be reported. For details, see the parameter description of the corresponding algorithm.

Response Parameters

Table 7-431 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.
result	String	Execution results • If the execution is successful, the value is success. • If the execution fails, the value is failed.

Example Request

To create a HyG graph, set the partitioning policy to **oec** and configure the graph to contain incoming edges.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}
{
  "policy": "oec",
  "inEdge": true
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Graph [cb1ed5491f7f466e8947ff09c06ca08c-ges_hyg1] does not exist, please check  
projectId and graphName.",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.6.2 Synchronizing HyG Graph Data

Function

This API is used to synchronize graph database updates to the HyG computing engine.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/sync

Table 7-432 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-433 Request body parameters

Parameter	Mandatory	Type	Description
vertex	No	JSON	Vertex property list. If the list is empty, vertex properties will not be synchronized. For details about the parameters, see Table 7-434 . During the initial data synchronization, this parameter will be applied. For subsequent synchronizations, this parameter will default to the value specified during the first synchronization.
edge	No	JSON	Edge property list. If the list is empty, edge properties will not be synchronized. For details about the parameters, see Table 7-434 . During the initial data synchronization, this parameter will be applied. For subsequent synchronizations, this parameter will default to the value specified during the first synchronization.

Table 7-434 vertex and edge parameters

Parameter	Mandatory	Type	Description
label	Yes	String	Label name

Parameter	Mandatory	Type	Description
property	Yes	Array of strings	Property name. The property must belong to the label.

Response Parameters

Table 7-435 Parameter description

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

NOTE

Suspended edges in a graph database (edges that do not have either a source or a target node) are not synchronized to the HyG engine.

Example Request

Synchronize the update information of the graph database to the HyG computing engine, with an empty vertex property list, an edge property list with the property name **Rating**, and a label name of **rate**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/sync
{
  "vertex": [],
  "edge": [
    {
      "property": [
        "Rating"
      ],
      "label": "rate"
    }
  ]
}
```

```
]
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.6.3 Querying General Information About a HyG Graph

Function

This API is used to query general information about a HyG graph, such as the number of vertices, number of edges, properties, and partitioning policies.

URI

GET /ges/v1.0/{project_id}/hyg/{graph_name}/summary

Table 7-436 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-437 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
data	Object	If the query is successful, the data field is returned. For details about the parameters, see Table 7-438 .
status	String	Job status returned for a successful query. The options are pending , running , and complete . This parameter is left blank when the query fails.
result	String	Execution results - If the execution is successful, the value is success. - If the execution fails, the value is failed.

Table 7-438 data parameter description

Parameter	Type	Description
vertex	JSON	Vertex labels and properties
edge	JSON	Edge labels and properties
policy	String	Partitioning policy
inEdge	Boolean	Whether incoming edges are contained
idIndex	Boolean	Whether the vertex ID index is contained
updateTime	String	Time when a graph is updated
vertexNum	Integer	Number of vertices
edgeNum	Integer	Number of edges

Example Request

Query general information about a HyG graph.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/summary
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "data": {
    "inEdge": true,
    "idIndex": true,
    "policy": "oec",
    "updateTime": "2023-08-03 15:13:16",
    "vertex": [],
    "edge": [
      {
        "label": "rate",
        "property": [
          "Rating"
        ]
      }
    ],
    "vertexNum": 150,
    "edgeNum": 1659
  },
  "result": "success"
}
```

Status code: 400

Example response for a failed request


```
Http Status Code: 400
{
  "errorMessage": "Graph [cb1ed5491f7f466e8947ff09c06ca08c-ges_hyg12] does not exist, please check  
projectId and graphName.",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.6.4 Deleting a HyG Graph

Function

This API is used to delete a HyG graph.

URI

DELETE /ges/v1.0/{project_id}/hyg/{graph_name}

Table 7-439 URI parameters

Parameter	Mandato ry	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-440 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs.
status	String	Job status returned for a successful query. The options are pending , running , and complete . This parameter is left blank when the query fails.
result	String	Execution results - If the execution is successful, the value is success. - If the execution fails, the value is failed.

Example Request

Delete a HyG graph.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
```

```
"jobId": "fb74314e-a82d-41b2-8900-96e2559fa0d9000168232"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.6.5 Listing HyG Graphs

Function

This API is used to list HyG graphs.

URI

GET /ges/v1.0/{project_id}/hyg

Table 7-441 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .

Request Parameters

None

Response Parameters

Table 7-442 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
data	Object	If the query is successful, the data field is returned. For details about the parameters, see Table 7-443 .
result	String	Execution results - If the execution is successful, the value is success. - If the execution fails, the value is failed.

Table 7-443 data parameter description

Parameter	Type	Description
name	String	Graph name
vertex	JSON	Vertex labels and properties
edge	JSON	Edge labels and properties
policy	String	Partitioning policy
inEdge	Boolean	Whether incoming edges are contained
idIndex	Boolean	Whether the vertex ID index is contained
updateTime	String	Time when a graph is updated
vertexNum	Integer	Number of vertices
edgeNum	Integer	Number of edges

Example Request

List HyG graphs.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/hyg
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "data": [
    {
      "name": "ges_hyg",
      "inEdge": true,
      "idIndex": true,
      "policy": "oec",
      "updateTime": "2025-02-11 17:39:50",
      "vertex": [
        {
          "label": "__vertex_label__"
        }
      ],
      "edge": [
        {
          "label": "rate",
          "property": [
            "Rating"
          ]
        },
        {
          "label": "__edge_label__"
        }
      ],
      "vertexNum": 144,
      "edgeNum": 1632
    }
  ],
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 404
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.

Return Value	Description
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.6.6 Importing an HyG Graph

Function

This API is used to import HyG graph data.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/import-graph

Table 7-444 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-445 Request body parameter descriptions

Parameter	Mandatory	Type	Description
edgesetPath	Yes	String	Edge file directory or edge file name. CSV and TXT files can be imported.
vertexsetPath	Yes	String	Vertex file directory or vertex file name. CSV and TXT files can be imported.

Parameter	Mandatory	Type	Description
schemaPath	Yes	String	OBS path of the metadata file of the new data
logDir	No	String	Directory for storing logs of imported graphs. This directory stores the data that fails to be imported during graph creation and detailed error causes.
delimiter	No	Character	Field separator in a CSV file. The default value is comma (,).
trimQuote	No	Character	Field quote character in a CSV file. The default value is double quotation marks ("). It is used to enclose a field if the field contains separators or line breaks.
obsParameters	Yes	Object	OBS parameters. For details about the parameters, see Table 7-446 .
vertex	No	Object	List of vertex properties. The specified properties must belong to the schema file. If the list is empty, vertex properties will not be imported. For details about the parameters, see Table 7-447 .
edge	No	Object	List of edge properties. The specified properties must belong to the schema file. If the list is empty, edge properties will not be imported. For details about the parameters, see Table 7-447 .

Table 7-446 obsParameters parameter descriptions

Parameter	Mandatory	Type	Description
accessKey	Yes	String	User AK
secretKey	Yes	String	User SK

Table 7-447 vertex and edge parameter descriptions

Parameter	Mandatory	Type	Description
label	Yes	String	Label name

property	Yes	Array of strings	Property name. The property must belong to the label. Supported property types include string, bool, int, long, double, and float.

Response Parameters

Table 7-448 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Example Request

Import a graph. The edge file directory is **testbucket/demo_movie/edges/**, and the edge dataset format is CSV. The vertex file directory is **testbucket/demo_movie/vertices/**, and the vertex dataset format is CSV. The OBS path of the metadata file of the new data is **testbucket/demo_movie/schema.xml**, and the log storage directory is **testbucket/importlogdir**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/import-graph
{
  "edgesetPath": "testbucket/demo_movie/edges/",
  "vertexsetPath": "testbucket/demo_movie/vertices/",
  "schemaPath": "testbucket/demo_movie/schema.xml",
  "logDir": "testbucket/importlogdir",
  "delimiter": ",",
  "trimQuote": "\"",
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  },
  "vertex": [
    {
      "property": [
```



```
        "title",
        "movieid"
      ],
      "label": "movie"
    }
  ],
  "edge": [
    {
      "property": [
        "Rating",
        "Datetime"
      ],
      "label": "rate"
    }
  ]
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7 HyG Algorithm APIs

7.2.7.1 Running Algorithms

Function

This API is used to run specified algorithms based on input parameters.

URL

POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm

Table 7-449 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

For details about the parameters, see [Common algorithm parameters](#).

Response Parameters

Table 7-450 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.

Parameter	Type	Description
jobId	String	ID of the algorithm execution job. This parameter is left blank if the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "pagerank",
  "parameters": {
    "alpha": 0.85,
    "convergence": 0.00001,
    "max_iterations": 1000,
    "directed": true
  }
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

7.2.7.2 Algorithm API Parameter Reference

7.2.7.2.1 Common Algorithm Parameters

Request Parameters

Table 7-451 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name. For details, see the description of each algorithm.
parameters	No	Object	Algorithm parameters. For details, see the parameter description of each algorithm.
output	No	Object	Result output parameter. For details, see Table 7-452 .
timeout	No	Integer	Timeout interval, in seconds. The value range is (0, 2147483648).

Table 7-452 output parameter description

Parameter	Mandatory	Type	Description
format	No	String	Output format of the algorithm result. The options are JSON and TXT . The default value is JSON . For the TXT format, see Algorithm Results in TXT Format .

Parameter	Mandatory	Type	Description
mode	No	String	Output mode of algorithm results. The options include FULL and TRUNCATED. The default value is TRUNCATED. • FULL indicates that all algorithm results are output. In FULL mode, only the TXT format is supported. Algorithm results are written to the local directory of the nodes in the cluster. To access all results, you must utilize the export-result interface to write them to OBS. • TRUNCATED indicates that algorithm results are truncated and then output. Currently, only the JSON format is supported.

 NOTE

For parameters with default values, if the passed parameter value is null, it will be treated as the default value.

Example Response

Algorithms are executed based on input parameters. You can call [Querying Job Status and Execution Results](#) to use the **job_id** returned by the algorithm to obtain the algorithm execution result.

Status code: 200

Example response for a successful request

```
{
  "data": {
    "outputs": {
      $response_data //Result of each algorithm. The results vary with the algorithm.
      "runtime": 1.365867,
      "data_return_size": 3,
      "data_offset": 0,
      "data_total_size": 100
    }
  },
  "status": "complete"
}
```

 NOTE

response_data indicates the result of each algorithm. The results vary with algorithms.

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
```

```
"status": "complete",  
"errorMessage": "Unknown algorithm name: pagerank12",  
"errorCode": "GES.8301",  
"result": "failed"  
}
```

7.2.7.2.2 PageRank

Function

This API is used to run the PageRank algorithm based on input parameters.

PageRank, also known as web page ranking, is a hyperlink analysis algorithm used to rank web pages (nodes) based on their search engine results. PageRank is a way of measuring the relevance and importance of web pages (nodes).

- If a web page is linked to many other web pages, the web page is important. That is, the PageRank value of the web page is relatively high.
- If a web page with a high PageRank value is linked to another web page, the PageRank value of the linked web page increases accordingly.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-453 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-454 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	No	parameters Object	Algorithm parameters

Table 7-455 parameters

Parameter	Mandatory	Type	Description
alpha	No	Double	Weight coefficient (also called damping coefficient) The value range is (0, 1). The default value is 0.85 .
convergence	No	Double	Convergence. The value range is (0, 1). The default value is 0.00001 .
max_iterations	No	Int	Maximum iterations. An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000]. The default value is 1000 .
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is true .

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank if the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
pagerank	List	PageRank value of each node. The format is [{vertexId: rankValue},...], where, • vertexId is of the string type. • rankValue is of the double type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "pagerank",
  "parameters": {
```

```
"alpha":0.85,  
"convergence":0.00001,  
"max_iterations":1000  
}  
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200  
{  
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"  
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400  
{  
  "errorMessage": "Parameter error!",  
  "errorCode": "GES.8005"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.3 PersonalRank

Function

This API is used to run the PersonalRank algorithm based on input parameters.

PersonalRank is also called Personalized PageRank. It inherits the idea of the classic PageRank algorithm and uses the graph link structure to recursively

calculate the importance of each node. Unlike the PageRank algorithm, the PersonalRank algorithm ensures that the probability of visiting each node during random walks reflects the user's preferences. This is achieved by having each jump during the random walk return to the source node with a probability of (1-alpha).

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm

Table 7-456 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-457 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-458 parameters

Parameter	Mandatory	Type	Description
source	Yes	String	Node ID
alpha	No	Double	Weight coefficient (also called damping coefficient). The value range is (0, 1). The default value is 0.85 .
convergence	No	Double	Convergence accuracy. The value range is (0, 1). The default value is 0.00001 .
max_iterations	No	Integer	Maximum iterations. An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000]. The default value is 1000 .

directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is true .

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
source	String	Source vertex ID
personalrank	List	PersonalRank value of each node. The format is [{vertexId: rankValue},...], where, • vertexId is of the string type. • rankValue is of the double type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "personalrank",
  "parameters": {
    "source": "lili",
    "alpha": 0.85,
    "convergence": 0.00001,
    "max_iterations": 1000
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.4 K-core

Function

This API is used to run the K-core algorithm based on input parameters.

K-core is a classic graph algorithm used to calculate the number of cores of each node. The calculation result is one of the most commonly used reference values for determining the importance of a node so that the propagation capability of the node can be better understood.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-459 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-460 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-461 parameters

Parameter	Mandatory	Type	Description
k	Yes	Integer	Number of cores. The algorithm returns vertices whose number of cores is greater than or equal to k. The value must be greater than or equal to 0.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.

jobId	String	ID of the algorithm execution job. This parameter is left blank if the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
coreness	List	Coreness value ($\text{coreness} \geq k$) of each vertex, in the format of [{vertexId:corenessValue},...] , where, • vertexId is of the string type. • corenessValue is of the integer type.
kmax	Integer	Maximum number of cores

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName":"kcore",
  "parameters":{
    "k":10
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.

Return Value	Description
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.5 K-Hop

Function

This API is used to run the K-Hop algorithm based on input parameters.

This algorithm is used to search all nodes in the k layer that are associated with the source node through Breadth-First Search (BFS). The found sub-graph is the source node's ego-net. This algorithm returns the number of nodes in the ego-net.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-462 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-463 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name

parameters	Yes	parameters Object	Algorithm parameters

Table 7-464 parameters

Parameter	Mandatory	Type	Description
k	Yes	Integer	Number of hops. The value ranges from 1 to 100.
source	Yes	String	ID of the start node
mode	No	String	Search direction. Possible values are: • OUT: Hop from the outgoing edges (default value) • IN: Hop from the incoming edges. • ALL: Hop along both incoming and outgoing edges NOTE When the dataset does not contain incoming edges, if this parameter is set to OUT and an algorithm that does not depend on incoming edges is selected to calculate the output, the performance deteriorates. If this parameter is set to IN or ALL , an error is reported.
switch	No	Integer	When the number of activated edges reaches the total number of edges/switch value, the pull/push mode is switched. The value range is [1, 2000]. The default value is 40 .

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.

jobId	String	ID of the algorithm execution job. This parameter is left blank if the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
vertices	List	ID of the vertex within k hops. The format is as follows: [vertexId,...], where vertexId is of the string type.
source	String	Source vertex ID
k	Integer	Number of hops
k_hop_neighbors	Integer	Number of vertices within k hops (excluding the source vertex)

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName":"k_hop",
  "parameters":{
    "k":3,
    "source":"66",
    "mode":"ALL"
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```


Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.7.2.6 Common Neighbors of Vertex Sets

Function

This API is used to execute the Common Neighbors of Vertex Sets algorithm based on input parameters.

By selecting the common neighbors of two vertex sets, you can intuitively discover the objects associated with the two sets.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-465 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-466 Request body parameters

Parameter	Mandatory	Type	Description

algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-467 parameters

Parameter	Mandatory	Type	Description
sources	Yes	String	Set of start vertex IDs in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana . The number of IDs cannot exceed 100,000.
targets	Yes	String	Set of end vertex IDs in the standard CSV format. IDs are separated by commas (,), for example, Mike,Amy . The number of IDs cannot exceed 100,000.
restricted	No	Boolean	Whether other constraints are involved. The value can be true or false . The default value is true . • false: There is no additional constraint. The found common neighbors are the intersection of the neighborhoods corresponding to the source vertex set and target vertex set. • true: There are additional constraints. The found common neighbors are not only the intersection of the neighborhoods corresponding to the source vertex set and target vertex set, but each vertex in the common neighbor set has at least two neighboring vertices in the source vertex set and target vertex set.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.

errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
vertices	List	Common neighbor. The format is: [vertexId,...] where, vertexId is of the string type.
common_neighbors	Integer	Number of common neighbors

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "common_neighbors_of_vertex_sets",
  "parameters": {
    "sources": "Alice,Nana",
    "targets": "Mike,Amy"
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.7 Shortest Path

Function

This API is used to execute the Shortest Path algorithm based on input parameters.

The Shortest Path algorithm is used to find the shortest path between two nodes in a graph.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-468 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-469 Request body parameters

Parameter	Mandatory	Type	Description

algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-470 parameters

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex ID of a path
target	Yes	String	Target vertex ID of a path, which is not the same as the source vertex ID
directed	No	Boolean	Whether to consider the edge direction The value can be true or false . NOTE • false: The current version does not support this function on weighted graphs. • When the dataset does not contain incoming edges, if this parameter is set to true and an algorithm that does not depend on incoming edges is selected to calculate the output, the performance deteriorates. If this parameter is set to false, an error is reported.
weight	No	String	Weight of an edge The value can be empty or a string. • Empty: The default weight and distance of edges are 1. • Character string: The property of the corresponding edge is the weight. If the edge does not have a property, the weight is 1 by default.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.

jobId	String	ID of the algorithm execution job. This parameter is left blank if the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
path	List	Shortest path. The format is: [vertexId,...] where, vertexId is of the string type.
source	String	Source vertex ID
target	String	Target vertex ID

Example Request

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName":"shortest_path",
  "parameters":{
    "source":"1",
    "target":"5",
    "directed":true,
    "weight":""
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.8 All Shortest Paths

Function

This API is used to execute the All Shortest Paths algorithm based on input parameters.

The All Shortest Paths algorithm is used to find all shortest paths between two nodes in a graph.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-471 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-472 Request body parameters

Parameter	Mandatory	Type	Description

algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-473 parameters

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex ID of a path
target	Yes	String	Target vertex ID of a path, which is not the same as the source vertex ID
directed	No	Boolean	Whether to consider the edge direction The value can be true or false . The default value is false . NOTE • false: The current version does not support this function on weighted graphs. • When the dataset does not contain incoming edges, if this parameter is set to true and an algorithm that does not depend on incoming edges is selected to calculate the output, the performance deteriorates. If this parameter is set to false, an error is reported.
weight	No	String	Weight of an edge. The value can be empty or a string. The default value is empty. • Empty: The default weight and distance of edges are 1. • String: The property of the edge is the weight. If the edge does not have a property, the weight is 1 by default.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.

errorCode	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
paths	List	All shortest paths between the source vertex and target vertex. The format is: [[path1],[path2]] For the format, see Shortest Path .
paths_number	Integer	Number of paths
source	String	Source vertex ID
target	String	Target vertex ID

Example Request

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "all_pairs_shortest_paths",
  "parameters": {
    "sources": "1,2",
    "targets": "5,6",
    "directed": true,
    "cutoff": 3,
    "path_limit": 10000
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
```

```
"errorMessage": "Parameter error!",  
"errorCode": "GES.8005"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.9 All Pairs Shortest Paths

Function

This API is used to execute the All Pairs Shortest Paths algorithm based on input parameters.

This algorithm is used to search for the shortest path between any two vertices in the graph that meets the condition.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-474 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-475 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-476 parameters

Parameter	Mandatory	Type	Description
sources	Yes	String	Set of start vertex IDs. The value is in the standard CSV input format, that is, multiple vertex IDs are separated by commas (,). The number of IDs cannot exceed 100,000.
targets	Yes	String	Set of end vertex IDs. The value is in the standard CSV input format, that is, multiple vertex IDs are separated by commas (,). The number of IDs cannot exceed 100,000.
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is false . NOTE When the dataset does not contain incoming edges, if this parameter is set to true and an algorithm that does not depend on incoming edges is selected to calculate the output, the performance deteriorates. If this parameter is set to false , an error is reported.
cutoff	No	Integer	Maximum length. The value range is [1, 100]. The default value is 6 .
path_limit	No	Integer	Maximum number of paths. The value range is [1, 1000000]. The default value is 1000000 .

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
batch_paths	List	Batch path. The format is: [paths_element,...] where, Paths_element indicates the path from a source to a target. The format is: <pre>{ "paths": [["Alice", "Janet", "Sue", "Serena", "Bonnie"]], "source": "Alice", "target": "Bonnie" },</pre>
paths_number	Integer	Number of paths

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "all_pairs_shortest_paths",
  "parameters": {
    "sources": "1,2",
    "targets": "5,6",
    "directed": true,
    "cutoff": 3,
    "path_limit": 10000
  }
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.10 SSSP

Function

This API is used to execute the SSSP algorithm based on input parameters.

This algorithm provides the shortest path from a given node (source node) to all other nodes.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-477 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-478 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-479 parameters

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex ID of a path
directed	No	Boolean	Whether to consider the edge direction The value can be true or false . NOTE false: The current version does not support this function on weighted graphs. - When the dataset does not contain incoming edges, if this parameter is set to true and an algorithm that does not depend on incoming edges is selected to calculate the output, the performance deteriorates. If this parameter is set to false, an error is reported.

weight	No	String	Weight of an edge. The value can be empty or a string. • Empty: The default weight and distance of edges are 1. • String: The property of the edge is the weight. If the edge does not have a property, the weight is 1 by default.

Response Parameters

Table 7-480 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank if the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
distance	List	Path length of each vertex in the graph from the source vertex. The format is: [{vertexId:distanceValue},...], where, vertexId is of the string type. distanceValue is of the double type.
source	String	Source Vertex ID

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "sssp ",
  "parameters": {
    "source": "1",
    "directed": true,
    "weight": ""
  }
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.11 Shortest Path of Vertex Sets

Function

This API is used to execute the Shortest Path of Vertex Sets algorithm based on input parameters.

This algorithm finds the shortest path between two vertex sets.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```


Table 7-481 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-482 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-483 parameters

Parameter	Mandatory	Type	Description
sources	Yes	String	Set of start vertex IDs in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana . The number of IDs cannot exceed 100,000.
targets	Yes	String	Set of end vertex IDs in the standard CSV format. IDs are separated by commas (,), for example, Mike,Amy . The number of IDs cannot exceed 100,000.
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is false . NOTE When the dataset does not contain incoming edges, if this parameter is set to true and an algorithm that does not depend on incoming edges is selected to calculate the output, the performance deteriorates. If this parameter is set to false , an error is reported.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
path	List	Shortest path. The format is: [vertexId,...] where, vertexId is of the string type.
source	String	Source vertex ID
target	String	Target vertex ID

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName":"shortest_path_of_vertex_sets",
  "parameters":{
    "sources": "145,78,77,76",
    "targets": "23,32",
    "directed": true
  }
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.12 All Shortest Paths of Vertex Sets

Function

This API is used to execute the All Shortest Paths of Vertex Sets algorithm based on input parameters.

This algorithm finds all shortest paths between vertex sets.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-484 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-485 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	No	parameters Object	Algorithm parameters

Table 7-486 parameters

Parameter	Mandatory	Type	Description
sources	Yes	String	Set of start vertex IDs in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana . The number of IDs cannot exceed 100,000.
targets	Yes	String	Set of end vertex IDs in the standard CSV format. IDs are separated by commas (,), for example, Mike,Amy . The number of IDs cannot exceed 100,000.
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is false . NOTE When the dataset does not contain incoming edges, if this parameter is set to true and an algorithm that does not depend on incoming edges is selected to calculate the output, the performance deteriorates. If this parameter is set to false , an error is reported.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.

errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
paths	List	All shortest paths between the source vertex and target vertex. The format is: [[path1],[path2], ...] For the format, see Shortest Path .
sources	String	Source ID set of a path
targets	String	Target ID set of a path
paths_number	Integer	Number of paths

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "all_shortest_paths_of_vertex_sets",
  "parameters": {
    "sources": "145,78,77,76",
    "targets": "48,129,34,36",
    "directed": true
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
}
```

```
}  "errorCode": "GES.8005"
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.13 Connected Component

Function

This API is used to run the Connected Component algorithm based on input parameters.

A connected component stands for a sub-graph, in which all nodes are connected with each other. Path directions are involved in the strongly connected components and are not considered in the weakly connected components. This algorithm generates weakly connected components.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-487 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-488 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	No	Object	Algorithm parameters

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank if the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
Max_WCC_size	Integer	Maximum number of vertices in the largest connected component
Max_WCC_id	String	ID of the largest connected component
community	List	Connected component set (community) corresponding to each vertex, in [{vertexId:communityId},...] format. where, vertexId is of the string type. communityId is of the string type.
community_num	Integer	Number of connected components

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "connected_component",
  "parameters": {
```

```
}  
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200  
{  
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"  
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400  
{  
  "errorMessage": "Parameter error!",  
  "errorCode": "GES.8005"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.7.2.14 Triangle Count

Function

This API is used to run the Triangle Count algorithm based on input parameters.

The Triangle Count algorithm counts the number of triangles in a graph. More triangles mean higher node association degrees and closer organization relationships.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm

Table 7-489 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-490 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-491 parameters

Parameter	Mandatory	Type	Description
statistics	No	Boolean	Whether only the overall statistical results are output. The options are true and false . The default value is true . true: Export only the statistical result. false: Export the number of triangles corresponding to each vertex.

Response Parameters

Parameter	Type	Description

errorMessage	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
triangle_count	Integer	Number of triangles
vertex_triangle_s	List	Number of triangles on each vertex. The format is: [{vertexId : vertexTriangleCount},...], where, vertexId is of the string type. vertexTriangleCount is of the integer type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm
{
  "algorithmName":"triangle_count",
  "parameters":{
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
```

```
"errorMessage": "Parameter error!",  
"errorCode": "GES.8005"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.15 Closeness Centrality

Function

This API is used to execute the Closeness Centrality algorithm based on input parameters.

Closeness centrality calculates the reciprocal of the shortest distance from a group of nodes to all other nodes it can reach, then adds up and normalizes the result.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-492 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-493 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-494 parameters

Parameter	Mandatory	Type	Description
sources	Yes	String	Set of compute node IDs, in CSV format. Separate IDs with commas (,), for example, Alice,Nana .

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .

closeness	List	Closeness centrality of each node [{vertexId : closenessValue } ,...], where, vertexId is of the string type. closenessValue is of the double type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName":"closeness",
  "parameters":{
    "sources":"1,2,3"
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.16 Betweenness Centrality

Function

This API is used to execute the Betweenness Centrality algorithm based on input parameters.

Betweenness centrality is a measure of centrality in a graph based on shortest paths. This algorithm calculates shortest paths that pass through a vertex.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm

Table 7-495 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-496 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	No	parameters Object	Algorithm parameters

Table 7-497 parameters

Parameter	Mandatory	Type	Description
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is true .

weight	No	String	Weight of an edge. The value can be empty or a string. If an edge does not have the corresponding property, the weight is 1 by default. • Empty: The default weight and distance of edges are 1. • String: The property of the corresponding edge is used as the weight. NOTE 1. Default handling of missing property values is not support. An error will be reported directly. 2. HyG supports default handling of missing properties.
seeds	No	String	Vertex ID. The number of vertices cannot be greater than 100,000. If the graph is large, precise betweenness calculation can be slow. You can set seeds to the sampling vertices for approximate calculation. The more seeds vertices, the closer results to the precise calculation.
k	No	Integer	Number of samples. The value is less than or equal to 100000. If the graph is large, betweenness calculation can be slow. You can set k to randomly select k sampling vertices from the graph. The larger value, the closer results to the accurate calculation.

 NOTE

When you perform approximate betweenness calculation, either **seeds** or **k** must be specified. If both are specified, **seeds** vertices will be sampled by default and **k** will be ignored.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.

errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
betweenness	List	Betweenness value of each node. The format is [{vertexId: betweennessValue}, ...], where - vertexId is of the string type. - betweennessValue is of the double type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName":"betweenness",
  "parameters":{
    "seeds":"1,2",
    "directed":true
  }
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```


Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.17 Edge Betweenness Centrality

Function

This API is used to execute the Edge Betweenness Centrality algorithm based on input parameters.

The Edge Betweenness Centrality algorithm calculates shortest paths that pass through an edge.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-498 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-499 Request body parameters

Parameter	Mandatory	Type	Description

algorithmName	Yes	String	Algorithm name
parameters	No	parameters Object	Algorithm parameters

Table 7-500 parameters

Parameter	Mandatory	Type	Description
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is false .
weight	No	String	Weight of an edge. The value can be empty or a string. - Empty: The default weight and distance of edges are 1. - String: The property of the edge is the weight. If the edge does not have a property, the weight is 1 by default.
seeds	No	String	Vertex ID. The number of vertices cannot be greater than 100,000. If the graph is large, betweenness calculation can be slow. You can set seeds to the sampling nodes for approximate calculation. The more seeds nodes, the closer results to the accurate calculation.
k	No	Integer	Number of samples. The value is less than or equal to 100000 . If the graph is large, betweenness calculation can be slow. You can set k to randomly select k sampling vertices from the graph. The larger value, the closer results to the accurate calculation.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.

errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
edge_betweenness	List	Betweenness value of each edge. The format is [{"source": sourceId, "target": targetId, "betweenness": betweennessValue },...], where betweennessValue is of the double type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "edge_betweenness",
  "parameters": {
    "seeds": "1,2",
    "directed": true
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.18 Origin-Destination Betweenness Centrality

Function

This API is used to execute the Origin-Destination Betweenness Centrality algorithm based on input parameters.

The Origin-Destination Betweenness Centrality algorithm calculates shortest paths that pass through a vertex/edge, with the origin and destination (OD) specified.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-501 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-502 Request body parameters

Parameter	Mandatory	Type	Description

algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-503 parameters

Parameter	Mandatory	Type	Description
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is false .
weight	No	String	Weight of an edge. The value can be empty or a string. • Empty: The default weight and distance of edges are 1. • String: The property of the edge is the weight. If the edge does not have a property, the weight is 1 by default.
OD_pairs	No	String	Pairs of OD vertices The value must be in the standard CSV format. The start vertex (origin) and end vertex (destination) are separated by commas (,), and the start and end vertex pairs are separated by newline characters (\n), for example, Alice,Nana\nLily,Amy .
seeds	No	String	ID of the node where a hot spot occurs. The number of IDs cannot exceed 30. Data that will be imported when the data of OD vertex pairs is unknown. The value is in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana .
modes	No	String	Hot spot vertex type • IN: The hot spot vertex ID is used as the start vertex ID. • OUT: The hot spot vertex ID is used as the end vertex ID.

attendees	No	String	Number of participants in a hot spot corresponding to seeds . The value range is [1, 1000000]. The value is in the standard CSV format. Numbers are separated by commas (,), for example, 10,20 .

 NOTE

Either **OD_pairs** or **seeds** must be specified. If they are both specified, the **OD_pair** value prevails.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
edge_betweenness	List	Betweenness value of each edge. The format is [{"source": sourceId, "target": targetId, "betweenness": betweennessValue },...], where betweennessValue is of the double type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "od_betweenness",
  "parameters":{"directed":"true","OD_pairs": "110,78"}
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.19 Circle Detection with a Single Vertex

Function

This API is used to execute the Circle Detection with a Single Vertex algorithm based on input parameters.

This Circle Detection with a Single Vertex algorithm detects loops in a graph. Vertices on looped paths reflect the importance of the vertices.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-504 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-505 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-506 parameters

Parameter	Mandatory	Type	Description
source	Yes	String	ID of the vertex
min_circle_length	No	Integer	Minimum circle length. The value range is [3, 15]. The default value is 3 .
max_circle_length	No	Integer	Maximum circle length (min_circle_length <= max_circle_length). The value range is [3, 15]. The default value is 10 .
limit_circle_number	No	Integer	Maximum number of circles you want to search for. The value range is [1, 100000]. The default value is 100 .

Response Parameters

Parameter	Type	Description

errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
circles	List	A set of circles that contain the given vertex. The format is: [[circle1],[circle2], ...]. The circle format is as follows: [vertexId,...]. vertexId is of the string type.
source	String	ID of the vertex
circle_number	Integer	Number of circles found
runtime	Double	Algorithm running time
min_circle_length	Integer	Minimum circle length
max_circle_length	Integer	Maximum circle length
limit_circle_number	Integer	Maximum number of circles you want to search for

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName":"single_vertex_circles_detection",
  "parameters":{
    "source":"66",
    "min_circle_length":3,
    "max_circle_length":7,
    "limit_circle_number":10000
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.20 TopicRank

Function

This API is used to run the TopicRank algorithm based on input parameters.

This algorithm is one of commonly used algorithms for ranking topics by multiple dimensions. For example, this algorithm is applicable to rank complaint topics obtained through a government hotline.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-507 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-508 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-509 parameters

Parameter	Mandatory	Type	Description
sources	Yes	String	Vertex ID. You can specify multiple node IDs in CSV format and separate them with commas (,). Currently, a maximum of 100000 IDs are allowed.
actived_p	No	Double	Initial weight of the start sources vertex. The value range is [0, 100000]. The default value is 1 .
default_p	No	Double	Initial weight of a non- sources vertex. The value range is [0, 100000]. The default value is 1 .
filtered	No	Boolean	Whether results are filtered. The options are true and false . The default value is false .
only_neighbors	No	Boolean	Whether only the neighboring vertices of sources are output. The value can be true or false . The default value is true .

alpha	No	Double	Weight coefficient (also called damping coefficient). The value range is (0, 1). The default value is 0.85 .
convergence	No	Double	Convergence accuracy. The value range is (0, 1). The default value is 0.00001 .
max_iterations	No	Integer	Maximum number of iterations. The value range is [0, 2000]. The default value is 1000 .
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is true .
num_thread	No	Integer	Number of threads. The value ranges from 1 to the maximum number of CPU threads. The default value is 4 .
vertex_filter	No	Json String	This parameter is optional when operator is set to inV , outV , or bothV . For details about the format, see Table 7-510 .

Table 7-510 vertex_filter parameter descriptions

Parameter	Mandatory	Type	Description
leftvalue	No	String	Left value of the string type. For details, see Table 7-511 .

Parameter	Mandatory	Type	Description
predicate	Yes	String	Filtering type. The options are: • Logical operations: & and Note: leftvalue and rightvalue must be nested using property_filter. Only logical operations support nesting. • Comparison operations: =, !=, >, >=, <, and <= • Set operations: IN and NOTIN 1. There is a difference in semantics between checking if leftvalue (label, ID, property value) is in rightvalue (which must be an array type) and whether there is an intersection between the left and right values of the memory edition. 2. Set operations such as CONTAIN, NOTCONTAIN, and SUBSET are not supported. • Matching: rightvalue is a PREFIX (prefix), NOTPREFIX (non-prefix), SUFFIX (suffix), NOTSUFFIX (non-suffix), SUBSTRING (substring), NOTSUBSTRING (non-substring), CISUBSTRING (case-insensitive substring), FUZZY (fuzzy match), or REGEX (regular match) of leftvalue. • HAS/HASNOT: whether this property exists. Only property filtering is supported. That is, the left value can only be property_name.
rightvalue	Yes	String	Right value of a search criterion. For details about the format, see Table 7-512 .

Table 7-511 leftvalue elements

Parameter	Mandatory	Type	Description
label_name	No	String	Filter label. The value is labelName , which is optional.
property_name	No	String	Filter property. The value is the property name.

Parameter	Mandatory	Type	Description
ID	No	String	Filter ID. This parameter is optional.
property_filter	No	String	Only when predicate is set to & or , property_filter can be nested in leftvalue and rightvalue .
degree	No	String	Direction of vertex degree filtering statistics. This parameter is optional. The value can be both , in , or out .

Table 7-512 rightvalue elements

Parameter	Mandatory	Type	Description
value	Yes	String	1. Filter label. The value is the label name. 2. Filter property. The value is the property value. When predicate is HAS or HASNOT , the value is only a placeholder and has no meaning. 3. Filter ID. The value is the ID value. NOTE If predicate is set to IN or NOTIN , the value of this parameter is of the List[string] type.
property_filter	No	String	If predicate is set to & or , property_filter can be nested in leftvalue and rightvalue .

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.

jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
topicrank	List	TopicRank value of each vertex. The format is: [{vertexId:rankValue},...] where, vertexId is of the string type. rankValue is of the double type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "topicrank",
  "vertex_filter": {
    "property_filter": {
      "leftvalue": {
        "label_name": "labelName"
      },
      "predicate": "=",
      "rightvalue": {
        "value": "user"
      }
    }
  },
  "parameters": {
    "sources": "lili,andy",
    "alpha": 0.85,
    "convergence": 0.00001,
    "max_iterations": 1000,
    "filtered": "true"
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example request with **vertex_filter**

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "topicrank",
  "parameters": {
    "sources": "lili,andy",
    "alpha": 0.85,
    "convergence": 0.00001,
    "max_iterations": 1000,
    "filtered": "true",
    "vertex_filter": {
      "property_filter": {
        "leftvalue": {
          "label_name": "labelName"
        },
        "predicate": "=",

```

```
"rightvalue": {  
  "value": "user"  
}  
}  
}  
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200  
{  
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"  
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400  
{  
  "errorMessage": "Parameter error!",  
  "errorCode": "GES.8005"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.21 Louvain

Function

This API is used to run the Louvain algorithm based on input parameters.

Louvain is a modularity-based community detection algorithm with high efficiency and effect. It detects hierarchical community structures and aims to maximize the modularity of the entire community network.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm

Table 7-513 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-514 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	No	parameters Object	Algorithm parameters

Table 7-515 parameters

Parameter	Mandatory	Type	Description
convergence	No	Double	Convergence accuracy. The value range is (0, 1). The default value is 0.00001 .
max_iterations	No	Integer	Maximum iterations. An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000]. The default value is 100 .

weight	No	String	Weight of an edge. The value can be empty or a string. If this property is not configured for the edge in a graph, the algorithm reports an error. • Empty: The default weight and distance of edges are 1. • String: The property of the corresponding edge is used as the weight.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
modularity	Double	Modularity
community_num	Integer	Number of communities
community	List	Community corresponding to each vertex. The format is: [{vertexId:communityId},...] where, vertexId is of the string type. communityId is of the string type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "louvain",
  "parameters": {
    "convergence": 0.00001,
```

```
"max_iterations":100
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.22 Bigclam

Function

This API is used to run the Bigclam algorithm based on input parameters.

The BigClam algorithm is an overlapping community detection algorithm that models the relationship between vertices and communities as a bipartite graph.

Assuming that the edges between vertices in the graph are generated based on community relationships, it can detect overlapping communities in the graph.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm

Table 7-516 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-517 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-518 parameters

Parameter	Mandatory	Type	Description
convergence	No	Float	Convergence accuracy. The value range is (0, 1). The default value is 0.00001 .
max_iterations	No	Integer	Maximum iterations. An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000]. The default value is 100 .
community_num_space	Yes	String	Search space for the number of communities, with multiple integer values separated by commas (.). There can be a maximum of 100 values, and each integer value must be within the range of [1, 10000].

learning_rate	No	Float	Model learning rate. The value is greater than 0. The default value is 0.01 .

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
communities	List	Community corresponding to each vertex. The format is: [{vertexId:[communityId]}],...] where, vertexId is of the string type. communityId is of the integer type.
community_num	Integer	Number of communities
log_likelihood	double	Confidence

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName":"bigclam",
  "parameters":{
    "community_num_space":"3,2,10"
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.23 Cesna

Function

This API is used to run the Cesna algorithm based on input parameters.

The Cesna algorithm is an overlapping community detection algorithm that models the relationship between vertices and communities as a bipartite graph. Assume that the edges between vertices in the graph are generated based on community relationships. In addition, this algorithm also models communities based on vertex properties, assuming that vertex properties are also generated based on community relationships.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-519 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-520 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-521 parameters

Parameter	Mandatory	Type	Description
convergence	No	Double	Convergence accuracy. The value range is (0, 1). The default value is 0.00001 .
max_iterations	No	Integer	Maximum iterations. An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000]. The default value is 100 .
community_num_space	Yes	String	Search space for the number of communities, with multiple integer values separated by commas (,). There can be a maximum of 100 values, and each integer value must be within the range of [1, 10000].
learning_rate	No	Double	Model learning rate. The value is greater than 0. The default value is 0.01 .
holdout_rate	No	Double	Proportion of the validation set required for cross-validation, with a value range of (0, 1) and a default value of 0.1 .

node_attributes	Yes	String	Format of vertex properties, with multiple integers separated by semicolons (;). Each integer represents the index of an element with a value of 1 in a binary array. For example, 1;2;3 means that the vertex has a value of 1 in dimensions 1, 2, and 3, and a value of 0 in all other dimensions. The index of vertex properties is greater than or equal to 0 and less than 10,000, and the number of indexes is less than or equal to 10,000.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
communities	List	Community corresponding to each vertex. The format is: [{vertexId:[communityId]}],...] where, vertexId is of the string type. communityId is of the integer type.
community_num	Integer	Number of communities
log_likelihood	double	Confidence

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName":"cesna",
```



```
"parameters":{  
  "community_num_space":"3,2,10",  
  "node_attributes": "weight"  
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200  
{  
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"  
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400  
{  
  "errorMessage": "Parameter error!",  
  "errorCode": "GES.8005"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.24 Infomap

Function

This API is used to run the Infomap algorithm based on input parameters.

The Infomap algorithm is a community detection algorithm based on information theory. It performs well in terms of efficiency and effectiveness, and can discover

hierarchical community structures. Its optimization goal is to find the optimal community structure that minimizes the length of the hierarchical encoding of vertices.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm

Table 7-522 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-523 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	No	parameters Object	Algorithm parameters

Table 7-524 parameters

Parameter	Mandatory	Type	Description
convergence	No	Double	Convergence accuracy. The value range is (0, 1). The default value is 0.00001 .
max_iterations	No	Integer	Maximum iterations. An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000]. The default value is 100 .

weight	No	String	Weight of an edge. The value can be empty or a string. If this property is not configured for the edge in a graph, the algorithm reports an error. • Empty: The default weight and distance of edges are 1. • String: The property of the corresponding edge is used as the weight.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
min_code_length	Double	Minimum code length
community_num	Integer	Number of communities
community	List	Community corresponding to each vertex. The format is: [{vertexId:communityId},...] where, vertexId is of the string type. communityId is of the string type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName":"infomap",
```

```
"parameters":{
  "convergence":0.00001,
  "max_iterations":100
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.25 Label Propagation

Function

This API is used to run the Label Propagation algorithm based on input parameters.

The Label Propagation algorithm is a graph-based semi-supervised learning method. Its basic principle is to predict the label information about unlabeled nodes using that of the labeled nodes. This algorithm can create graphs based on the relationships between samples. Nodes include labeled data and unlabeled data, and the edge indicates the similarity between two nodes. Node labels are transferred to other nodes based on the similarity. Labeled data is like a source used to label unlabeled data. The greater the node similarity is, the easier the label propagation will be.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm

Table 7-525 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-526 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	No	parameters Object	Algorithm parameters

Table 7-527 parameters

Parameter	Mandatory	Type	Description
convergence	No	Double	Convergence accuracy. The value range is (0, 1). The default value is 0.00001 .
max_iterations	No	Integer	Maximum iterations. An integer ranging from 1 to 2147483647. For frontend calls, the range is [1, 2000]. The default value is 1000 .

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
community	List	Community corresponding to each vertex. The format is: [{vertexId:communityId},...] where, vertexId is of the string type. communityId is of the string type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName":"label_propagation",
  "parameters":{
    "convergence":"0.00001",
    "max_iterations":"1000"
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.26 Subgraph Matching

Function

This API is used to run the Subgraph Matching algorithm based on input parameters.

The subgraph matching algorithm is used to find all subgraphs of a given small graph that is isomorphic to a given large graph. This is a basic graph query operation and is intended to explore important substructures of a graph.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-528 URI parameters

Name	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-529 Request body parameters

Name	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-530 parameters

Name	Mandatory	Type	Description
edges	Yes	String	Set of edges of the subgraphs to be matched. The value is in standard CSV format. The start and end vertices of an edge are separated by a comma (,). Edges are separated by a newline character (\n), for example, 1,2\n2,3 .
vertices	Yes	String	Label of each vertex on the subgraph to be matched. The value is in standard CSV format. Vertices and their labels are separated by commas (,), and labels are separated by newline characters (\n). For example, 1,BP\n2,FBP\n3,CP .
n	No	Integer	Maximum number of subgraphs to be searched. The value range is [1, 100000], with 100 as the default.
batch_number	No	Integer	Number of subgraphs that are batch processed in each round. The value range is [1, 1000000]. The default value is 10000 .
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . The default value is true .
statistics	No	Boolean	Whether the number of all subgraphs that meet the criteria is output. The value can be true or false . The default value is false .

Response Parameters

Name	Type	Description
errorMessage	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
subgraphs	List	Subgraphs with the same pattern of the pattern_graph . The value is in the [[subgraph1], [subgraph2], ...] format. Each subgraph is in the [vertex1,vertex2, ...] format, where vertex is of the string type. The vertices of each subgraph correspond to those of pattern_graph .
pattern_graph	List	Graph pattern. The value is in the [vertex1,vertex2, ...] format, where vertex is of the string type.
subgraph_number	Integer	Number of matched graphs. When statistics is set to true , the total number of graphs that meet search criteria is displayed.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/ algorithm
{
  "algorithmName": "subgraph_matching",
  "parameters": {
    "edges": "1,2\n2,3\n3,4\n5,6\n5,4",
    "vertices": "1,movie\n2,user\n3,user\n4,user\n5,user\n6,user",
    "statistics": "true",
    "directed": "true",
    "n": 55,
    "batch_number": "500"
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.27 Link Prediction

Function

This API is used to run the Link Prediction algorithm based on input parameters.

This algorithm is used to calculate the similarity between two nodes and predict their relationship based on the Jaccard measurement method.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-531 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-532 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-533 parameters

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex ID
target	Yes	String	Target vertex ID, which is not the same as the source vertex ID

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.

jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
source	String	Source vertex ID
target	String	Target vertex ID
link_prediction	Double	Link prediction result

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/ algorithm
{
  "algorithmName": "link_prediction",
  "parameters": {
    "source": "3",
    "target": "15"
  }
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.

Return Value	Description
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.28 n Paths

Function

This API is used to run the n Paths algorithm based on input parameters.

The n Paths algorithm is used to find the n paths between two vertices on the k layer of a graph.

URI

```
POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
```

Table 7-534 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-535 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters Object	Algorithm parameters

Table 7-536 parameters

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex ID of a path
target	Yes	String	Target vertex ID of a path, which is not the same as the source vertex ID
directed	No	Boolean	Whether to consider the edge direction. The value can be true or false . NOTE false: The current version does not support this function on weighted graphs. - When the dataset does not contain incoming edges, if this parameter is set to true and an algorithm that does not depend on incoming edges is selected to calculate the output, the performance deteriorates. If this parameter is set to false, an error is reported.
n	No	Integer	Number of paths. The value range is [1, 100]. The default value is 10 .
k	No	Integer	Number of layers. The value range is [1, 10]. The default value is 5 .

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .

paths	List	Paths between the source and target vertices. The format is: [[path1],[path2], ...] For the format, see Shortest Path .
paths_number	Integer	Number of paths
source	String	Source vertex ID
target	String	Target vertex ID

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "n_paths",
  "parameters": {
    "source": "129",
    "target": "78",
    "directed": "false",
    "n": "50",
    "k": "3"
  }
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.

Return Value	Description
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.29 Cluster Coefficient

Function

This API is used to run the Cluster Coefficient algorithm based on input parameters.

This algorithm is used to calculate the aggregation degree of vertices in a graph.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm

Table 7-537 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-538 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	No	parameters Object	Algorithm parameters

Table 7-539 parameters

Parameter	Mandatory	Type	Description
statistics	No	Boolean	Whether only the overall statistical results are output. The options are true and false . The default value is true . • true: Only the total average clustering coefficient is output. • false: The clustering coefficient corresponding to each vertex is additionally output.
directed	No	Boolean	Whether a graph is considered as a directed graph for calculation. The value can be true or false . The default value is false . • true: directed graph • false: undirected graph

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
cluster_coefficient	Double	Clustering coefficient

vertex_cluster_coefficient	List	Clustering coefficient of each vertex. The format is: [{vertexId: cluster_coefficient },...] where, vertexId is of the string type. cluster_coefficient is of the double type.

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "cluster_coefficient",
  "parameters": {
    "statistics": "false",
    "directed": "false"
  }
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.

Return Value	Description
503 Service Unavailable	Service unavailable.

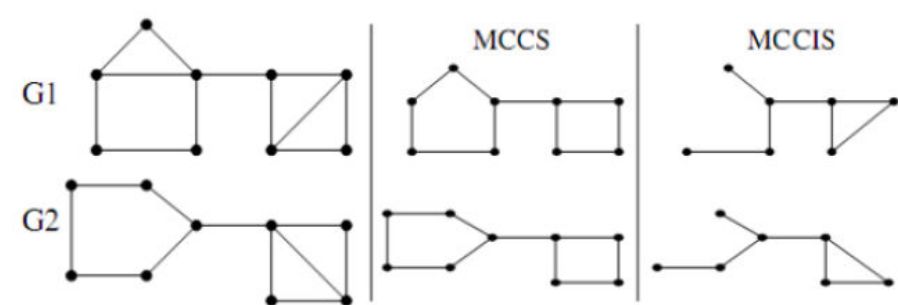
Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.2.30 Maximum Common Connected Induced Subgraph (MCCIS)

Function

The MCCIS algorithm is suitable for calculating the similarity between two graphs.



similarity = 2*|E(mccis)| / (|E(G1)| + |E(G2)|)

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/algorithm

Table 7-540 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-541 Request body parameters

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name
parameters	Yes	parameters object	Algorithm parameters

Table 7-542 parameters

Parameter	Mandatory	Type	Description
vertex_list_1	Yes	String	Sequence of vertices in subgraph 1. There can be up to 10,000 vertices.
vertex_list_2	Yes	String	Sequence of vertices in subgraph 2. There can be up to 10,000 vertices.
recursion_max_number	No	Integer	Maximum number of recursions. The value range is [10000, 100000000], with 1000000 as the default.
directed	No	Boolean	Whether an edge is directed. The options are true and false , with false as the default.

Response Parameters

Parameter	Type	Description
errorMessage	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If the execution succeeds, this parameter may be left blank. If the execution fails, this parameter is used to display the error code.

Parameter	Type	Description
jobId	String	ID of the algorithm execution job. This parameter is left blank if the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .
matches	List	Matched vertex pairs: [{vertex1: vertex2}, ...], where vertex is of the string type Vertex 1 belongs to subgraph 1, and vertex 2 belongs to subgraph 2.
similarity	Double	Similarity between two graphs

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm
{
  "algorithmName": "mccis",
  "parameters": {
    "vertex_list_1": "46,47,48,49,38",
    "vertex_list_2": "50,51,52,42"
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.3 Algorithm Results in TXT Format

Table 7-543 Algorithm results in TXT format

Algorithm	Support ed On	header	content	e.g.
all_pairs_short est_paths	Local host, OBS	# runtime: {runtime} # paths_number: {paths_number} # data_total_size: {data_total_size} # data_return_size: {data_return_size} # data_offset: {data_offset} # batch_paths:	Each line contains multiple paths of a pair. The format is as follows: {sourceID}, {targetID}, "[[{sourceID}, {v1},..., {targetID}],...]"	# runtime: 4.411 # paths_number: 20 # data_total_size: 25 # data_return_size: 25 # data_offset: 0 # batch_paths: "121","66","[["121","25","66"]]"

Algorithm	Supported On	header	content	e.g.
all_shortest_paths	Local host, OBS	# runtime: {runtime} # source: {source} # target: {target} # paths_number: {paths_number} # data_total_size: {data_total_size} # data_return_size: {data_return_size} } # data_offset: {data_offset} # paths:	Each line is a path. The format is as follows: {sourceID}, {vertexID1},..., {targetID}	# runtime: 0.207 # source: 121 # target: 66 # paths_number: 2 # data_total_size: 2 # data_return_size: 2 # data_offset: 0 # paths: 121,7,66 121,25,66
all_shortest_paths_of_vertex_sets	Local host, OBS	# runtime: {runtime} # source: {source} # target: {target} # paths_number: {paths_number} # data_total_size: {data_total_size} # data_return_size: {data_return_size} } # data_offset: {data_offset} # paths:	Each line is a path. The format is as follows: {sourceID}, {vertexID1},..., {targetID}	# runtime: 2.772 # sources: 48,129,34,36 # targets: 46,66,101 # paths_number: 15 # data_total_size: 15 # data_return_size: 15 # data_offset: 0 # paths: 36,72,101 36,59,46 36,73,46

Algorithm	Support ed On	header	content	e.g.
betweenness	Local host, OBS	# runtime: {runtime} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # betweenness:	{vertexID}, {betweenness}	# runtime: 1.593 # data_total_size: 32 # data_return_siz e: 32 # data_offset: 0 # betweenness: 79,20.69722222 222223 80,12.29058441 5584414 81,1.5
bigclam	Local host, OBS	# runtime: {runtime} # community_num : {community_nu m} # log_likelihood: {log_likelihood} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # communities:	{vertexID}, {community}	# runtime: 2.754 # community_nu m: 1 # log_likelihood: -5593.4549824 494925 # data_total_size: 32 # data_return_siz e: 32 # data_offset: 0 # communities: 6,0 13,0

Algorithm	Support ed On	header	content	e.g.
cesna	Local host, OBS	# runtime: {runtime} # community_num : {community_nu m} # log_likelihood: {log_likelihood} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # communities:	{vertexID}, {community}	# runtime: 40114.213 # community_nu m # log_likelihood # data_total_size: 1344 # data_return_siz e: 1344 # data_offset: 0 # communities: 3850,3 3858,3 3866,3
closeness	Local host, OBS	# runtime: {runtime} # source: {source} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # closeness:	{closeness}	# runtime: 0.394 # source: 12 # data_total_size: 1 # data_return_siz e: 1 # data_offset: 0 # closeness: 0.50877192982 45614

Algorithm	Support ed On	header	content	e.g.
cluster_coefficient (statistic = true)	Local host, OBS	# runtime: {runtime} # cluster_coefficient: {cluster_coefficient} # data_total_size: {data_total_size} # data_return_size: {data_return_size} # data_offset: {data_offset} # vertex_cluster_coefficient:	{vertexID}, {cluster_coefficient}	# runtime: 0.661 # cluster_coefficient: 0.13517429595852912 # data_total_size: 32 # data_return_size: 32 # data_offset: 0 # vertex_cluster_coefficient:
common_neighbors_of_vertex_sets	Local host, OBS	# runtime: {runtime} # common_neighbors: {common_neighbors} # data_total_size: {data_total_size} # data_return_size: {data_return_size} # data_offset: {data_offset} # vertices:	{vertexID}	# runtime: 0.42 # common_neighbors: 26 # data_total_size: 26 # data_return_size: 26 # data_offset: 0 # vertices: 103 138 98

Algorithm	Support ed On	header	content	e.g.
connected_component	Local host, OBS	# runtime: {runtime} # community_num: {community_num} # Max_WCC_size: {Max_WCC_size} # Max_WCC_id: {Max_WCC_id} # data_total_size: {data_total_size} # data_return_size: {data_return_size} # data_offset: {data_offset} # community:	{vertexID}, {community}	# runtime: 0.263 # community_num: 1 # Max_WCC_size # Max_WCC_id # data_total_size: 32 # data_return_size: 32 # data_offset: 0 # community: 2,0 6,0 13,0
edge_betweenness	Local host, OBS	# runtime: {runtime} # data_total_size: {data_total_size} # data_return_size: {data_return_size} # data_offset: {data_offset} # edge_betweenness:	{sourceID}, {targetID}, {edge_betweenness}	# runtime: 153.006 # data_total_size: 311 # data_return_size: 311 # data_offset: 0 # edge_betweenness: 51,20,1.333333 3333333333 51,33,7.192099 567099566 51,10,3.476190 4761904763

Algorithm	Support ed On	header	content	e.g.
infomap	Local host, OBS	# runtime: {runtime} # min_code_length : {min_code_lengt h} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # community:	{vertexID}, {community}	# runtime: 98.158 # min_code_lengt h: 6.26800955194 43135 # data_total_size: 32 # data_return_siz e: 32 # data_offset: 0 # community: 2,200000000055 6,200000000050 13,20000000001 4
k_hop	Local host, OBS	# runtime: {runtime} # source: {source} # k: {k} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # vertices:	{vertexID}	# runtime: 0.442 # source: 76 # k: 6 # data_total_size: 32 # data_return_siz e: 32 # data_offset: 0 # vertices: 2 6 13

Algorithm	Support ed On	header	content	e.g.
kcore	Local host, OBS	# runtime: {runtime} # kmax: {kmax} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # coreness:	{vertexID}, {coreness}	# runtime: 10.882 # kmax: 15 # data_total_size: 32 # data_return_siz e: 32 # data_offset: 0 # coreness: 2,14 6,15 13,15
label_propagat ion	Local host, OBS	# runtime: {runtime} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # community:	{vertexID}, {community}	# runtime: 2.624 # data_total_size: 32 # data_return_siz e: 32 # data_offset: 0 # community: 2,10000000024 6,10000000024 13,1000000002 4

Algorithm	Supported On	header	content	e.g.
link_prediction	Local host, OBS	# runtime: {runtime} # source: {source} # target: {target} # data_total_size: {data_total_size} # data_return_size: {data_return_size} # data_offset: {data_offset} # link_prediction:	{link_prediction}	# runtime: 0 # source: 123 # target: 43 # data_total_size: 1 # data_return_size: 1 # data_offset: 0 # link_prediction: 0.07017543859649122
louvain	Local host, OBS	# runtime: {runtime} # modularity: {modularity} # data_total_size: {data_total_size} # data_return_size: {data_return_size} # data_offset: {data_offset} # community:	{vertexID}, {community}	# runtime: 45.835 # modularity: 0.16375671670152867 # data_total_size: 32 # data_return_size: 32 # data_offset: 0 # community: 2,200000000626,2000000005013,20000000050

Algorithm	Support ed On	header	content	e.g.
n_paths	Local host, OBS	# runtime: {runtime} # source: {source} # target: {target} # paths_number: {paths_number} # data_total_size: {data_total_size} # data_return_size: {data_return_size} } # data_offset: {data_offset} # paths:	Each line is a path. The format is as follows: {sourceID}, {vertexID1},..., {targetID}	# runtime: 8.025 # source: 123 # target: 87 # paths_number: 100 # data_total_size: 100 # data_return_siz e: 100 # data_offset: 0 # paths: 123,21,87 123,13,87 123,32,87
od_betweenne ss	Local host, OBS	# runtime: {runtime} # data_total_size: {data_total_size} # data_return_size: {data_return_size} } # data_offset: {data_offset} # edge_betweenne ss:	{sourceID}, {targetID}, {edge_betwee nness}	# runtime: 1.391 # data_total_size: 311 # data_return_siz e: 311 # data_offset: 0 # edge_betweenne ss: 51,20,0 51,33,0 51,10,0

Algorithm	Support ed On	header	content	e.g.
pagerank	Local host, OBS	# runtime: {runtime} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # pagerank:	{vertexID}, {pagerank}	# runtime: 4.044 # data_total_size: 32 # data_return_siz e: 32 # data_offset: 0 # pagerank: 2,0.0078889040 51903298 6,0.0132158636 92849642 13,0.018605301 99450448
personalrank	Local host, OBS	# runtime: {runtime} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # personalrank:	{vertexID}, {personalrank}	# runtime: 2.326 # source: 46 # data_total_size: 49 # data_return_siz e: 49 # data_offset: 0 # personalrank: 0,0.0021350905 350732297 1,0.0045911514 06893241

Algorithm	Supported On	header	content	e.g.
shortest_path	Local host, OBS	# runtime: {runtime} # source: {source} # target: {target} # data_total_size: {data_total_size} # data_return_size: {data_return_size} # data_offset: {data_offset} # path:	Each line is a path. The format is as follows: {sourceID}, {vertexID1},..., {targetID}	# runtime: 0.308 # source: 123 # target: 5 # data_total_size: 1 # data_return_size: 1 # data_offset: 0 # path: 123,10,137,5
shortest_path_of_vertex_sets	Local host, OBS	# runtime: {runtime} # source: {source} # target: {target} # data_total_size: {data_total_size} # data_return_size: {data_return_size} # data_offset: {data_offset} # path:	Each line is a path. The format is as follows: {sourceID}, {vertexID1},..., {targetID}	# runtime: 1.832 # source: 24 # target: 121 # data_total_size: 1 # data_return_size: 1 # data_offset: 0 # path: 24,121

Algorithm	Support ed On	header	content	e.g.
single_vertex_c ircles_detectio n	Local host, OBS	# runtime: {runtime} # source: {source} # min_circle_lengt h: {min_circle_lengt h} # max_circle_lengt h: {max_circle_lengt h} # limit_circle_num ber: {limit_circle_num ber} # circle_number: {circle_number} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # circles:	Each line is a path. The format is as follows: {sourceID}, {vertexID1},..., {sourceID}	# runtime: 37.46 # source: 122 # target: # min_circle_lengt h: 3 # max_circle_lengt h: 10 # limit_circle_nu mber: 100 # circle_number: 100 # data_total_size: 100 # data_return_siz e: 100 # data_offset: 0 # circles: 122,82,79,76,65 ,122 122,125,135,77, 65,122 122,82,114,96,6 5,122

Algorithm	Support ed On	header	content	e.g.
sssp	Local host, OBS	<pre># runtime: {runtime} # source: {source} # data_total_size: {data_total_size} # data_return_size: {data_return_size} } # data_offset: {data_offset} # distance:</pre>	<pre>{vertexID}, {distance}</pre>	<pre># runtime: 0.452 # source: 32 # data_total_size: 48 # data_return_size: 48 # data_offset: 0 # distance: 0,2 5,2 7,2</pre>

Algorithm	Support ed On	header	content	e.g.
subgraph_mat ching	Local host, OBS	# runtime: {runtime} # pattern_graph: {pattern_graph} # data_total_size: {data_total_size} # data_return_size: {data_return_size } # data_offset: {data_offset} # subgraphs:	Each line is a matched subgraph. The format is as follows: {vertexID1}, {vertexID2},..., {vertexIDn}	----- statistics = true----- # runtime: 1.376 # pattern_graph: 2,3,1 # data_total_size: 1 # data_return_siz e: 1 # data_offset: 0 # subgraph_num ber: 1556 ----- statistics = false----- # runtime: 0.956 # pattern_graph: 2,3,1 # subgraph_num ber: 0 # data_total_size: 100 # data_return_siz e: 100 # data_offset: 0 # subgraphs: 0,51,126 0,51,131 0,126,113

Algorithm	Support ed On	header	content	e.g.
topic_rank	Local host, OBS	# runtime: {runtime} # data_total_size: {data_total_size} # data_return_size: {data_return_size} } # data_offset: {data_offset} # topicrank:	{vertexID}, {topicrank}	# runtime: 1.11 # data_total_size: 32 # data_return_size: 32 # data_offset: 0 # topicrank: 2,0.0066306827 4092574 6,0.0072781302 08954746 13,0.007869137 668788257
triangle_count (statistic = true)	Local host, OBS	# runtime: {runtime} # triangle_count: {triangle_count} # data_total_size: {data_total_size} # data_return_size: {data_return_size} } # data_offset: {data_offset} # vertex_triangles:	{vertexID}, {vertex_triangles}	# runtime: 0.491 # triangle_count: 1653 # data_total_size: 32 # data_return_size: 32 # data_offset: 0 # vertex_triangles :

NOTE

Example response for a failed algorithm request:

```
Http Status Code: 400
{
  "errorMessage": "Unsupported output file format",
  "errorCode": "GES.8301"
}
```

7.2.7.4 Executing the DSL Algorithm

Function

This API is used to provide flexible DSLs to help users design and run algorithms at low costs. For details about the DSL algorithm, see [DSL Syntax](#).

NOTE

After the DSL algorithm is executed, you need to use the "Dumping HyG Algorithm Results" API to dump the DSL execution results to OBS. After the dump, you can view algorithm results in files such as **stdout**. Because HyG graphs are distributed, there may be multiple result files, which correspond to the results of different partitions.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/dsl

Table 7-544 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-545 Request body parameters

Parameter	Mandatory	Type	Description
scriptPath	Yes	String	Path of the DSL algorithm file that the user has written.
obsParameters	Yes	Object	OBS authentication parameters. For details, see Table 7-546 .
timeout	No	Integer	Timeout interval, in seconds. The value range is (1, 2147483647).

Table 7-546 obsParameters parameter description

Parameter	Mandatory	Type	Description
accessKey	Yes	String	AK

secretKey	Yes	String	SK

Response Parameters

Table 7-547 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
result	String	Execution results - If the execution is successful, the value is success. - If the execution fails, the value is failed.

Example Request

Example request 1: Cancel a submitted job.
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/dsl
{
 "scriptPath": "bucket/run_sssp.py",
 "obsParameters": {
 "accessKey": "XXX",
 "secretKey": "XXX"
 }
}

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200
{
 "jobId": "6-57222f3d-f6b8-41ba-b492-60ed9b879223"
}

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.7.5 DSL Syntax**7.2.7.5.1 Graph Operation APIs**

Graph operation APIs aim to provide users with an end-to-end process APIs for input, computation, and output.

Graph Property Value Type

The current Python DSL supports three data types: int, float, and bool, which correspond to the basic data types int64_t, double, and bool in C++.

Combiner Type

The Combiner is used to perform local calculations on data during the computation process that satisfies the commutative and associative laws, reducing data exchange. The current Python DSL provides three types of Combiners: sum, max, and min, which support local calculations on int and float data.

Graph Loading

```
hyg.analytics.graph.load_base_graph(graph_name:str)
```

This API returns a BaseGraph object, where graph_name specifies the graph name.

7.2.7.5.2 API for Running Custom Algorithms (Currently, the Pregel Programming Model Is Supported)

When the built-in graph analysis algorithms cannot meet the user's needs, HyG allows users to implement custom algorithms using the Python language in the Pregel programming model. The API for running custom Pregel algorithms is:

```
BaseGraph.run_pregel(model:class, result_filter=None, debug_mode=False)
```

The **model** is a class type and a subclass of **hyg.analytics.model.PregelModel**. Before calling the **run_pregel** interface, you need to first implement the algorithm calculation logic in your own PregelModel subclass, for example, the Pregel programming model. First, users need to specify the vertex value type (**ntype**) and message type (**mtype**) in the **@pregel_type** decorator. The **mtype** can be left unset, and it will default to the same as **ntype**. Next, users need to implement the vertex-centered **init** and **compute** methods in **UserPregelAlgorithm**. The **init** method is executed only once at the beginning of the algorithm, while the **compute** method is iterated multiple times. If these methods are not implemented, they default to empty. Additionally, the **combiner** parameter combines messages sent to the same target vertex to reduce communication overhead. By default, it is set to **None**, which means messages are not combined.

Pregel programming model:

```
from hyg.analytics.model import pregel_types, PregelModel
@pregel_types(ntype=None, mtype=None, combiner=None)
class PregelModel:
    @staticmethod
    def init(ctx, nid):
        pass

    @staticmethod
    def compute(ctx, nid, msgs):
        pass
```

The **result_filter** parameter is of type **function** and supports lambda functions. It takes (**ctx, nid**) as input and returns a boolean value, which is used to filter the Pregel calculation results. The **debug_mode** is of type **bool**. When set to **True**, the UDF is not compiled into native code but is interpreted by the Python interpreter. In this case, the number of concurrent threads in the HyG framework process is forced to 1, and users can use **print** statements to print debugging information in the UDF. When **debug_mode** is set to **False**, using **print** statements in the UDF will result in an error.

In addition, the BaseGraph also has an interface **BaseGraph.nid(ext_id:str) -> int**, which is used to obtain the internal ID of a vertex.

7.2.7.5.3 Pregel Programming API

When implementing the **init** and **compute** methods in **UserPregelAlgorithm**, users mainly rely on the **PregelContext** object, which provides the following APIs:

Table 7-548 PregelContext API

Method and Attribute	Description	Remarks
ext_id(nid)->int	Obtains the user-defined external ID of the current vertex. Only IDs that can be converted to the int type are supported.	Basic operations on graph data (values and topologies)
value(nid)->int/float/bool	Obtains the value of the current vertex.	
set_value(nid, new_value)->None	Sets the value of the current vertex.	
out_edges(nid)-> List[int]	Obtains the list of outgoing edges of the current vertex.	
edge_dst(eid)->int	Obtains the target vertex of the current edge.	
num_nodes	Obtains the number of vertices in a full graph.	
num_edges	Obtains the total number of edges in a full image.	
send(dst_nid, msg)->None	Sends messages to the target vertex.	Data exchange operation between vertices
halt(nid) ->None	Sets the current vertex to the halt state.	When all vertices in the graph are in the halt state and there are no messages, or when supersteps reach the maximum number of iterations, the algorithm terminates.
superstep -> int	Obtain the current number of supersteps.	

7.2.7.5.4 Programming Example of Creating Custom Graph Analysis Algorithms

Creating a Custom SSSP Algorithm

```
# Import necessary packages.
from hyg.analytics.graph import load_base_graph
from hyg.analytics.model import pregel_types, PregelModel

# Set graph_name to load graph data.
graph = load_base_graph("movie")
# Obtain the internal ID based on the external ID.
SOURCE_NODE = graph.nid(100)

# Implement the custom SSSP algorithm based on the Pregel model and set ntype to int.
# By default, set mttype to the same value as ntype, and set combiner to min.
@pregel_types(ntype=int, combiner=min)
```

```
class PregelSSSP(PregelModel):
    @staticmethod
    def compute(ctx, nid, msgs):
        if ctx.superstep == 0:
            ctx.set_value(nid, 10000)

        min_dist = 0 if nid == SOURCE_NODE else 10000

        if len(msgs) != 0:
            min_dist = min(min_dist, min(msgs))

        if min_dist < ctx.value(nid):
            ctx.set_value(nid, min_dist)
            for e in ctx.out_edges(nid):
                ctx.send(ctx.edge_dst(e), min_dist + 1)

        ctx.halt(nid)

# Run the custom SSSP algorithm and obtain the result.
result = graph.run_pregel(PregelSSSP)
print(type(result), result)
```

Creating a Custom PageRank Algorithm

```
# Import necessary packages.
from hyg.analytics.graph import load_base_graph
from hyg.analytics.model import pregel_types, PregelModel

# Set graph_name to load graph data.
graph = load_base_graph("movie")

# Implement the custom PageRank algorithm based on the Pregel model.
# Set ntype to float and retain the default value for mtype.
# Set combiner to sum.
@pregel_types(ntype=float, combiner=sum)
class PregelPageRank(PregelModel):
    @staticmethod
    def init(ctx, nid):
        ctx.set_value(nid, 1.0)

    @staticmethod
    def compute(ctx, nid, msgs):
        if ctx.superstep >= 1:
            new_value = 0.85 * sum(msgs) + 0.15 / ctx.num_nodes

            if (abs(new_value - ctx.value(nid)) < 0.001
                or ctx.superstep == 1000):
                ctx.halt(nid)
                return

            ctx.set_value(nid, new_value)
            # Only when debug_mode is set to True,
            # You can use print debugging statements in the custom algorithm.
            print(f'in step {ctx.superstep}, node nid {nid}, '
                  f'value {ctx.value(nid)}")
            out_edges = ctx.out_edges(nid)
            if len(out_edges) > 0:
                new_msg = ctx.value(nid) / len(out_edges)
                for e in out_edges:
                    ctx.send(ctx.edge_dst(e), new_msg)

# Run the custom PageRank algorithm in debug mode and obtain the result.
result = graph.run_pregel(PregelPageRank, debug_mode=True)
print(type(result), result)
```

Creating a Custom K-Hop Algorithm

```
# Import necessary packages.
from hyg.analytics.graph import load_base_graph
```

```
from hyg.analytics.model import pregel_types, PregelModel

# Set graph_name to load graph data.
graph = load_base_graph("movie")

# Set global values for the custom algorithm.
INFINITY_VALUE = False
SOURCE_NODE = graph.nid("100")
MAX_HOP = 3

# Implement the custom K-Hop algorithm based on the Pregel model.
# Set ntype to bool, indicating whether the vertex value is traversed.
# The parameter combiner cannot be set to bool. Even if it is set, the value will be ignored.
@pregel_types(ntype=bool)
class PregelKHop(PregelModel):
    @staticmethod
    def compute(ctx, nid, msgs) -> None:
        if ctx.superstep == 0:
            if nid == SOURCE_NODE:
                ctx.set_value(nid, True)
            if ctx.superstep < MAX_HOP:
                for e in ctx.out_edges(nid):
                    ctx.send(ctx.edge_dst(e), True)
        else:
            ctx.set_value(nid, INFINITY_VALUE)
        else:
            if ctx.value(nid) == INFINITY_VALUE:
                ctx.set_value(nid, True)
            if ctx.superstep < MAX_HOP:
                for e in ctx.out_edges(nid):
                    ctx.send(ctx.edge_dst(e), True)

        ctx.halt(nid)

# Run the custom K-Hop algorithm and filter the results using the result_filter parameter.
result = graph.run_pregel(PregelKHop,
                           result_filter=
                               lambda ctx, nid: nid != SOURCE_NODE
                                   and ctx.value(nid))
print(len(result), result.keys())
```

7.2.8 HyG Job Management APIs

7.2.8.1 Dumping HyG Algorithm Results

Function

This API is used to dump the execution results of the algorithm (jobId) to OBS.

URI

POST /ges/v1.0/{project_id}/hyg/{graph_name}/jobs/{job_id}/export-result

Table 7-549 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .

Parameter	Mandatory	Type	Description
graph_name	Yes	String	Graph name
job_id	Yes	String	Job ID of the algorithm task in the response result

Request Parameters

Table 7-550 Request body parameters

Parameter	Mandatory	Type	Description
exportPath	Yes	String	Dump path
obsParameters	Yes	String	OBS authentication parameters. For details, see Table 7-551 .
erase	No	Boolean	Whether to delete the original job result after dumping. The value can be true or false . The default value is true , indicating that the job result is deleted and resources are released by default.

Table 7-551 obsParameters parameter description

Parameter	Mandatory	Type	Description
accessKey	Yes	String	AK
secretKey	Yes	String	SK

Response Parameters

Table 7-552 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt code - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job. This parameter is left blank if the request fails. NOTE You can use the returned job ID to view the task execution status and obtain the algorithm return result. For details, see Querying Job Status on the Service Plane (1.0.0) .

Example Request

Dumps algorithm results and returns the job ID.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/jobs/{job_id}/export-result
{
  "exportPath": "demo_movie/",
  "erase": true,
  "obsParameters": {
    "accessKey": "xxxx",
    "secretKey": "xxxx"
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.8.2 Canceling a HyG Job

Function

This API is used to cancel a submitted HyG job.

NOTE

Canceling a running job does not terminate immediately and may have a delay.

URI

DELETE /ges/v1.0/{project_id}/hyg/{graph_name}/jobs/{job_id}

Table 7-553 URI parameters

Parameter	Mandator y	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
job_id	Yes	String	Job ID of the algorithm task in the response result

Request Parameters

None

Response Parameters

Table 7-554 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
result	String	Execution results - If the execution is successful, the value is success. - If the execution fails, the value is failed.

Example Request

- Example request 1: Cancel a submitted job.
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/jobs/{job_id}
- Example request 2: Cancel all jobs in a queue.
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/hyg/{graph_name}/jobs

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Graph [
{project_id}
-movie1] does not exist, please check project_id and graph_name.",
  "errorCode": "GES.8000",
```



```
"result": "failed"  
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.9 Native Algorithm APIs

7.2.9.1 Running Algorithms

Function

This API is used to run specified algorithms based on input parameters.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm

Table 7-555 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

For details about the parameters, see [Common algorithm parameters](#).

Response Parameters

Table 7-556 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
jobId	String	ID of the algorithm execution job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Example Request

Run a specified algorithm with the name **shortest_path**. The source vertex ID of the path is **145**, the target vertex ID is **10**, and the edges is directed.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-algorithm
{
  "algorithmName": "shortest_path",
  "parameters": {
    "source": "145",
    "target": "10",
    "directed": true
  },
  "executionMode": "sync"
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "4448c9fb-0b16-4a78-8d89-2a137c53454a001679122",
  "jobType": 1
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Graph [854d9316e71d4dc599c5f27ea323d445-ges_fdb1234] does not exist, please check projectId and graphName.",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.9.2 Algorithm API Parameter References

7.2.9.2.1 Common Algorithm Parameters

Request Parameters

Table 7-557 Request body parameter descriptions

Parameter	Mandatory	Type	Description
algorithmName	Yes	String	Algorithm name. The options are (algorithm names you can call): • shortest_path • shortest_path_of_vertex_sets • common_neighbors_of_vertex_sets
parameters	Yes	Object	Algorithm parameters. For details, see the parameter descriptions of each algorithm.

Table 7-558 New request body parameters in version 2.1.7

Parameter	Mandatory	Type	Description
executionMode	No	String	• sync: synchronous • async: asynchronous The default value is sync . Supported algorithms are as follows (algorithm names you can call): • shortest_path • shortest_path_of_vertex_sets
offset	No	Integer	Synchronization result offset. The default value is 0 . NOTE This parameter is available when executionMode is sync . Supported algorithms are as follows (algorithm names you can call): • shortest_path • shortest_path_of_vertex_sets • common_neighbors_of_vertex_sets
limit	No	Integer	Maximum number of returned synchronization results. The maximum value is 100000 . The default value is 100000 . NOTE This parameter is available when executionMode is sync . Supported algorithms are as follows (algorithm names you can call): • shortest_path • shortest_path_of_vertex_sets • common_neighbors_of_vertex_sets

Response Parameters

Table 7-559 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.
status	String	Job status returned for a successful query. The options are pending , running , and complete . This parameter is left blank when the query fails.
data	Object	Algorithm execution result. This parameter is left blank when the query fails.

Example Request

```
{
  "algorithmName": "XXX",
  "parameters": {
    ...
  }
}
```

Example Response

Algorithms are executed based on input parameters. You can call [Querying Job Status and Execution Results](#) to use the **job_id** returned by the algorithm to obtain the algorithm execution result.

Status code: 200

Example response for a successful request

```
{
  "data": {
    "outputs": {
      "path": [
        "46",
        "133",
        "138"
      ],
      "source": "46",
      "target": "138"
    }
  },
  "result": "success"
}
```

NOTE

response_data indicates the result of each algorithm. The results vary with algorithms.

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Unsupported value [shortest_path11] for parameter [algorithmName].",
  "errorCode": "GES.8402",
  "result": "failed"
}
```

7.2.9.2.2 Shortest Path

Table 7-560 parameters parameter description

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex ID of a path
target	Yes	String	Target vertex ID of a path
directed	No	Boolean	Whether to consider the edge direction. The value is true .
timeWindow	No	Object	Time window for time filtering. For details, see Table 7-561 . NOTE timeWindow does not support the shortest path with weight. That is, parameters timeWindow and weight cannot be both specified.

Table 7-561 timeWindow parameter description

Parameter	Mandatory	Type	Description
filterName	No	String	Character string: The property on the corresponding vertex/edge is used as the time.
filterType	No	String	Filters vertices or edges. The default value is BOTH . • V: filtering by vertex • E: filtering by edge • BOTH: filtering by vertex and edge
startTime	No	String	Start time, which is a string of the date type or a timestamp.
endTime	No	String	End time, which is a string of the date type or a timestamp.

Table 7-562 response_data parameter description

Parameter	Type	Description
path	List	Shortest path. The format is as follows: [vertexId,...] where vertexId is of the string type.

Parameter	Type	Description
source	String	Source vertex ID
target	String	Target vertex ID

7.2.9.2.3 Shortest Path of Vertex Set

Table 7-563 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
sources	Yes	Source vertex ID set	List	The value is in the standard CSV format. IDs are separated by commas (,), for example, ["Alice","Nana"] . The maximum ID number is 100000.	-
targets	Yes	Target vertex ID set	List	The value is in the standard CSV format. IDs are separated by commas (,), for example, ["Alice","Nana"] . The maximum ID number is 100000.	-
directed	No	Whether an edge is directed	Boolean	The value can only be true .	true
timeWindow	No	Time window used for time filtering	Object	For details, see Table 7-564 .	-

Table 7-564 timeWindow parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
filterName	No	Name of the time property used for time filtering	String	Character string: The property on the corresponding vertex/edge is used as the time.	-
filterType	No	Filtering by vertex or edge	String	V : filtering by vertex E : filtering by edge BOTH : filtering by vertex and edge	BOTH
startTime	No	Start time	String	Date character string or timestamp	-
endTime	No	End time	String	Date character string or timestamp	-

Table 7-565 response_data parameter description

Parameter	Type	Description
path	List	Shortest path. The format is as follows: [vertexId,...] where vertexId is of the string type.
source	String	Source vertex ID
target	String	Target vertex ID

7.2.9.2.4 Common Neighbors of Vertex Sets

Table 7-566 parameters parameter descriptions

Parameter	Mandatory	Description	Type	Value Range	Default Value
sources (2.2.6)	Yes	Source vertex ID set	List	The value is in the standard CSV format. IDs are separated by commas (,), for example, ["Alice","Nana"] . The maximum number of IDs is 100,000.	-
targets (2.2.6)	Yes	Target vertex ID set	List	The value is in the standard CSV format. IDs are separated by commas (,), for example, ["Mike","Amy"] . The maximum number of IDs is 100,000.	-

Parameter	Mandatory	Description	Type	Value Range	Default Value
restricted (2.2.13)	No	Whether other constraints are included	Boolean	true or false false: There is no additional constraint. The found common neighbors are the intersection of the neighborhoods corresponding to the source vertex set and target vertex set. true: There are additional constraints. The found common neighbors are not only the intersection of the neighborhoods corresponding to the source vertex set and target vertex set, but each vertex in the common neighbor set has at least two neighboring vertices in the source vertex set and target vertex set.	true

Table 7-567 response_data parameter descriptions

Parameter	Type	Description
vertices	List	Common neighbor vertices. The format is as follows: [vertexId,...], where vertexId is of the string type.
common_neighbors	Integer	Number of common neighbors

7.2.10 Graph Statistics APIs

7.2.10.1 Querying General Information About a Graph

Function

This API is used to query the general information about a graph, such as the numbers of vertices and edges.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/summary?
label_details={labelDetails}

Table 7-568 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
label_details	No	Boolean	Whether to return the number of vertices and edges under each label. The default value is false . If this parameter is set to true , the numbers of vertices and edges under each label are returned.

NOTE

- To return the count of vertices under different labels, when creating a vertex index, set the index type to **GlobalCompositeVertexIndex**, ensure that **hasLabel** is **true**, and keep the property list empty. Refer to [Creating an Index](#) for detailed instructions on how to create an index.
- To return the count of edges under different labels, when creating an edge index, set the index type to **GlobalCompositeEdgeIndex**, ensure that **hasLabel** is **true**, and keep the property list empty. Refer to [Creating an Index](#) for detailed instructions on how to create an index.

Response Parameters

Table 7-569 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job The API for querying the job ID is called to check the task execution status and obtain the returned result. For details, see Querying Job Status on the Service Plane (1.0.0) . For details about the calling result parameters, see Table 7-570 .

Table 7-570 data parameter descriptions

Parameter	Type	Description
vertexNum	Integer	Number of vertices in a graph. In multi-label scenarios, the value is repeatedly counted based on the number of label occurrences. To count the actual number of vertices, run the match (n) where n is not null return count(*) statement.
edgeNum	Integer	Number of edges in a graph
labelDetails	Object	Numbers of vertices and edges under each label. This parameter is available only when label_details in the URL is set to true .

Table 7-571 Description of each element in labelDetails when the execution is successful

Parameter	Type	Description
labelInVertex	Object	Numbers of vertices under each label.
labelInEdge	Object	Numbers of edges under each label.

Parameter	Type	Description
errorMessage	String	System prompt. When there is no vertex index or edge index, an error message is displayed. Otherwise, the field remains empty.
errorCode	String	System prompt. When there is no vertex index or edge index, an error code is displayed. Otherwise, the field remains empty.

Example Request

Query general information about a graph, such as the numbers of vertices and edges. The value **true** indicates that the numbers of vertices and edges of different labels are returned.

GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/summary?label_details=true

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.

Return Value	Description
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.10.2 Querying the Graph Version

Function

This API is used to query the graph version.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/version

Table 7-572 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-573 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
version	String	Query results. This parameter is left blank when the request fails.

Example Request

Query the graph version.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/version
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "version": "2.4.9",
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 404
{
  "errorMessage": "Not found. Please check the input parameters.",
  "errorCode": "GES.8000"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.

Return Value	Description
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.11 Graph Operation APIs

7.2.11.1 Importing a Graph

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-graph

Table 7-574 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-575 Request body parameter descriptions

Parameter	Mandatory	Type	Description
edgesetPath	No	String	Set edgesetPath , vertexsetPath , or schemaPath . Edge file directory or name
edgesetFormat	No	String	Edge dataset format. Currently, the options are csv and txt . The default value is csv .
vertexsetPath	No	String	Set edgesetPath , vertexsetPath , or schemaPath . Vertex file directory or name

Parameter	Mandatory	Type	Description
vertexsetFormat	No	String	Vertex dataset format. Currently, the options are csv and txt . The default value is csv .
schemaPath	No	String	Set edgesetPath , vertexsetPath , or schemaPath . OBS path of the metadata file of the new data
logDir	No	String	Directory for storing logs of imported graphs. This directory stores the data that fails to be imported during graph creation and detailed error causes.
parallelEdge	No	Object	How to process repetitive edges.
action	No	String	Way of processing repetitive edges. If the value of action is override , the previous repetitive edges are overwritten. If the value of action is ignore , repetitive edges are ignored. This means if an edge already exists, the edge will be ignored and will not be overwritten.
ignoreLabel	No	Boolean	Whether to ignore labels on repetitive edges. The value is false . false : Indicates that the repetitive edge definition contains the label. That is, the <source vertex, target vertex, label> indicates an edge.
delimiter	No	Character	Field separator in a CSV file. The default value is comma (,). The default element separator in a field of the list/set type is semicolon (;).
trimQuote	No	Character	Field quote character in a CSV file. The default value is double quotation marks ("). They are used to enclose a field if the field contains separators or line breaks.

Parameter	Mandatory	Type	Description
offline	No	Boolean	Whether to import a graph offline. The value is false . false : Online import is selected. Compared with offline import, online import is slower. However, the graph can be read (cannot be written) during the import.
obsParameters	Yes	Object	OBS parameters

Table 7-576 parallelEdge parameter type

Parameter	Mandatory	Type	Description
sortKeyColumn	No	Int/String	Index of the sort key column in the edge file. The value starts from 1. If this parameter is set to last Column , the sort key column is the last column. The options are: • A positive integer • A positive integer of the string type • "lastColumn"

Table 7-577 obsParameters parameters

Parameter	Mandatory	Type	Description
accessKey	Yes	String	User AK
secretKey	Yes	String	User SK

Response Parameters

Table 7-578 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Example Request

Import a graph. The edge file directory is **testbucket/demo_movie/edges/**, and the edge dataset format is CSV. The vertex file directory is **testbucket/demo_movie/vertices/**, and the vertex dataset format is CSV. The OBS path of the metadata file of the new data is **testbucket/demo_movie/incremental_data_schema.xml**, and the log storage directory is **testbucket/importlogdir**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=import-graph
{
  "edgesetPath": "testbucket/demo_movie/edges/",
  "edgesetFormat": "csv",
  "vertexsetPath": "testbucket/demo_movie/vertices/",
  "vertexsetFormat": "csv",
  "schemaPath": "testbucket/demo_movie/incremental_data_schema.xml",
  "logDir": "testbucket/importlogdir",
  "parallelEdge": {
    "action": "override",
    "ignoreLabel": false
  },
  "delimiter": ",",
  "trimQuote": "\"\"",
  "offline": false,
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  }
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.11.2 Clearing a Graph

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=clear-graph

Table 7-579 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Parameter	Mandatory	Type	Description
graph_name	Yes	String	Graph name

Request Parameters

Table 7-580 Request body parameter

Parameter	Mandatory	Type	Description
clearMetadata	No	Boolean	Whether to clear schema data. The default value is false .

Response Parameters

Table 7-581 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Example Request

Clear a graph by deleting its schema data.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=clear-graph
{
  "clearMetadata": true
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.11.3 Exporting a Graph

NOTE

If you choose to export CSV files to your local host, the files are opened using the spreadsheet software by default. You are advised to open the files in a text editor. If the data contains special characters such as plus signs (+), minus signs (-), equal signs (=), and at signs (@), the data will be parsed into formulas by the software. To ensure system security, pay attention to the following when opening such files:

1. Do not select **Enable Dynamic Data Exchange Server Launch (not recommended)**.
2. Do not select **Enable** or **Yes** if a dialog box indicating a security issue is displayed.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph

Table 7-582 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-583 Request body parameters

Parameter	Mandatory	Type	Description
graphExportPath	Yes	String	OBS path a graph is exported to
edgeSetName	Yes	String	Name of the exported edge data set
vertexSetName	Yes	String	Name of the exported vertex data set
schemaName	Yes	String	Name of the exported metadata file
paginate	No	Object	Pagination parameters. For details, see paginate parameter description .
obsParameters	Yes	String	OBS parameters. For details, see obsParameters parameter description .
accessKey	Yes	String	User AK
secretKey	Yes	String	User SK

Table 7-584 paginate parameter description

Parameter	Mandatory	Type	Description
rowCountPerFile	No	Integer	Maximum number of rows in each file when graphs are exported by page. The default value is 1000000 .

Parameter	Mandatory	Type	Description
numThread	No	Integer	Number of concurrent threads when graphs are exported by page. The default value is 8 . The maximum value cannot exceed the number of kernels.
maxSizePer File	No	Integer	Maximum size of each file when graphs are exported by page, in bytes. By default, the size cannot exceed the maximum size of the file that was imported.

Table 7-585 obsParameters parameter description

Parameter	Mandatory	Type	Description
accessKey	Yes	String	AK
secretKey	Yes	String	SK

Response Parameters

Table 7-586 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt code - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Example Request

Export a graph. The OBS path for exporting the graph is **demo_movie/**, the name of the exported edge data set is **set_edge**, the name of the exported vertex data set is **set_vertex**, and the name of the exported metadata file is **set_schema.xml**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph
{
  "graphExportPath": "demo_movie/",
  "edgeSetName": "set_edge",
  "vertexSetName": "set_vertex",
  "schemaName": "set_schema.xml",
  "paginate": {
    "numThread": 8,
    "rowCountPerFile": 1000000
  },
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  }
}
```

Example request 2

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=export-graph
{
  "graphExportPath": "demo_movie/",
  "edgeSetName": "set_edge",
  "vertexSetName": "set_vertex",
  "schemaName": "set_schema.xml",
  "obsParameters": {
    "accessKey": "xxxxxx",
    "secretKey": "xxxxxx"
  }
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.

Return Value	Description
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.11.4 Creating a Graph

URI

POST /ges/v1.0/{project_id}/graphs

Table 7-587 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Request Parameters

Table 7-588 Request body parameters

Parameter	Mandatory	Type	Description
graphName	Yes	String	Graph name. The value can contain 4 to 32 characters and must start with a letter. Only letters, digits, and underscores (_) are allowed.
idType	Yes	String	ID type. The options are hash , fixedLengthString , and varString .
idLength	No	Integer	This parameter is mandatory if idType is fixedLengthString . The value ranges from 1 to 128.

Parameter	Mandatory	Type	Description
sortKeyType	Yes	String	Mandatory. The options are: • int: integer. • string: The value is a string that contains a maximum of 40 characters. • varString: The value is a variable-length string with no length limit. The sort key type applies to the entire graph.

Response Parameters

Table 7-589 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.
result	String	Query result. If the query is successful, the value is success . If the query fails, the value is failed .

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs
{
  "graphName": "xxx",
  "idType": "fixedLengthString",
  "idLength": 20,
  "sortKeyType": "varString"
}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "graph [movie] already exists",
  "errorCode": "GES.8012",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.11.5 Deleting a Graph

URI

DELETE /ges/v1.0/{project_id}/graphs/{graph_name}

Table 7-590 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-591 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Example Request

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}
```

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "jobId": "3-f78ec641-ed66-4983-bf93-7f9b3a716c780000000019090"
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.11.6 Graph List

URI

GET /ges/v1.0/{project_id}/graphs

Table 7-592 URI parameter

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Request Parameters

None

Response Parameters

Table 7-593 Response body parameters

Parameter	Type	Description
graphs	Array of graphs objects	Returned graph list
result	String	Query result. The value is success for a successful query and failed for a failed query.

Table 7-594 graphs

Parameter	Type	Description
graphName	String	Graph name
idType	String	ID type
idLength	Integer	ID length
sortKeyType	String	sort key type
averageImportRate	Integer	Average import rate

Example Request

GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "graphs": [
    {
      "graphName": "movieee",
      "idType": "fixedLengthString",
      "idLength": 20,
      "sortKeyType": "varString",

      "averageImportRate": 0
    },
    {
      "graphName": "ldbc",
      "idType": "fixedLengthString",
      "idLength": 20,
      "sortKeyType": "int",

      "averageImportRate": 0
    }
  ],
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.12 Job Management APIs

7.2.12.1 Querying the Job List

Function

Query all asynchronous jobs saved in the engine and return the ID, status, and original request of each job.

NOTE

A maximum of 100 thousand records can be returned for database edition graphs.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status?
limit={limit}&offset={offset}

Table 7-595 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
limit	No	Integer	Maximum number of records that can be queried. The default value is 100000 .
offset	No	Integer	Offset of a query. The default value is 0 .

Request Parameters

None

Response Parameters

Table 7-596 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
result	String	Query result. The value is success for a successful query and failed for a failed query.
jobs	Object	Job status list stored in the system. If execution succeeds, this parameter is contained in the response. Table 7-597 describes the structure of a single jobs field.
totalCount	Integer	Total number of tasks to be queried

Table 7-597 Job status structure

Parameter	Type	Description
job_id	String	Job name.
request	Object	Request content, including the command, URL, and body.
status	String	Job status. The value can be pending , running , complete , success , or failed .

Example Request

Query the job list and return the job ID and status of each job.

GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/status

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "jobs": [
    {
      "jobId": "4ae64af4226f9b91a469c7e609e00b800000382489090",
      "status": "success",
      "request": {
        "command": "batch_build_index",
        "url": "/ges/v1.0/854d9316e71d4dc599c5f27ea323d445/graphs/ges_j60071198_hdfs_fdb/indices/
action?action_id=batch-build&targetGraphName=ges_j60071198_hdfs_fdb",
        "rawRequest": "{ \"indices\": [{ \"indexType\": \"GlobalCompositeEdgeIndex\", \"indexName
\": \"cypher_edge_index\", \"indexProperty\": [], \"hasLabel\": true }, { \"indexType
\": \"GlobalCompositeVertexIndex\", \"indexName\": \"cypher_vertex_index\", \"indexProperty\": [], \"hasLabel
\": true } ] }"
      },
    },
    {
      "jobId": "d33bdeef9923fc24e20a38b11dbb13540000382489090",
      "status": "success",
      "request": {
        "command": "import_graph",
        "url": "/ges/v1.0/854d9316e71d4dc599c5f27ea323d445/graphs/ges_j60071198_hdfs_fdb/action?
action_id=import-graph",
        "rawRequest": "{ \"graphName\": \"ges_j60071198_hdfs_fdb\", \"offline\": false, \"edgesetPath\": \"/
user/GES/audata/ranking_edge.csv\", \"vertexsetPath\": \"/user/GES/audata/movies_vertex_new.csv
\", \"delimiter\": \",\", \"trimQuote\": \"\", \"vertexsetFormat\": \"csv\", \"parallelEdge\": { \"ignoreLabel
\": false, \"action\": \"override\" }, \"edgesetFormat\": \"csv\" }"
      },
    },
    {
      "jobId": "3d29c4aac80382aa161214ca716bbf860000382489090",
      "status": "success",
      "request": {
        "command": "import_graph",
        "url": "/ges/v1.0/854d9316e71d4dc599c5f27ea323d445/graphs/ges_j60071198_hdfs_fdb/action?
action_id=import-graph",
        "rawRequest": "{ \"graphName\": \"ges_j60071198_hdfs_fdb\", \"offline\": false, \"edgesetPath
\": \"\", \"vertexsetPath\": \"\", \"delimiter\": \"\", \"schemaPath\": \"/user/GES/audata/schema_aikv.xml
\", \"trimQuote\": \"\", \"vertexsetFormat\": \"csv\", \"parallelEdge\": { \"ignoreLabel\": false, \"action
\": \"override\" }, \"edgesetFormat\": \"csv\" }"
      },
    },
  ],
  "totalCount": 3,
  "result": "success"
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "Parameter error!",
  "errorCode": "GES.8005"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.12.2 Querying the Job Status

Function

This API is used to query the execution status of a job. After asynchronous APIs such as those for querying vertices and edges or executing algorithms are used, job IDs are returned. You can use the job ID to query the execution status of each job.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?
offset=*offset*&limit=*limit*

Table 7-598 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
job_id	Yes	String	Job ID
offset	No	Integer	Offset of a query. The default value is 0 .
limit	No	Integer	Maximum number of records that can be queried. The default value is 100000 .

Response Parameters

Table 7-599 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
status	String	Job status when the query is successful. The options are: pending running success complete failed This parameter is left blank when the query fails.
data	Object	Algorithm execution result. This parameter is left blank when the query fails.
result	String	Query results. The value is success for a successful request and failed for a failed request.

Table 7-600 data parameter descriptions

Parameter	Type	Description
vertices	List	Vertex-associated algorithm result
edges	List	Edge-associated algorithm result
outputs	Object	Other results
data_return_size	Integer	Number of records returned after a query
data_offset	Integer	Result offset of a query

Parameter	Type	Description
data_total_size	Integer	Total amount of result data generated by asynchronous jobs

Example Request

Query the execution status of a job. The query offset is **0**, and the maximum number of returned results is **2**.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/jobs/{job_id}/status?offset=0&limit=2
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
Http Status Code: 200
{
  "status": "complete",
  "data": {
    "outputs": {
      "path": [
        "46",
        "133",
        "130",
        "78"
      ],
      "source": "46",
      "target": "78"
    }
  },
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "can not find job, jobId is 0ebb322076be429359b721c46fde16ff00003824890901",
  "errorCode": "GES.8301",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.

Return Value	Description
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.13 Cypher Operation APIs

7.2.13.1 Executing Cypher Queries

Function

Cypher is a widely used declarative graph database query language. It can be used to query data in GES and returns results. Graph statistics are used in Cypher implementation. Currently, the label-based vertex and edge indexes are used during Cypher query and compilation. To use Cypher normally, create indexes by referring to [Cypher Prerequisites](#).

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query

Table 7-601 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

Table 7-602 Request body parameter description

Parameter	Mandatory	Type	Description
statements	Yes	List	Statement group that contains one or more statements. Table 7-603 describes the format of each element.

Table 7-603 statements parameter descriptions

Parameter	Mandatory	Type	Description
statement	Yes	String	Cypher statement
parameters	Yes	Object	Cypher statement parameters, which are used for parameterized queries. By default, this field is left blank. For details, see parameterized queries .
resultDataContents	No	String or List	Format of the returned result. You can set one or more formats. The options are row , graph , and raw (new in version 2.2.27).
includeStats	No	Boolean	Whether the returned result contains addition, deletion, and modification statistics. If this parameter is not set, the returned result does not contain the information by default.
runtime	No	String	Executor type. The value can be map , slotted , or block . The default value is map . The slotted executor is supported since version 2.3.15, and the block executor is supported since version 2.4.1.
executionMode (2.2.23)	No	String	Execution mode. Set this parameter to sync for synchronous execution and to async for asynchronous execution. If this parameter is not set, the execution is synchronous by default. For details about how to obtain the query result in asynchronous mode, see Querying Job Status on the Service Plane .

Parameter	Mandatory	Type	Description
limit (2.2.23)	No	Int	Maximum number of asynchronous results. This parameter applies only when executionMode is async . The default value is 100000 .
transactional	No	Bool	Whether the Cypher request is transactional. The default value is false . For details about Cypher transactions, see Cypher transactions .

 NOTE

- You can add the **explain** or **profile** prefix before a statement to display the query plan. The **explain** prefix displays only the query plan but does not execute the statement. The **profile** prefix displays the query plan and executes the statement.
- Description of the **runtime** field: Compared with the map executor, the slotted executor and block executor complete more statement data flow analysis in the plan generation phase of statements. In most cases, they execute faster while requiring less memory.
- Cypher transaction:
 - For database edition graphs, Cypher transactions are supported. You can set **transactional** to **true** to enable the function to ensure the atomicity of a single Cypher statement. Transactions supporting multiple Cypher statements are not available. Transactions in GES use a serializability isolation level.
 - The transaction time window is limited to 5s in the underlying storage engine of GES. Cypher transactions cannot last exceeding 5s. For complex queries, such as multi-hop queries, the running time may exceed 5s. The transaction times out and the submission fails.

In this case, you can use the **dbms.killQuery** program of Cypher to terminate a Cypher transaction (for details, see [Cypher API-Functions and Procedures](#)) and restore all changes caused by the Cypher request.

Response Parameters

Table 7-604 Response body parameter descriptions

Parameter	Type	Description
results	List	Each element is the return result of a Cypher statement.
errors	List	Each element contains the code and message information in string form.

Table 7-605 Elements of the results parameter

Parameter	Type	Description
columns	List	Name of a returned field
data	List	Returned data value. Each element indicates a record.
stats	Object	Addition, deletion, and modification statistics
plan	Object	If the Cypher statement contains the explain or profile prefix, this field contains the query plan. Otherwise, this field is not displayed. The profile feature is supported since version 2.3.12.
jobId(2.3.10)	String	Asynchronous job ID if the request is executed asynchronously

Table 7-606 Elements of the data parameter

Parameter	Type	Description
row	List	Content of a specific row. Each element corresponds to a field in the row. This parameter is displayed only when resultDataContents is empty or contains row .
meta	List	Type of each field in a row. This parameter is displayed only when resultDataContents is empty or contains row .
graph	Object	Information returned in graph format. This parameter is displayed only when resultDataContents contains graph .
raw(2.2.27)	List	Information returned in raw format. This parameter is displayed only when resultDataContents contains raw .

Table 7-607 stats elements in a response

Parameter	Type	Description
contains_updates	Boolean	Whether data is modified during the query
edges_created	Integer	Number of created edges
edges_deleted	Integer	Number of deleted edges

Parameter	Type	Description
labels_set	Integer	Number of labels that have been set
properties_set	Integer	Number of properties that have been set
vertices_created	Integer	Number of created vertices
vertices_deleted	Integer	Number of deleted vertices

Example Request

Run the Cypher statement **match (n) return n limit 1**.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query
{
  "statements": [{
    "statement": "match (n) return n limit 1",
    "parameters": {},
    "resultDataContents": ["row"],
    "includeStats": false
  }]
}
```

Example Response

Status code: 200

Example response for a successful request (synchronous call)

```
Http Status Code: 200
{
  "results": [
    {
      "columns": ["n"],
      "data": [
        {
          "row": [
            {
              "occupation": "artist",
              "gender": "F",
              "Zip-code": "98133",
              "userid": 0,
              "age": "25-34"
            }
          ]
        }
      ],
      "meta": [
        {
          "id": "46",
          "type": "node",
          "labels": [
            "user"
          ]
        }
      ]
    }
  ],
  "stats": {
```

```
        "contains_updates": false,
        "edges_created": 0,
        "edges_deleted": 0,
        "labels_set": 0,
        "properties_set": 0,
        "vertices_created": 0,
        "vertices_deleted": 0
      }
    }
  ],
  "errors": []
}
```

Status code: 200

Example response for a successful request (asynchronous call)

```
Http Status Code: 200
{
  "results": [
    {
      "columns": [
        "jobId"
      ],
      "jobId": "cbb39b88f8613a164222dce0a3831d000000382489090",
      "data": [
        {
          "row": [
            "cbb39b88f8613a164222dce0a3831d000000382489090"
          ],
          "meta": [
            null
          ]
        }
      ],
      "stats": {}
    }
  ],
  "errors": []
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "results": [],
  "errors": [
    {
      "code": "GES.8902",
      "message": "Parentheses are required to identify nodes in patterns, i.e. (n) (line 1, column 7 (offset:
6))"
    }
  ]
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.

Return Value	Description
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes](#).

7.2.13.2 Basic Operations and Compatibility

Basic Operations

Operation	Cypher Statement
Querying vertices	match (n) return n
Querying edges	match (n)-[r]->(m) return n, r, m
Querying paths	match (n:user)-[r]->(m:movie)-->(s:series) return n,r,m,s
Querying vertices based on filter criteria	match(n:user) where n.userid>=5 return n
Grouping and aggregating	match(n:movie) return n.genres, count(*)
Deduplicating	match(n:movie) return distinct n.genres
Sorting	match(n:movie) return n order by n.movieid
Creating a vertex	create (n:user{userid:1}) return n
Creating an edge	match (n:user{userid:15}),(m:movie{movieid:10}) create (n)-[r:rate]->(m)
Deleting a vertex	match (n:user{userid:1}) delete n
Modifying labels	match (n:user{userid:1}) set n:movie return n
Modifying properties	match (n:user{userid:1}) set n.userid=2 return n

Compatibility to Cypher

1. Cypher clauses

Cypher implements a couple of clauses. You can combine clauses to implement different query semantics, including vertex and edge filtering, multi-hop query, sorting and deduplication, and grouping and aggregation. Currently, GES supports the Cypher clauses listed in the following table.

Table 7-608 Supported Cypher clauses

Clause	Support	Example
match	Partially supported	match (n:movie) return n
optional match	Partially supported	optional match (n)-->(m) where id(n)='1' return m
return	Supported	return [1,2,3] as p
with	Supported	match (n) with labels(n) as label, count(*) as count where count > 10 return *
where	Supported	match (n:movie) where n.movieid > 10 return n
order by	Supported	match (n:movie) return n order by n.genres
skip	Supported	match (n:movie) return n order by n.genres skip 5
limit	Supported	match (n:movie) return n order by n.genres skip 5 limit 10
create	Supported	create (n:user{_ID_: 'Jack' }) return n
delete	Supported	match (n:movie)-[r]-(m:user) delete r
set	Supported	match (n:user{userid:0}) set n.gender='M' return n
call procedures	Supported	call db.schema()
unwind	Supported	unwind [1, 2, 3] as p return p
union	Supported	match (n:movie) return id(n) union match (n:user) return id(n)

 NOTE

1. Currently, merge and foreach operations are not supported. Cypher statements cannot add or delete indexes.
 2. GES metadata is not schema-free, and the vertex and edge label properties are strictly restricted. Therefore, the remove operation is not supported.
 3. The order by clause does not support sorting of the list type. When Cardinality of the property value is not single, the sorting result is unknown.
 4. The CREATE statement does not support referencing the properties of the vertex to be created. For example, in the following statement, when the prop property of vertex b is set to **a.id**, it will not be assigned the value 0.
CREATE (a:End {prop: 42, id: 0}), (b:Begin {prop: a.id})
- Available items for the match clause

Item	Description	Example Clauses	Earliest Version Required
Vertex pattern	Patterns for matching vertex with specified labels, properties, and IDs.	match (n:movie{title:'hello'}) match (n) where id(n)='xx'	2.2.16
Edge pattern	Patterns for matching directional and non-directional edges with specified labels and properties. Specified IDs of both start and end vertices are supported.	match (n)-[r] -> (m) match (n)-[r]- (m) match (n)-[r:rate{Rating:1}] - (m) match (n)-[r]- (m) where id(n)='x'and id(m)='y'	2.2.16
Path	Anonymous paths	match (n)-[r]->(m)-->(s)	2.2.16
	Named paths	match p=(n)-[r]->(m)-->(s)	2.2.19
Multiple patterns	You can enter multiple patterns following match and separate them with commas (.). match (n)-[r]->(m), (m)-->(s)		2.2.16
Multi-match	You can enter multiple match clauses. You can use with to connect multiple clauses. match (n)-[r]->(m) with m match (m)-->(s)		2.2.16
Variable-length path pattern	Patterns for matching variable-length paths starting with a specified vertex. match p=(n)-[r*1..3]->(m) where id(n)='xx' return p match p=(n{title:'name'})-[r*1..3]->(m) return p		2.2.19
	Traversal conditions for matching variable-length paths. match p=(n)-[r*1..3]->(m) where id(n)='xx'and all (x in nodes(p) where x.prop='value1') return p		2.2.28
	Both start vertex and end vertex of a variable-length path can be specified. match p=(n)-[r*1..3]->(m) where id(n)='xx' and id(m)='y' return p		2.3.9

	Deduplication by end vertex is not supported: match p=(n)-[r*1..3]->(m) where id(m)='xx' return distinct m	No

2. Parameterized queries

Cypher supports parameterized queries. Numeric and string values in a query statement are extracted and converted to parameters for faster compilation, improving the query speed.

There are some examples of parameterized queries:

– Example 1

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?
action_id=execute-cypher-query
{
  "statements": [{
    "statement": " match (n:user) where n.occupation = $occupation return n",
    "parameters": {
      "occupation" : "artist"
    },
    "resultDataContents": ["row"]
  }]
}
```

– Example 2

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?
action_id=execute-cypher-query
{
  "statements": [{
    "statement": " match (n:user {`Zip-code`: '98133'}) set n = $props return n",
    "parameters": {
      "props": {
        "gender": "M",
        "age": "56+"
      }
    },
    "resultDataContents": ["row"]
  }]
}
```

NOTE

There are some scenarios where parameterized queries are not supported. The following syntax is not valid:

1. Using **\$param** to search by property key and value. For example, **match (n) where n.\$param = 'something'**
2. Using **\$code** for vertex and edge labels. For example, **match (n:user) set n:\$code**

3. Supported data types

Currently, GES supports 10 data types: char, char_array, float, double, Boolean, long, Integer, date, enum, and string. Both Boolean and numeric types are supported in the Cypher syntax. The following table lists the mapping between other types and Cypher data types.

Table 7-609 Mapping between data types of GES and Cypher

GES	Cypher	Description
char	String	-

GES	Cypher	Description
char_array	String	-
string	String	-
date	Temporal	Currently, it supports input and output based on GES dates.

Table 7-610 Special types supported by Cypher

Type	Supported	Example
Node	Yes	match (n) return n limit 10
Relationship	Yes	match (n)-[r]->(m) return r limit 10
List	Yes	return [1,2,3] as li
Map	Yes	match (n)-->(m) return {start:id(n), end:id(m)}
Path	Yes	match p=(n1)-[:friends*1..2]-(n2) return p limit 10
Point, Spatial	No	-

NOTE

For the special types listed above, only the **List** can be used to match multi-value properties in GES. Other types cannot be set as the value of a property on vertices or edges using the set statement.

4. Vertex ID compatibility

- Cypher does not provide the syntax for setting the ID when a vertex is added. In GES, however, an ID of the string type is required to uniquely identify a vertex. To use the Cypher syntax in GES, add **_ID_** to specify the ID of a vertex in the create statement. For example, the **create(n{_ID_:'123456'})** statement creates a vertex whose ID is 123456.
- If the ID is not specified, a random ID is generated for the vertex.

NOTE

The **_ID_** identifier is supported only in the create statement. The match and set clauses do not support the **_ID_** identifier. In the match clause, you can use the **id()** function to obtain the vertex ID.

7.2.13.3 Supported Expressions, Functions, and Procedures

Expression

Cypher queries support multiple expressions and can be used in combination to form various filter criteria. Currently, the following expressions are supported:

Operation Type	Expression	Example
Logical operations	and	match (n:user) where n.age='Under 18' and n.gender='F' return n
	or	match(n:user) where n.`Zip-code`='22181' or n.userid=6 return n
	not	match(n:movie) where not n.genres contains 'Drama' return n
Null value judgment	is null	match (n) where n.userid is null return n
	is not null	match (n) where n.userid is not null return n
Comparison calculation	>,>=,<,<=,=,<>	match(n:user) where n.userid>=5 return n
Arithmetic operators (2.3.10)	+, -, *, /, %, ^	return (1+3)%3
String comparisons	starts with	match(n:movie) where n.genres starts with 'Comedy' return n
	ends with	match(n:movie) where n.genres ends with 'Drama' return n
	contains	match(n:movie) where n.genres contains 'Drama' return n
List-related operation	in	match(n:student) where 'math' in n.courses return n
	[]	match(n:user) return n['userid'] with [1, 2, 3, 4] as list return list[0] with [1, 2, 3, 4] as list return list[0..1] match p=(n)-->(m) return [x in nodes(p) where x.gender='F' id(x)]
Date expressions (2.3.10)	.year, .month, .day, .hour, .minute, .second, .dayOfWeek	Year, month, and day of a specific date: with '2000-12-27 23:44:41' as strVal with datetime(strVal) as d2 return d2.year, d2.month, d2.day, d2.hour, d2.minute, d2.second, d2.dayOfWeek, d2.ordinalDay

 NOTE

The where clause in Cypher queries does not support regular expressions.

Function

Cypher supports the following functions for grouping, aggregation, and vertex and edge operations:

 NOTE

Cypher queries are case-insensitive when it comes to input functions, meaning they do not distinguish between uppercase and lowercase letters.

1. Aggregate

Function	Earliest Version Supported	Description	Example
count	2.2.17	Returns the total number of results.	match (n) return count(*) match (n) return count(n.userid)
collect	2.2.17	Collects results into a list.	match (n:movie) return n.genres, collect(n) as movieList
sum	2.3.3	Returns the sum of values.	unwind [1, 2.0, 3] as p return sum(p)
avg	2.3.3	Returns the average of values.	unwind [1, 2.0, 3] as p return avg(p)
min	2.3.3	Returns the minimum value.	unwind [1, 2.0, 3] as p return min(p)
max	2.3.3	Returns the maximum value.	unwind [1, 2.0, 3] as p return max(p)

2. Regular functions

Based on the types of input parameters, regular functions are classified into vertex and edge functions, path functions, list functions, and value functions.

Table 7-611 Vertex and edge functions

Function	Earliest Version Supported	Description	Example
id	2.2.16	Obtains the ID of a vertex.	match (n) return id(n)

Function	Earliest Version Supported	Description	Example
labels	2.2.16	Obtains labels of a vertex.	match (n) return labels(n)
type	2.2.16	Obtains the label of an edge.	match(n)-[r]->(m) return type(r)
degree	2.2.26	Obtains the degree of a vertex.	match (n) where id(n)='Vivian' return degree(n)
inDegree	2.2.26	Obtains the indegree of a vertex.	match (n) where id(n)='Vivian' return inDegree(n)
outDegree	2.2.26	Obtains the outdegree of a vertex.	match (n) where id(n)='Vivian' return outDegree(n)
startNode	2.3.10	Obtains the start vertex of an edge.	match (n)-[r]->(m) return startNode(r)
endNode	2.3.10	Obtains the end vertex of an edge.	match (n)-[r]->(m) return endNode(r)

Table 7-612 Path functions (2.2.19)

Function	Earliest Version Supported	Description	Example
length	2.2.19	Obtains the path length.	match p=(n)-[:friends*1..2]->(m) return length(p)

Table 7-613 List functions

Function	Earliest Version Supported	Description	Example
head	2.3.10	Obtains the first element of a list.	with [1,2,3,4] as list return head(list)
last	2.3.10	Obtains the last element of a list.	with [1,2,3,4] as list return last(list)
size	2.3.10	Obtains the list length.	with [1,2,3,4] as list return size(list)
range(2.3.10)	2.3.10	Generates a list.	return range(1,5), range(1,5,2)

Table 7-614 Value functions

Function	Earliest Version Supported	Description	Example
toString	2.2.21	Converts a value to a string.	match (n) where toString(labels(n)) contains 'movi' return n
toUpper	2.2.26	Converts a string into uppercase letters.	match (n:movie) return toUpper(n.title)
toLower	2.2.26	Converts a string into lowercase letters.	match (n:movie) return toLower(n.title)
toInteger	2.2.29	Converts a string to an int number.	with '123' as p return toInteger(p)
toLong	2.2.29	Converts a string to a long number.	with '123' as p return toLong(p)
toFloat	2.2.29	Converts a string to a float number.	with '123.4' as p return toFloat(p)

toDouble	2.2.29	Converts a string to a double number.	with '123.4' as p return toDouble(p)
toBoolean	2.2.29	Converts a string to a bool value.	with 'true' as p return toBoolean(p)
size	2.2.29	Obtains the string length.	with 'GES' as p return size(p)
subString	2.3.10	Truncates a part of a string.	return subString('abc', 1), subString('abcde', 1,2)
coalesce	2.3.10	Obtains the first non-null value of the parameters.	return coalesce(null, '123')
trim	2.3.11	Removes whitespace characters on both sides of a string.	return trim(' hello ')
lTrim	2.3.11	Removes whitespace characters on the left side of a string.	return lTrim(' hello')
rTrim	2.3.11	Removes whitespace characters on the right side of a string.	return rTrim('hello ')
reverse	2.3.11	Returns a string with the characters in reverse order.	return reverse('hello')
left	2.3.11	Obtains several characters from the left side of a string.	with 'hello' as p return left(p, 3)

right	2.3.11	Obtains several characters from the right side of a string.	with 'hello' as p return right(p, 3)
replace	2.3.11	Replaces a string.	with 'hello' as p return replace(p, 'll', 'o')
split	2.3.11	Splits a string.	with 'hello' as p return split(p, 'e')

Table 7-615 Mathematical functions

Function	Earliest Version Supported	Description	Example
floor	2.3.10	Rounds a number down to the nearest integer.	return floor(4.1)
ceil	2.3.10	Rounds a number up to the nearest integer.	return ceil(4.1)
round	2.3.14	Round	return round(3.4), round(3.5)
abs	2.3.14	Absolute value function	return abs(-3),abs(-3.5)
sin	2.3.14	Sine function	return sin(pi()/2)
cos	2.3.14	Cosine function	return cos(0),cos(pi()/2)
tan	2.3.14	Tangent function	return tan(pi()/4)
acos	2.3.14	Inverse cosine function	return acos(1)
asin	2.3.14	Inverse sine function	return asin(0)
atan	2.3.14	Inverse tangent function	return atan(1)
cot	2.3.14	Cotangent function	return cot(pi()/4)

Function	Earliest Version Supported	Description	Example
radians	2.3.14	Converts degree to radian.	return radians(180)
degrees	2.3.14	Converts radian to degree.	return degrees(pi())
pi	2.3.14	Returns the approximate value of Pi (π).	return pi()

Table 7-616 Date and time functions

Function	Earliest Version Supported	Description	Example
datetime(val)	2.3.10	Returns the time based on the timestamp.	return datetime(1688696395)
datetime()	2.3.14	Obtains the current time (valid only for read statements).	return datetime()
timestamp(val)	2.3.10	Returns the timestamp based on the time string.	return timestamp('2023-07-07 02:20:42')
timestamp()	2.3.14	Obtains the current timestamp (valid only for read statements).	return timestamp()
localDatetime	2.3.14	Converts a time or timestamp to a local time string.	return localDatetime(timestamp())

Table 7-617 Predicate functions

Function	Earliest Version Supported	Description	Example
all	2.2.19	If all elements meet the expression, true is returned.	all (x in p where x>1)
any	2.2.19	If any element meets the expression, true is returned.	any (x in p where x>1)
none	2.2.19	If all elements cannot meet the expression, true is returned.	none (x in p where x>1)
single	2.2.19	If only one element meets the expression, true is returned.	single (x in p where x>1)

Table 7-618 Algorithm expressions

Function	Earliest Version Supported	Description	Example
shortestPath	2.3.2	Returns the shortest path between two vertices.	The following statement returns the shortest path between the given vertices n and m . The direction is from m to n , and the edge label is rate : with n,m, shortestPath((n)<-[rate*]-(m)) as p return p
allShortestPaths	2.3.2	Returns all shortest paths between two vertices.	The following statement returns all shortest paths between the given vertices n and m : with n,m, allShortestPaths((n)-[*]-(m)) as p return p

 NOTE

- Aggregate functions, such as **sum()**, **avg()**, **max()**, and **min()**, are not available for database edition graphs. Mathematical functions, such as **sin()** and **cos()**, will be available in the future.
- Degree functions are not available for the database edition.
- Degree functions and path functions are not available for database edition graphs. Value operation functions, except **toString**, are not available for the database edition.

Procedure

Currently, GES supports the following procedures.

Procedure	Statement
Obtaining graph pattern information	call db.schema()
Obtaining vertex labels	call db.labels()
Querying the Cypher statements that are being executed	call dbms.listQueries()
Terminating a Cypher statement based on queryId	call dbms.killQuery('queryId')
Querying indexes	call db.indexes()
Full-text indexing for querying vertices that meet the search criteria	call db.index.fulltext.queryNodes()
Full-text indexing for querying edges that meet the search criteria	call db.index.fulltext.queryRelationships()
Merging nodes	call apoc.refactor.mergeNodes(nodeList, refactorConfig)

 NOTE

- Full-text indexes support six types of queries: prefix, wildcard, regexp, fuzzy, match, and combine. To use full-text indexes, you need to call the API for creating a full-text index.
- Function and procedure names are case sensitive and must be in lower camel case.
- Example of a full-text index query request**
POST `http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-cypher-query`

```
{
  "statements": [
    {
      "statement": "call db.index.fulltext.queryNodes('combine', {title:'1977'}) yield node, score
return node, score skip 1 limit 10",
      "resultDataContents": [
        "row"
      ],
      "parameters": {}
    }
  ]
}
```

```
}  
]  
}
```

- Parallel edge processing policy
When using Cypher to add edges, you can add duplicate edges. Duplicate edges are two edges with the same source vertex and target vertex.
- How to add an edge without a label
When you use a Cypher statement to add an edge, set the label of the edge to the default value `__DEFAULT__`. For example, `create ()-[r:__DEFAULT__]->() return r`.

Querying the Schema Structure Using Cypher

- Function
Query the generated schema structure.
- Query statement
 - Name: Schema structure query
 - Statement: **call db.schema()**
 - Note:
If you did not call the API for generating the schema structure, the returned schema file contains all labels.
If you have called the API for generating the schema structure, this API returns the labels as the vertices and the relationships between the labels as edges.

7.2.14 GQL APIs

7.2.14.1 Executing GQL Queries

Function

GQL is a declarative query language with its core syntax resembling Cypher. Use GQL to query data in GES and retrieve results. For detailed information on GQL queries, refer to [GQL Syntax](#).

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-gql-query

Table 7-619 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID, which is used for resource isolation. For how to obtain a project ID, see Obtaining a Project ID .

Parameter	Mandatory	Type	Description
graph_name	Yes	String	Graph name

Request Parameters

Table 7-620 Body parameters

Parameter	Mandatory	Type	Description
statements	Yes	List	Statement group that contains one or more statements.

Table 7-621 statements parameters

Parameter	Mandatory	Type	Description
statement	Yes	String	GQL statement
parameters	No	Json	GQL statement parameters, which are used for parameterized queries. This parameter is left blank by default. For details, see parameterized queries .
resultDataContents	No	String or List	Format of the execution results. You can set one or more formats. The options are row , graph , and raw .
includeStats	No	Boolean	Whether the returned result contains addition, deletion, and modification statistics. If this parameter is not set, the returned result does not contain the information by default.
runtime	No	String	Executor type. The default value is block .
executionMode	No	String	Execution mode. Set this parameter to sync for synchronous execution and to async for asynchronous execution. If this parameter is not set, the execution is synchronous by default. For how to obtain the query results in asynchronous mode, refer to section 5.10.1.

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of asynchronous results. This parameter applies only when executionMode is async . The default value is 100000 .

 NOTE

1. You can add the **explain** or **profile** prefix before a statement to display the query plan. The **explain** prefix displays only the query plan but does not execute the statement. The **profile** prefix displays the query plan and executes the statement.
2. In asynchronous mode (**executionMode** is **async**), GQL query results of the memory edition can be exported to CSV files. This edition currently allows returning the following value types: vertex and edge single-value properties, vertex and edge IDs, and group count results. Object types cannot be exported in the current version. They are treated as null values in CSV files. For more details, refer to [Exporting Job Execution Results to Files \(2.2.1\)](#).

Response Parameters

Table 7-622 Response parameter descriptions

Parameter	Type	Description
results	List	Each element is the return result of a GQL statement. For details about the parameters, see Table 7-623 .
errors	List	Each element contains the code and message information in string form.

Table 7-623 Elements of the results parameter

Parameter	Type	Description
columns	List	Name of a returned field
data	List	Returned data value. Each element indicates a record. For detailed parameter descriptions, refer to Table 7-624 .
stats	Json	Returned statistics about addition, deletion, and modification operations. For detailed descriptions, refer to Table 7-625 .

Parameter	Type	Description
plan	Json	If the GQL statement contains the explain or profile prefix, this field contains the query plan. Otherwise, this field is not displayed.
jobId	String	Asynchronous job ID if the request is executed asynchronously
jobType	Integer	Type of the asynchronous job if the request is executed asynchronously

Table 7-624 Elements of the data parameter

Parameter	Type	Description
row	List	Content of a specific row. Each element corresponds to a field in the row. This parameter is displayed only when resultDataContents is empty or contains row .
meta	List	Type of each field in a row. This parameter is displayed only when resultDataContents is empty or contains row .
graph	Json	Information returned in graph format. This parameter is displayed only when resultDataContents contains graph .
raw	List	Content of a specific row. Each element corresponds to a field in the row. This parameter is displayed only when resultDataContents contains raw .

Table 7-625 Elements of the stats parameter

Parameter	Type	Description
contains_updates	Boolean	Whether data is modified during the query
edges_created	Integer	Number of created edges
edges_deleted	Integer	Number of deleted edges
labels_set	Integer	Number of labels that have been set
properties_set	Integer	Number of properties that have been set

Parameter	Type	Description
vertices_created	Integer	Number of created vertices
vertices_deleted	Integer	Number of deleted vertices

Example Request

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/action?action_id=execute-gql-query
{
  "statements": [{
    "statement": "match (n) return n limit 1",
    "runtime": "block",
    "parameters": {},
    "resultDataContents": ["raw"]
  }],
  "includeStats": true
}
```

Example Response

- Example response for a successful synchronous request

Status code: 200

```
Http Status Code: 200
{
  "results": [
    {
      "columns": ["n"],
      "data": [
        {
          "row": [
            {
              "occupation": "artist",
              "gender": "F",
              "Zip-code": "98133",
              "userid": 0,
              "age": "25-34"
            }
          ]
        }
      ],
      "meta": [
        {
          "id": "46",
          "type": "node",
          "labels": [
            "user"
          ]
        }
      ]
    }
  ],
  "stats": {
    "contains_updates": false,
    "edges_created": 0,
    "edges_deleted": 0,
    "labels_set": 0,
    "properties_set": 0,
    "vertices_created": 0,
    "vertices_deleted": 0
  }
},
"errors": []
}
```

- Example response for a successful asynchronous request

Status code: 200

```
Http Status Code: 200
{
  "results": [
    {
      "columns": ["n"],
      "data": [
        {
          "row": [
            {
              "occupation": "artist",
              "gender": "F",
              "Zip-code": "98133",
              "userid": 0,
              "age": "25-34"
            }
          ]
        }
      ],
      "meta": [
        {
          "id": "46",
          "type": "node",
          "labels": [
            "user"
          ]
        }
      ]
    }
  ],
  "stats": {
    "contains_updates": false,
    "edges_created": 0,
    "edges_deleted": 0,
    "labels_set": 0,
    "properties_set": 0,
    "vertices_created": 0,
    "vertices_deleted": 0
  }
},
"errors": []
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 200
{
  "results": [],
  "errors": [
    {
      "code": "GES.8904",
      "message": "Label index in vertices is not found."
    }
  ]
}
```

Status Codes

Table 7-626 Return code for failed requests

Return Value	Description
400 Bad Request	Request error.

Return Value	Description
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

7.2.14.2 GQL Syntax

GQL is a standard query language designed for graph databases. For details about the GQL standard, see the [GQL document](#) released by ISO.

Supported GQL Clauses

GES supports the following GQL clauses:

Table 7-627 Supported GQL clauses

Keyword	Support	Example	Remarks
insert	Only supported by the database edition	insert (n:movie{_ID_:'1a', genres: 'Comedy Drama'}) return n	-
set	Only supported by the database edition	match (n) where element_id(n)='1a' set n.movieid=1 return n	-
remove	Only supported by the database edition	match (n) where element_id(n)='1a' remove n.movieid return n	-
delete	Only supported by the database edition	match (n) where element_id(n)='1a' delete n	-
match	Partially supported	match (n where element_id(n)='1a')-->(m) return count(*)	For details, see Pattern Matching .

Keyword	Support	Example	Remarks
optional match	Partially supported	match (n) where id(n)= '1a' optional match (n)-->(m) return count(*)	For details, see Pattern Matching .
let	Supported	let a=1, value b::long=2 return a,b	When type matching is not considered, it can be equivalently represented as the WITH semantics in Cypher.
for	Supported	for a in [1,2,3] return a	The offset/ordinality parameter is currently not supported. It can be equivalently represented as the UNWIND semantics in Cypher.
filter/ where	Supported	match (n) where element_id(n)='1a' return n	-
order by	Supported	match (n:movie) return n order by n.genres asc	-
limit/ offset	Supported	match (n:movie) return n skip 10 limit 10	-
union& union all	Supported	return 1 as a union return 2 as a	-
next yield	Supported	match (n:user) return n union all match (n:movie) return n next yield n as b match (b)-->(m) return m	Currently, the yield keyword must be included to declare variables used in later stages.
call	Partially supported	call db.indexes()	-

 NOTE

1. Currently, multi-graph joint query and the **USE GRAPH** semantics are not supported.
2. The **SELECT... FROM... GROUP BY... HAVING** clause is not supported.
3. The **at schema** operation is not supported.
4. Currently, only the **union** operation is supported for the result set. The set operation semantics such as **except**, **intersect**, and **otherwise** are not supported.
5. The keyword **call** can only be followed by a procedure name. Subqueries of the call style are not supported.

Pattern Matching

In GQL, pattern matching generally refers to Cypher. Patterns described in the following table that do not include the **match** keyword are graph patterns within the match clause.

Table 7-628 Pattern matching

Item	Description	Example	Supported Version
Vertex pattern	Indicates a group of vertices sharing the same feature.	(n:movie{title:'hello'})	2.4.15
		(n:movie where n.title='hello')	
		(n:movie user)	
Edge pattern	Supports edge patterns with direction, without direction, and with label/property filtering. Allows edge queries by specifying the IDs of both start and end vertices.	(n)-[r]-> (m), (n)<-[r]-(m), (n)-[r]-(m)	
		(n)-[r:rate{Rating:1}]- (m)	
		(n)-[r:rate where r.Rating=1]- (m)	
Path	Supports anonymous paths.	match (n)-[r]->(m)-->(s)	
	Supports named paths.	match p=(n)-[r]->(m)-->(s)	
	Variable-length paths	match p=trail (n)-[r]->{1,3}(m) where element_id(n)='46' return p	
	Deduplicated variable-length traversals	match p=trail (n)-[r]->{1,3}(m) where element_id(n)='46' return distinct m	

Item	Description	Example	Supported Version
	Nested paths	match (d:Stop) ((s:Stop)-[:NEXT]->(f:Stop) where s.time <> f.time){1,3} (a:Stop) RETURN d.departs AS departureTime, a.arrives AS arrivalTime	Not supported
Path mode	Supports walk, trail, simple, and acyclic path modes. See Path Mode for details.	match trail (n)-[r]-{1,3}(m), trail (n)--(m1) where element_id(n)='0' return count(*)	2.4.15
Other	Multiple patterns in the match clause	match (n)-[r]->(m), (m)-->(s) return count(*)	
	Multiple match statements	match (n)-[r]->(m) with m match (m)-->(s) return count(*)	

 NOTE

1. The **optional** keyword cannot be followed by a subquery enclosed in curly braces.
2. The **SHORTEST** and **GROUP** keywords cannot follow the **match** keyword.
3. The **keep** keyword is not currently supported.

Path Mode

In the **match** statement, you can input several graph patterns representing multiple paths, with each path separated by commas. GQL supports the following four path modes, which are configured as optional keywords before the graph pattern. If no keyword is configured, the default value **walk** is used.

Table 7-629 Path modes

Keyword	Description
walk	In the same path, both vertices and edges can be repeated.
trail	In the same path, edges cannot be duplicated.
simple	In the same path, intermediate vertices except the end vertex cannot be the same.
acyclic	In the same path, each vertex must be unique.

Supported Types

1. The mapping between basic data types in GQL and GES types is as follows:

Table 7-630 GQL basic types

GQL Type	GES Type	Remarks
bool	Boolean	-
string	string, char array, and enum	-
int32	Integer	-
int64	long	-
float	float	-
double	double	-
datetime	date	The smallest time unit supported in GES is second.

2. Additionally, there are some complex types in GQL, with their construction methods and GES support detailed as follows:

Table 7-631 GQL complex types

GQL Type	Support	Example
vertex/node	Supported	match (n) return n
edge/ relationship	Supported	match (n)-[r]->(m) return r
array/list	Supported	return [1,2,3] as li
record	Supported	match (n)-->(m) return {start:id(n), end:id(m)}
path	Supported	match p=(n1)-[:friends]-{1,2}(n2) return p limit 10

 NOTE

1. Among the complex types mentioned in [GQL complex types](#), only **List** can be used to correspond to multi-valued properties in GES. Other types cannot be assigned as values to properties on vertices or edges using the **SET** statement.
2. Currently, the **path** type can be created using the **match** statement, but connecting multiple paths or constructing a path from vertices and edges is not yet supported.
3. Display construction and assignment for graph and table objects are currently unsupported.
4. Support for the **Duration** type compliant with the GQL standard is also unavailable.

Supported Expressions and Functions

1. GQL supports the expressions and functions in [Supported Expressions, Functions, and Procedures](#). GQL also supports the following functions.

Table 7-632 Functions supported by GQL

Function	Corresponding Cypher Function	Description	Example
collect_list	collect	Aggregate function, which collects results into a list.	match (n:movie) return n.genres, collect_list(n) as movieList
element_id	id	Obtains the vertex or edge ID.	match (n) where element_id(n)= '0' return n
character_length	size	Obtains the length of a string.	return character_length('abc')
path_length	length	Obtains the path length.	match p=TRAIL (n)--{1,2}(m) where id(n)='0' return path_length(p)

2. The following expressions are supported:

Table 7-633 Expressions supported by GQL

Expression	Description	Example
	Supports concatenation between lists and between strings.	return 'abc' 'de', [1,2] [3,4]

3. Supports GQL-style value type predicate expressions.

Table 7-634 Judgment expressions supported by GQL

Category	Type Keyword	Example
Basic type determination	Boolean, string, int, long, float, double, datetime, node, edge, path, list, record	let a=1 return a is ::int
Nested type determination	list	let a=[1,2] return a is ::list<int>
Union type	any Note: any<int string long> indicates that a type is one of int, string, and long.	for a in [1,2,'a1'] return a is ::any<int string long>

 NOTE

Currently, mod and trigonometric functions are not supported.

Index Support

Currently, GQL supports vertex indexes with **hasLabel** set to **true** and supports single-value matching and multi-value matching. For details about how to create and edit indexes, refer to [Index Operation APIs](#).

Example:

Index Type	Label	Index Attribute	Remarks
GlobalComposite-VertexIndex	true	movieid	Single-attribute index
GlobalComposite-VertexIndex	true	movieid, title	Multi-attribute index

The indexes in the above table serve as an example. The following queries can utilize the indexes:

Type	Matching Mode	Example
Single-attribute index	Exact match - single value	match (n:movie) where n.movieid = 0 return n
	Exact match - multiple values	match (n:movie) where n.movieid in [0, 1, 2] return n
	Range match	match (n:movie) where n.movieid > 0 return n

Type	Matching Mode	Example
		match (n:movie) where n.movieid >= 0 return n
		match (n:movie) where n.movieid < 0 return n
		match (n:movie) where n.movieid <= 0 return n
Multi-value index	Exact match	match (n:movie) where n.movieid = 0 and n.title="" return n

When performing **EXPLAIN** or **PROFILE** on a query statement that utilizes an index, the `NodeIndexSeek` operator is printed, indicating that the index was used for the query.

7.2.14.3 GQL Compatibility

For details about the GQL standard, refer to the [GQL document](#) published by ISO. GES provides minimal support for GQL (GQL minimum conformance) as follows:

1. Supports features GG02 "Graph with a closed graph type" and GG20 "Explicit element type names".
2. Supports Unicode 13.
3. Supports basic data types, including: STRING, BOOLEAN, INT, and FLOAT.
4. Supports a range of features. See the [Supported Features](#) for details.

Supported Features

The following table lists the GQL features supported by GES:

Chapter	Section	Syntax/Feature	Associated ID	Example
9 Procedure specification	9.1	<next statement>	GQ 20	match (n:user) return n union all match (n:movie) return n next yield n as b match (b)-->(m) return m
13 Data-modifying statements (supported only by the database edition)	13.2	<insert statement>	-	insert (n:movie{_ID_: '1a', genres: 'Comedy Drama'}) return n
	13.3	<set statement>	GD0 2	match (n) where element_id(n) = '1a' set n.movieid=1 return n

Chapter	Section	Syntax/Feature	Associated ID	Example
	13.4	<remove statement>	GD02	match (n) where element_id(n) = '1a' remove n.movieid return n
	13.5	<delete statement>	-	match (n) where element_id(n) = '1a' delete n
14 Query statement	14.2	union	GQ03	return 1 as a union return 2 as a
	14.4	<simple match statement>	-	match (n) return n
		<optional match statement>	-	match (n) where id(n)= '1a' optional match (n)-->(m) return count(*)
	14.6	<filter statement>	GQ08	match (n) where element_id(n) = '1a' return n
	14.7	<let statement>	GQ09	let a=1, value b::long=2 return a,b
	14.8	<for statement>	GQ10	for a in [1,2,3] return a
	14.11	<return statement>	-	match (n) return n.name as name, n.age
15 Procedure calling	15.3	<named procedure call>	GP04	call db.schema()
16 Comment elements	16.4	<graph pattern>	G004	For details, see Pattern Matching .
	16.5	<insert graph pattern>	-	insert (n:movie{ _ID_: '1a', genres: 'Comedy Drama'}) return n
	16.7	<quantified path primary>	G036	match p=(n)-[r]->{1,3}(m) where element_id(n)= 'xx' return p
		Non-local element pattern predicates	G041	let a = 'Jaws (1975) ' match (n:movie where n.title=a) return n

Chapter	Section	Syntax/Feature	Associated ID	Example
		<full edge pattern>	G043	match (n),(m) where element_id(n)= '100' and element_id(m)= '120' optional match (n)-[r]->(m) with n,r,m optional match (n)-[r1]-(m) with n,r,r1,m optional match (n)<-[r2]-(m) return r,r1,r2
	16.8	<label expression>	-	match (n:movie) return count(*)
		<label disjunction>	-	match (n:movie user) return n limit 10
		<label negation>	-	match (n:!movie) return count(*)
		<parenthesized label expression>	-	match (n:(!movie&!default) user) return count(*)
	16.11	<fixed quantifier>	-	For details, see "Variable-length path" in Pattern Matching .
		<general quantifier>	-	match p=(n)-[r]->{,3}(m) where element_id(n)= 'xx' return p
	16.13	<where clause>	-	Refer to sections 14.4 and 14.6.
	16.16	<order by clause>	-	match (n:movie) return n order by n.genres asc
	16.18	<limit clause>	GQ13	match (n:movie) return n skip 10 limit 10
	16.19	<offset clause>	GQ12	
19 Predicate	19.3	<comparison predicate>	-	let a=1,b=2 return a=b,a<>b,a<b,a>b,a<=b,a>=b
	19.5	<null predicate>	-	let value a::long=1 return a is null
	19.9	<label predicate>	G111	match (n) where element_id(n)= '0' return n:movie

Chapter	Section	Syntax/Feature	Associated ID	Example
20 Value expressions and specifications	20.4	<dynamic parameter specification>	-	See Basic Operations and Compatibility and 2.
	20.7	<simple case>	-	let a =2 return case a when 1 then a+1 else a+2 end
		<searched case>	-	let a =2 return case when a=1 then a+1 else a+2 end
	20.9	Horizontal Aggregation	GE09	match (n) return labels(n),count(distinct n),avg(n.age),max(n.age),min(n.age),sum(n.age)
	20.10	<element_id function>	G100	match (n) where element_id(n) = '1a' return n
	20.11	<property reference>	-	match (n) where element_id(n) = '1a' return n.movieid
		record property	-	return {name: 'Jay',age:1}.name
	20.15	<list concatenation>	-	return [1,2,3,4] [5,6,7,8]
	20.17	<list value constructor>	-	return [1,2,3,4]
	20.18	<record constructor>	GV45	return {name: 'Jay', age:10, details:{salary:1}}
	20.20	<boolean value expression>	-	return (true and false) or false
		Boolean Xor	GE07	return true xor false
	20.21	<numeric value expression>	-	return 1+2,1-2,1*2,1/2,-2
	20.22	<char length expression>	-	let value s::STRING = 'hello_world' return character_length(s)
		<path length expression>	GF04	match p=TRAIL (n)--{1,2}(m) where id(n)= '0' return path_length(p)

Chapter	Section	Syntax/Feature	Associated ID	Example
		size function	GF13	let list = [1,2,3,4] return size(list)

Supported GQL Optional Features

For optional GQL features, the GQL standard provides feature IDs and names. Currently, GES supports the following features:

Feature ID	Feature Name
G004	Path variables
G011	Advanced path modes: TRAIL
G036	Quantified edges
G041	Non-local element pattern predicates
G043	Complete full edge patterns
G060	Bounded graph pattern quantifiers
G100	ELEMENT_ID function
G111	IS LABELED predicate
GD01	Updatable graphs
GD02	Graph label set changes
GE07	Boolean XOR
GE09	Horizontal aggregation
GF01	Enhanced numeric functions
GF02	Trigonometric functions
GF11	Advanced aggregate functions: binary set functions
GF13	SIZE function
GG02	Graph with a closed graph type
GG20	Explicit element type names
GP04	Named procedure calls
GQ03	Composite query: UNION
GQ08	Filter statement

GQ09	LET statement
GQ10	FOR statement: list value support
GQ12	ORDER BY and page statement: OFFSET clause
GQ13	ORDER BY and page statement: LIMIT clause
GQ20	Advanced linear composition with NEXT
GV45	Record types
GV47	Open record type
GV48	Nested record types
GV50	List value types
GV55	Path value types

7.2.15 Interactive Transaction APIs

7.2.15.1 Creating a Transaction

Function

This API is used to create a transaction.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/transaction

Table 7-635 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-636 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
result	String	Transaction creation result. The value is success for a successful request and failed for a failed request.
commit	String	ID of the created transaction.

Example Request

Create a transaction.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/transaction
{}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
{
  "commit": "140507984406272-2fa8a507-f13b-4f23-8dde-4693db982b900000000019090",
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
{
  "errorMessage": "Graph [11-moie] does not exist, please check projectId and graphName.",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.15.2 Executing Transaction Cypher Statements

Function

This API is used to execute transaction Cypher statements.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/transaction/{commit}

Table 7-637 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
commit	Yes	String	Transaction ID, which is obtained by calling the transaction creation API.

Request Parameters

Table 7-638 Request body parameter description

Parameter	Mandatory	Type	Description
statements	Yes	List	Statement group that contains one or more statements. Table 7-639 describes the format of each element.

Table 7-639 statements parameter descriptions

Parameter	Mandatory	Type	Description
statement	Yes	String	Cypher statement
parameters	Yes	Object	Cypher statement parameters, which are used for parameterized queries. This parameter is left blank by default. For details, see parameterized queries .
resultDataContents	No	String or List	Format of the execution results. You can set one or more formats. The options are row , graph , and raw (new in version 2.2.27).
includeStats	No	Boolean	Whether the returned result contains addition, deletion, and modification statistics. If this parameter is not set, the returned result does not contain the information by default.
runtime	No	String	Executor type. The value can be map , slotted , or block . The default value is map . The slotted executor is supported since version 2.3.15, and the block executor is supported since version 2.4.1.

Parameter	Mandatory	Type	Description
executionMode (2.2.23)	No	String	Execution mode. Set this parameter to sync for synchronous execution and to async for asynchronous execution. If this parameter is not set, the execution is synchronous by default. For how to obtain the query results in asynchronous mode, refer to Querying Job Status on the Service Plane (1.0.0) .
limit (2.2.23)	No	Int	Maximum number of asynchronous results. This parameter applies only when executionMode is async . The default value is 100000 .

 NOTE

- You can add the **explain** or **profile** prefix before a statement to display the query plan. The **explain** prefix displays only the query plan but does not execute the statement. The **profile** prefix displays the query plan and executes the statement.
- Description of the **runtime** field: Compared with the map executor, the slotted executor and block executor complete more statement data flow analysis in the plan generation phase of statements. In most cases, they execute faster while requiring less memory.
- Cypher transaction:
 - For database edition graphs, Cypher transactions are supported. You can set **transactional** to **true** to ensure the atomicity of a single Cypher statement. Transactions supporting multiple Cypher statements are not supported. Transactions in GES use a serializability isolation level.
 - Since the underlying storage engine has a transaction time window limit of 5 seconds, the transactions for Cypher statements cannot exceed 5 seconds. For complex queries, such as those with multiple hops, the execution time may exceed 5 seconds, resulting in timeouts and failed submissions.

In this case, you can use the **dbms.killQuery** program of Cypher to terminate a Cypher transaction (see [Supported Expressions, Functions, and Procedures](#)) and roll back all changes caused by this Cypher request.

Response Parameters

Table 7-640 Response body parameter descriptions

Parameter	Type	Description
results	List	Each element of the list is the return result of a Cypher statement.
errors	List	Each element in the list contains the code and message information in string form.

Table 7-641 Elements of the results parameter

Parameter	Type	Description
columns	List	Name of a returned field
data	List	Returned data value. Each element indicates a record.
stats	Object	Addition, deletion, and modification statistics
plan	Object	If the Cypher statement contains the explain or profile prefix, this field contains the query plan. Otherwise, this field is not displayed. The profile feature is supported since version 2.3.12.
jobId(2.3.10)	String	Asynchronous job ID if the request is executed asynchronously

Table 7-642 Elements of the data parameter

Parameter	Type	Description
row	List	Content of a specific row. Each element corresponds to a field in the row. This parameter is displayed only when resultDataContents is empty or contains row .
meta	List	Type of each field in a row. This parameter is displayed only when resultDataContents is empty or contains row .
graph	Object	Information returned in graph format. This parameter is displayed only when resultDataContents contains graph .
raw(2.2.27)	List	Information returned in raw format. This parameter is displayed only when resultDataContents contains raw .

Table 7-643 stats elements in a response

Parameter	Type	Description
contains_updates	Boolean	Whether data is modified during the query
edges_created	Integer	Number of created edges
edges_deleted	Int	Number of deleted edges

Parameter	Type	Description
labels_set	Integer	Number of labels that have been set
properties_set	Integer	Number of properties that have been set
vertices_created	Integer	Number of created vertices
vertices_deleted	Integer	Number of deleted vertices

Example Request

Execute a Cypher query. The Cypher statement is **match (n) return n limit 1**. The returned results are in the format that each element corresponds to a field in the row.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/transaction/{commit}
{
  "statements": [{
    "statement": "match (n) return n limit 1",
    "parameters": {},
    "resultDataContents": ["row"],
    "includeStats": false
  }]
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request (synchronous call)

```
Http Status Code: 200
{
  "results": [
    {
      "columns": ["n"],
      "data": [
        {
          "row": [
            {
              "occupation": "artist",
              "gender": "F",
              "Zip-code": "98133",
              "userid": 0,
              "age": "25-34"
            }
          ]
        }
      ],
      "meta": [
        {
          "id": "46",
          "type": "node",
          "labels": [
```

```
        "user"
      ]
    }
  ]
},
"stats": {
  "contains_updates": false,
  "edges_created": 0,
  "edges_deleted": 0,
  "labels_set": 0,
  "properties_set": 0,
  "vertices_created": 0,
  "vertices_deleted": 0
}
},
"errors": []
}
```

Status code: 200

Example response for a successful request (asynchronous call)

```
Http Status Code: 200
"results": [
  {
    "columns": [
      "jobId"
    ],
    "jobId": "247f395d813e53804b022df6306ca4160000382489090",
    "data": [
      {
        "row": [
          "247f395d813e53804b022df6306ca4160000382489090"
        ],
        "meta": [
          null
        ]
      }
    ]
  }
],
"errors": []
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
"results": [],
"errors": [
  {
    "code": "GES.8902",
    "message": "Parentheses are required to identify nodes in patterns, i.e. (n) (line 1, column 7 (offset:
6))"
  }
]
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.

Return Value	Description
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.15.3 Committing a Transaction

Function

This API is used to commit a transaction.

NOTE

If a transaction Cypher statement is not committed within 5 seconds after it is created, it will automatically roll back.

URI

POST /ges/v1.0/{project_id}/graphs/{graph_name}/transaction/{commit}/commit

Table 7-644 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
commit	Yes	String	Transaction ID, which is obtained by calling the transaction creation API.

Request Parameters

None

Response Parameters

Table 7-645 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
result	String	Transaction creation result. The value is success for a successful request and failed for a failed request.

Example Request

This API is used to commit a transaction.

```
POST http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/transaction/{commit}/commit
{}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
{
  "errorMessage": "Graph [11-moie] does not exist, please check projectId and graphName.",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.15.4 Rolling Back a Transaction

Function

This API is used to roll back a transaction.

URI

DELETE /ges/v1.0/{project_id}/graphs/{graph_name}/transaction/{commit}

Table 7-646 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name
commit	Yes	String	Transaction ID, which is obtained by calling the transaction creation API.

Request Parameters

None

Response Parameters

Table 7-647 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
result	String	Transaction creation result. The value is success for a successful request and failed for a failed request.

Example Request

Delete a transaction.

```
DELETE http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/transaction/{commit}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

Example Response

Status code: 200

Example response for a successful request

```
{
  "result": "success"
}
```

Status code: 400

Example response for a failed request

```
{
  "errorMessage": "Graph [11-moie] does not exist, please check projectId and graphName.",
  "errorCode": "GES.8000",
  "result": "failed"
}
```

Status Codes

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.16 O&M Monitoring APIs

7.2.16.1 Viewing Monitoring Metrics

Function

This API is used to view monitoring metrics, including node metrics and graph instance performance monitoring metrics.

URI

GET /ges/v1.0/{project_id}/graphs/{graph_name}/om/metrics?
real_time=&with_performance_metrics=true

Table 7-648 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Table 7-649 Query parameters

Parameter	Mandatory	Type	Description
real_time	No	Boolean	Whether to query real-time monitoring metrics. The value can be true or false . The default value is false . false: Metrics for a graph instance within 2 minutes are queried. true: Real-time monitoring metrics are queried, and the query is responded to in 3 to 5 seconds.
with_performance_metrics	No	Boolean	Whether to query performance metrics. The graph instance performance metrics and metrics of each node are returned. The value can be true or false . The default value is true . If set to false , only node metrics are returned, and the response time is reduced by 1 to 2 seconds.

Request Parameters

None

Response Parameters

Table 7-650 Response body parameter descriptions

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
project_id	String	Project ID
id	String	Graph ID
name	String	Graph name

Parameter	Type	Description
timestamp	Long	Current timestamp
node_metrics	Array of objects	Node metrics
performance_metrics	Object	Performance metrics

Table 7-651 node_metrics parameter descriptions

Parameter	Type	Description
overview	Object	Node overview
disk_details	Array of objects	Disk details of the node
network_details	Array of objects	Network details of the node

Table 7-652 overview parameter descriptions

Parameter	Type	Description
ges_instance_name	String	Node name
instance_id	String	Node ID
work_ip	String	Node IP address
role	String	Node role
cpu_usage	Double	CPU usage
cpu_usage_usr	Double	CPU usage in user mode
cpu_usage_sys	Double	CPU usage in kernel mode
cpu_iowait	Double	CPU I/O wait rate
cpu_idle	Double	CPU idle rate
mem_total	Double	Total memory size, in GB
mem_usage	Double	Used memory size, in GB
mem_free	Double	Available memory size, in GB
mem_cached	Double	Memory cache size, in GB
mem_buffer	Double	Memory buffer size, in GB

Parameter	Type	Description
disk_total	Double	Total disk space, in GB
disk_usage_avg	Double	Average disk usage
disk_used	Double	Used disk space, in GB
disk_available	Double	Total available disk space, in GB
disk_io_read	Double	Disk read rate of a node, in KB/s
disk_io_write	Double	Disk read rate of a node, in KB/s
disk_io	Double	Total read and write speed of a node's disk
swap_total	Double	Total swap disk space of a node
swap_free	Double	Remaining swap disk space of a node
network_io_rate	Double	Network I/O rate of a node, in KB/s
host_stat	String	Node status
host_name	String	Node name

Table 7-653 disk_details parameter descriptions

Parameter	Type	Description
disk_name	String	Disk name
disk_type	String	Disk type
total	Double	Total disk space
available	Double	Available disk space
used	Double	Used disk space
used_percentage	Double	Disk usage
svctm	Long	Disk I/O service time, in milliseconds
await	Double	Disk I/O wait time, in milliseconds
util	Double	Disk I/O usage
write_rate	Double	Disk read rate
read_rate	Double	Disk write rate

Table 7-654 network_details parameter descriptions

Parameter	Type	Description
status	Integer	NIC status
interface_name	String	NIC name
packets_recv	Long	Number of received packets
packets_send	Long	Number of sent packets
packets_drop	Long	Number of lost packets
send_rate	Double	Sending rate, in KB/s
recv_rate	Double	Receiving rate, in KB/s

Table 7-655 performance_metrics parameter descriptions

Parameter	Type	Description
cpu_usage	Double	CPU usage
memory_usage	Double	Memory usage
disk_usage	Double	Average disk usage
disk_io_rate	Double	Disk I/O rate
network_io_rate	Double	Network I/O rate
tomcat_connections_usage	Double	Tomcat connection usage
qps	Long	Number of requests per second
storage_usage	Double	Graph storage usage
graph_name	String	Graph instance name
jvm_heap_usage	Double	Java heap memory usage

Example Request

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/om/metrics?real_time=true
```

Example Response

Status code: 200
Example response for a successful request

```
{
  "node_metrics": [
    {
```

```
"overview": {
  "network_io_rate": 8.51,
  "role": "ges-dispatch",
  "disk_io_write": 84.74,
  "mem_cached": 1.99,
  "cpu_usage_usr": 2.4,
  "cpu_usage_sys": 0.72,
  "disk_io_read": 3.99,
  "ges_instance_name": "ges_fdb_old-ges-dispatch-cn-1-1",
  "disk_used": 5.75,
  "swap_total": 0.0,
  "mem_buffer": 251.93,
  "disk_available": 244.15,
  "cpu_iowait": 0.04,
  "cpu_idle": 96.83,
  "disk_io": 88.73,
  "mem_total": 31.19,
  "instance_id": "e8d67304-6107-4935-9fe3-6cabf3d32adc",
  "mem_usage": 9.69,
  "disk_total": 249.9,
  "host_stat": "200",
  "mem_free": 25.94,
  "swap_free": 0.0,
  "cpu_usage": 3.12,
  "disk_usage_avg": 2.3,
  "work_ip": "172.16.29.231",
  "host_name": "ges_fdb_old-ges-dispatch-cn-1-1"
},
"disk_details": [
  {
    "svctm": 0.0,
    "total": 50.0,
    "util": 0.07,
    "write_rate": 65.59,
    "disk_name": "vda",
    "disk_type": "system",
    "used_percentage": 5.54,
    "available": 47.23,
    "await": 15.7,
    "read_rate": 3.84,
    "used": 2.77
  },
  {
    "svctm": 0.0,
    "total": 99.95,
    "util": 0.09,
    "write_rate": 18.14,
    "disk_name": "vdb",
    "disk_type": "data",
    "used_percentage": 2.79,
    "available": 97.17,
    "await": 40.31,
    "read_rate": 0.06,
    "used": 2.78
  },
  {
    "svctm": 0.0,
    "total": 99.95,
    "util": 0.11,
    "write_rate": 1.01,
    "disk_name": "vdc",
    "disk_type": "log",
    "used_percentage": 0.19,
    "available": 99.76,
    "await": 2.14,
    "read_rate": 0.09,
    "used": 0.19
  }
],
```

```
    "network_details": [
      {
        "send_rate": 0.08,
        "packets_drop": 0,
        "packets_rcv": 698597,
        "packets_send": 108397,
        "interface_name": "eth0",
        "rcv_rate": 0.06,
        "status": 1
      },
      {
        "send_rate": 0.0,
        "packets_drop": 0,
        "packets_rcv": 25766,
        "packets_send": 21486,
        "interface_name": "eth1",
        "rcv_rate": 0.0,
        "status": 1
      },
      {
        "send_rate": 5.33,
        "packets_drop": 0,
        "packets_rcv": 2012566,
        "packets_send": 3091470,
        "interface_name": "eth2",
        "rcv_rate": 3.04,
        "status": 1
      }
    ]
  },
  "performance_metrics": {
    "tomcat_connections_usage": 0.14,
    "network_io_rate": 10847.12,
    "graph_name": "ges_fdb_old",
    "disk_io_rate": 115.34,
    "qps": 0,
    "storage_usage": 0.63,
    "disk_usage": 2.98,
    "memory_usage": 29.92,
    "cpu_usage": 5.75,
    "jvm_heap_usage": 68.31
  },
  "project_id": "3571fe9ff5dc415d99adef84b6488e3b",
  "name": "ges_fdb_old",
  "id": "3518bb16-74b3-4259-a1d0-f38c2836cd11",
  "timestamp": 1733880320828
}
```

Status code: 400

Example response for a failed request

```
Http Status Code: 400
{
  "errorMessage": "graph ges_fdb is not exist.",
  "errorCode": "GES.8000"
}
```

Error Codes

See [Error Codes for Service Plane APIs](#).

7.2.16.2 Viewing Real-Time Requests

Function

This API is used to view the real-time requests on the current primary node.

NOTE

Only running Cypher queries are currently returned.

URI

GET

/ges/v1.0/{project_id}/graphs/{graph_name}/om/real-time-queries

Table 7-656 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request Parameters

None

Response Parameters

Table 7-657 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
instance_name	String	Instance name

Parameter	Type	Description
current_queries	List	Details of the current query list

Table 7-658 CurrentQuery parameter descriptions

Parameter	Type	Description
task_name	String	Task name
request_id	String	Request ID
status	String	Request execution status
request	String	Request parameter
running_duration	Double	Request execution duration, in seconds
begin_time	String	Request start time

Example Request

View real-time requests.

```
GET http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/om/real-time-queries
```

Example Response

Status code: 200

Example response for a successful request

Http Status Code: 200

```
{
  "current_queries": [
    {
      "begin_time": "2024-12-11 09:24:33",
      "request": "profile match (n)-[r:rate*1..2]->(m) return r limit 5",
      "request_id": "1-346e5188-4b20-4175-b2ef-f7bcbff2d730",
      "running_duration": 29.765045,
      "status": "RUNNING",
      "task_name": "cypher_plan_query"
    },
    {
      "begin_time": "2024-12-11 09:24:33",
      "request": "profile match (n)-[r:rate*1..2]->(m) return r limit 5",
      "request_id": "1-b6da5a44-d51e-4f3a-8538-907f84a37abb",
      "running_duration": 29.81796,
      "status": "RUNNING",
      "task_name": "cypher_plan_query"
    },
    {
      "begin_time": "2024-12-11 09:24:31",
      "request": "profile match (n)-[r:rate*1..2]->(m) return r limit 5",
      "request_id": "1-4ae2b9a3-b415-4f51-a751-8f93ce0979ab",
      "running_duration": 32.004372,
      "status": "RUNNING",
      "task_name": "cypher_plan_query"
    }
  ]
}
```



```
    },
    {
      "begin_time": "2024-12-11 09:24:31",
      "request": "profile match (n)-[r:rate*1..2]->(m) return r limit 5",
      "request_id": "1-18d27ac4-8777-44c0-9e7e-781a4897821b",
      "running_duration": 32.054717,
      "status": "RUNNING",
      "task_name": "cypher_plan_query"
    },
    {
      "begin_time": "2024-12-11 09:24:30",
      "request": "profile match (n)-[r:rate*1..2]->(m) return r limit 5",
      "request_id": "1-7a48582c-a79e-404f-9bea-f13407a22773",
      "running_duration": 33.097715,
      "status": "RUNNING",
      "task_name": "cypher_plan_query"
    },
    {
      "begin_time": "2024-12-11 09:24:24",
      "request": "profile match (n)-[r:rate*1..2]->(m) return r limit 5",
      "request_id": "1-d16660aa-a1fa-435c-a58a-1364b9a2b2ba",
      "running_duration": 38.848532,
      "status": "RUNNING",
      "task_name": "cypher_plan_query"
    }
  ],
  "instance_name": "ges_fdb_old-ges-dispatch-cn-2-1"
}
```

Status code: 400

Example response for a failed request

Http Status Code: 400

```
{
  "errorMessage": "graph ges_fdb is not exist.",
  "errorCode": "GES.8000"
}
```

Error Codes

See [Error Codes for Service Plane APIs](#).

8 Application Examples

8.1 Analyzing Graphs Using HyG Algorithms

GES offers a range of fundamental graph algorithms, graph analysis algorithms, and graph metric algorithms. These algorithms can be used for analyzing relationships within a graph.

Prerequisites

The **HyG computing engine** slider was toggled on when creating a graph of the database edition.

Figure 8-1 HyG computing engine



Procedure

1. Create a HyG graph.
 - a. Send **POST** `/ges/v1.0/{project_id}/hyg/{graph_name}`. *project_id* indicates the project ID, and *graph_name* indicates the name of a graph created in the graph database.
 - b. Add **X-Auth-Token** to the request header.
 - c. Specify the following parameters in the request body:

```
{  "inEdge": true // Whether the graph contains the incoming edge}
```
 - d. Check the response. The request is successful if the following response is displayed:

```
{  "result": "success"}
```

If the request is abnormal, locate the fault by referring to [Error Codes for Service Plane APIs](#).

2. Synchronize data from the graph database to the HyG computing engine.
 - a. Send **POST** `/ges/v1.0/{project_id}/hyg/{graph_name}/sync`. *project_id* indicates the project ID, and *graph_name* indicates the graph name.
 - b. Add **X-Auth-Token** to the request header.
 - c. Specify the following parameters in the request body:

```
{
  "vertex": [], // Vertex property list

  "edge": [
    {
      "property": [
        "Rating"
      ],
      "label": "rate"
    }
  ] // Edge property list
}
```

- d. Check the response based on the job ID. The request is successful if the following response is displayed:

```
{
  "status": "complete",
  "result": "success"
}
```

If the request is abnormal, locate the fault by referring to [Error Codes for Service Plane APIs](#).

With this API, you can synchronize any changes made to the graph database, whether it be adding, deleting, or modifying data, to the HyG computing engine. During the initial data synchronization, the **vertex** and **edge** parameters specified in the request body will be applied. For subsequent synchronizations, these parameters will default to the values specified during the first synchronization.

3. Check details about the HyG graph.
 - a. Send **GET** `/ges/v1.0/{project_id}/hyg/{graph_name}/summary`. *project_id* indicates the project ID, and *graph_name* indicates the graph name.
 - b. Add **X-Auth-Token** to the request header.
 - c. Check the response. The request is successful if the following response is displayed:

```
{
  "data": {
    "inEdge": true,
    "idIndex": true,
    "policy": "oec",
    "updateTime": "2024-01-25 10:55:31",
    "vertex": [],
    "edge": [
      {
        "label": "rate",
        "property": [
          "Rating"
        ]
      }
    ],
    "vertexNum": 146,
    "edgeNum": 1659
  },
  "result": "success"
}
```

If the request is abnormal, locate the fault by referring to [Error Codes for Service Plane APIs](#).

4. Execute the algorithm.
 - a. Send **POST** `/ges/v1.0/{project_id}/hyg/{graph_name}/algorithm`. *project_id* indicates the project ID, and *graph_name* indicates the graph name.
 - b. Add **X-Auth-Token** to the request header.
 - c. Specify the following parameters in the request body (using PageRank as an example):

```
{
  "algorithmName": "pagerank",
  "parameters": {
    "alpha": 0.85,
    "convergence": 0.00001,
    "max_iterations": 1000,
    "directed": true
  }
}
```

- d. Check the response based on the job ID. The request is successful if the following response is displayed:

```
{
  "status": "complete",
  "data": {
    "outputs": {
      "data_offset": 0,
      "data_return_size": 147,
      "data_total_size": 147,
      "pagerank": [
        {
          "38": 0.02115960730038959
        },
        {
          "13": 0.018535705068819635
        },
        {
          "7": 0.0166381431701182
        },
        ... ..
      ],
      "runtime": 0.022
    }
  },
  "result": "success"
}
```

If the request is abnormal, locate the fault by referring to [Error Codes for Service Plane APIs](#).

5. Delete the HyG graph.
 - a. Send **DELETE** `/ges/v1.0/{project_id}/hyg/{graph_name}`. *project_id* indicates the project ID, and *graph_name* indicates the graph name.
 - b. Add **X-Auth-Token** to the request header.
 - c. Check the response based on the job ID. The request is successful if the following response is displayed:

```
{
  "status": "complete",
  "result": "success"
}
```

If the request is abnormal, locate the fault by referring to [Error Codes for Service Plane APIs](#).

9 Permissions Policies and Supported Actions

This chapter describes fine-grained permissions management for your GES. If your Huawei Cloud account does not need individual IAM users, then you may skip over this chapter.

By default, new IAM users do not have permissions assigned. You need to add the users to one or more groups, and attach permissions policies or roles to these groups. The users then inherit permissions from the groups to which they are added. After authorization, the users can perform specified operations on GES based on the permissions. For more information about policy syntax and example policies, see [Permissions Management](#).

Type: There are roles and policies.

- **Roles:** A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you need to also assign other roles on which the permissions depend to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- **Policies:** A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant GES users only the permissions for managing a certain type of cloud servers.

NOTE

- If you want to allow or deny the access to an API, fine-grained authorization is a good choice.
- Because of the cache, it takes about 13 minutes for an OBS role to take effect after being granted to users and user groups. After a policy is granted, it takes about 5 minutes to take effect.

An account has full API permissions, but IAM users within the account need to be granted the necessary permissions to call the APIs. The permissions required for calling an API are determined by the actions supported by the API. Only users who have been granted permissions allowing the actions can call the API successfully.

For example, if an IAM user queries a graph using an API, the user must have been granted permissions that allow the **ges:graph:list** action.

Supported Actions

Operations supported by policies are specific to APIs. The following are common concepts related to policies:

- **Permissions:** defined by actions in a custom policy.
- **APIs:** RESTful APIs that can be called in a custom policy.
- **Actions:** added to a custom policy to control permissions for specific operations.
- **Related actions:** Actions on which a specific action depends to take effect. When assigning permissions for the action to a user, you also need to assign permissions for the related actions.
- **IAM or enterprise projects:** Type of projects for which an action will take effect. Policies that contain actions supporting both IAM and enterprise projects can be assigned to user groups and take effect in both IAM and Enterprise Management. Policies that only contain actions supporting IAM projects can be assigned to user groups and only take effect in IAM. Such policies will not take effect if they are assigned to user groups in Enterprise Management. For details about the differences between IAM and enterprise projects, see [Differences Between IAM and Enterprise Management](#).

NOTE

The check mark (√) indicates that an action takes effect. The cross mark (x) indicates that an action does not take effect.

GES supports the following actions that can be defined in custom policies:

Table 9-1 Actions

Permission	API	Action	Related Action	IAM Project (Project)	Enterprise Project
Listing graphs	GET /v1.0/{project_id}/graphs?offset={offset}&limit={limit}	ges:graph:list	Project-level service VPC vpc:publicips:get	√	√
Querying graph details	GET /v1.0/{project_id}/graphs/{graph_id}	ges:graph:get Detail	-	√	√

Permission	API	Action	Related Action	IAM Project (Project)	Enterprise Project
Creating a graph	POST /v1.0/{project_id}/graphs	ges:graph:create	Project-level service ECS ecs:cloudServerFlavors:get ecs:cloudServerNics:update Project-level service EVS evs:types:get Project-level service VPC vpc:ports:create vpc:ports:update vpc:securityGroups:get vpc:publicIps:list vpc:ports:get vpc:subnets:get vpc:vpcs:list	√	√
Stopping a graph	POST /v1.0/{project_id}/graphs/{graph_id}/action?action_id=stop	ges:graph:operate	-	√	√

Permission	API	Action	Related Action	IAM Project (Project)	Enterprise Project
Starting a graph	POST /v1.0/{project_id}/graphs/{graph_id}/action?action_id=start	ges:graph:operate	-	√	√
Deleting a graph	DELETE /v1.0/{project_id}/graphs/{graph_id}	ges:graph:delete	Project-level service VPC vpc:publicips:get vpc:ports:delete vpc:ports:get	√	√
Incrementally importing data into a graph	POST /v1.0/{project_id}/graphs/{graph_id}/action?action_id=import-graph	ges:graph:operate	-	√	√
Exporting a graph	POST /v1.0/{project_id}/graphs/{graph_id}/action?action_id=export-graph	ges:graph:operate	-	√	√
Clearing a graph	POST /v1.0/{project_id}/graphs/{graph_id}/action?action_id=clear-graph	ges:graph:operate	-	√	√
Upgrading a graph	POST /v1.0/{project_id}/graphs/{graph_id}/action?action_id=upgrade	ges:graph:operate	-	√	√
Restarting a graph	POST /v1.0/{project_id}/graphs/{graph_id}/action	ges:graph:operate	-	√	√
Resizing a graph	POST /v1.0/{project_id}/graphs/{graph_id}/resize	ges:graph:resize	-	√	√

Permission	API	Action	Related Action	IAM Project (Project)	Enterprise Project
Expanding a graph	POST /v1.0/{project_id}/graphs/{graph_id}/expand	ges:graph:expand	Project-level service ECS ecs:cloudServerFlavors:get ecs:cloudServerNics:update Project-level service EVS evs:types:get Project-level service VPC vpc:ports:create vpc:ports:update vpc:securityGroups:get vpc:publicIps:list vpc:ports:get vpc:subnets:get vpc:vpcs:list	√	√
Binding an EIP	POST /v1.0/{project_id}/graphs/{graph_id}/action?action_id=bindEip	ges:graph:operate	-	√	√

Permission	API	Action	Related Action	IAM Project (Project)	Enterprise Project
Unbinding an EIP	POST /v1.0/{project_id}/graphs/{graph_id}/action?action_id=unbindEip	ges:graph:operate	-	√	√
Checking the list of all backups	GET /v1.0/{project_id}/graphs/backups?offset={offset}&limit={limit}	ges:backup:list	-	√	√
Checking the backup list of a graph	GET /v1.0/{project_id}/graphs/{graph_id}/backups?offset={offset}&limit={limit}	ges:backup:list	-	√	√
Creating a backup	POST /v1.0/{project_id}/graphs/{graph_id}/backups	ges:backup:create	-	√	√
Deleting a backup	DELETE /v1.0/{project_id}/graphs/{graph_id}/backups/{backup_id}	ges:backup:delete	-	√	√
Listing metadata files	GET /v1.0/{project_id}/graphs/metadatas?offset={offset}&limit={limit}	ges:metadata:list	-	√	√
Querying metadata	GET /v1.0/{project_id}/graphs/metadatas/{metadata_id}	ges:metadata:list	-	√	√
Verifying metadata	POST /v1.0/{project_id}/graphs/action?action_id=check-schema	ges:metadata:operate	-	√	√
Adding metadata	POST /v1.0/{project_id}/graphs/metadatas	ges:metadata:create	-	√	√

Permission	API	Action	Related Action	IAM Project (Project)	Enterprise Project
Deleting metadata	DELETE /v1.0/{project_id}/graphs/metadatas/{metadata_id}	ges:metadata:delete	-	√	√
Querying task status	GET /v1.0/{project_id}/graphs/{graph_id}/jobs/{job_id}/status	ges:jobs:getDetail	-	√	√
Listing tasks	GET /v1.0/{project_id}/graphs/jobs?offset={offset}&limit={limit}	ges:jobs:list	-	√	√

10 GES Metrics on Cloud Eye

Function

This chapter describes metrics reported by GES to Cloud Eye as well as their namespaces, lists, and dimensions. You can use APIs provided by Cloud Eye to query the metric information generated for GES.

Namespace

SYS.GES

Metrics

Table 10-1 GES metrics

Metric ID	Name	Description	Value Range	Monitored Object
ges001_vertex_util	Vertex Capacity Usage	Capacity usage of vertices in a graph instance. The value is the ratio of the number of used vertices to the total vertex capacity. Unit: %	0 to 100 Value type: Float	GES instance
ges002_edge_util	Edge Capacity Usage	Capacity usage of edges in a graph instance. The value is the ratio of the number of used edges to the total edge capacity. Unit: %	0 to 100 Value type: Float	GES instance
ges003_average_import_rate	Average Import Rate	Average rate of importing vertices or edges to a graph instance Unit: count/s	0 to 400000 Value type: Float	GES instance

Metric ID	Name	Description	Value Range	Monitored Object
ges004_request_count	Request Quantity	Number of requests received by a graph instance Unit: count	≥ 0 Value type: Integer	GES instance
ges005_average_response_time	Average Response Time	Average response time of requests received by a graph instance Unit: ms	≥ 0 Value type: Integer	GES instance
ges006_min_response_time	Minimum Response Time	Minimum response time of requests received by a graph instance Unit: ms	≥ 0 Value type: Integer	GES instance
ges007_max_response_time	Maximum Response Time	Maximum response time of requests received by a graph instance Unit: ms	≥ 0 Value type: Integer	GES instance
ges008_read_task_pending_queue_size	Length of the Waiting Queue for Read Tasks	Length of the waiting queue for read requests received by a graph instance. This metric is used to view the number of read requests waiting in the queue. Unit: count	≥ 0 Value type: Integer	GES instance
ges009_read_task_pending_max_time	Maximum Waiting Duration of Read Tasks	Maximum waiting duration of read requests received by a graph instance Unit: ms	≥ 0 Value type: Integer	GES instance
ges010_pending_max_time_read_task_type	Type of the Read Task That Waits the Longest	Type of the read request that waits the longest in a graph instance. You can find the corresponding task name in GES documents.	≥ 1 Value type: Integer	GES instance
ges011_read_task_running_queue_size	Length of the Running Queue for Read Tasks	Length of the running queue for read requests received by a graph instance. This metric is used to view the number of running read requests. Unit: count	≥ 0 Value type: Integer	GES instance

Metric ID	Name	Description	Value Range	Monitored Object
ges012_read_task_running_max_time	Maximum Running Duration of Read Tasks	Maximum running duration of read requests received by a graph instance Unit: ms	≥ 0 Value type: Integer	GES instance
ges013_running_max_time_read_task_type	Type of the Read Task That Runs the Longest	Type of the read request that runs the longest in a graph instance. You can find the corresponding task name in GES documents.	≥ 1 Value type: Integer	GES instance
ges014_write_task_pending_queue_size	Length of the Waiting Queue for Write Tasks	Length of the waiting queue for write requests received by a graph instance. This metric is used to view the number of write requests waiting in the queue. Unit: count	≥ 0 Value type: Integer	GES instance
ges015_write_task_pending_max_time	Maximum Waiting Duration of Write Tasks	Maximum waiting duration of write requests received by a graph instance Unit: ms	≥ 0 Value type: Integer	GES instance
ges016_pending_max_time_write_task_type	Type of the Write Task That Waits the Longest	Type of the write request that waits the longest in a graph instance. You can find the corresponding task name in GES documents.	≥ 1 Value type: Integer	GES instance
ges017_write_task_running_queue_size	Length of the Running Queue for Write Tasks	Length of the running queue for write requests received by a graph instance. This metric is used to view the number of running write requests. Unit: count	≥ 0 Value type: Integer	GES instance

Metric ID	Name	Description	Value Range	Monitored Object
ges018_write_task_running_max_time	Maximum Running Duration of Write Tasks	Maximum running duration of write requests received by a graph instance Unit: ms	≥ 0 Value type: Integer	GES instance
ges019_running_max_time_write_task_type	Type of the Write Task That Runs the Longest	Type of the write request that runs the longest in a graph instance. You can find the corresponding task name in GES documents.	≥ 1 Value type: Integer	GES instance
ges020_computer_resource_usage	Computing Resource Usage	Computing resource usage of each graph instance Unit: %	0 to 100 Value type: Float	GES instance
ges021_memory_usage	Memory Usage	Memory usage of each graph instance Unit: %	0 to 100 Value type: Float	GES instance
ges022_iops	IOPS	Number of I/O requests processed by each graph instance per second Unit: count/s	≥ 0 Value type: Integer	GES instance
ges023_bytes_in	Network Input Throughput	Data input to each graph instance per second over the network Unit: byte/s	≥ 0 Value type: Float	GES instance
ges024_bytes_out	Network Output Throughput	Data sent to the network per second from each graph instance Unit: byte/s	≥ 0 Value type: Float	GES instance
ges025_disk_usage	Disk Usage	Disk usage of each graph instance Unit: %	0 to 100 Value type: Float	GES instance
ges026_disk_total_size	Total Disk Size	Total data disk space of each graph instance Unit: GB	≥ 0 Value type: Float	GES instance

Metric ID	Name	Description	Value Range	Monitored Object
ges027_disk_used_size	Disk Space Used	Used data disk space of each graph instance Unit: GB	≥ 0 Value type: Float	GES instance
ges028_disk_read_throughput	Disk Read Throughput	Data volume read from the disk in a graph instance per second Unit: byte/s	≥ 0 Value type: Float	GES instance
ges029_disk_write_throughput	Disk Write Throughput	Data volume written to the disk in a graph instance per second Unit: byte/s	≥ 0 Value type: Float	GES instance
ges030_avg_disk_sec_per_read	Average Time per Disk Read	Average time used each time when the disk of a graph instance reads data Unit: second	≥ 0 Value type: Float	GES instance
ges031_avg_disk_sec_per_write	Average Time per Disk Write	Average time used each time when data is written to the disk of a graph instance Unit: second	≥ 0 Value type: Float	GES instance
ges032_avg_disk_queue_length	Average Disk Queue Length	Average I/O queue length of the disk in a graph instance Unit: count	≥ 0 Value type: Integer	GES instance

Dimensions

Table 10-2 Dimensions

Key	Value
instance_id	GES instance

Mapping Between Task Types and Names

Table 10-3 Task types and corresponding task names

Type	Name
100	Querying a vertex
101	Creating a vertex
102	Deleting a vertex
103	Modifying a vertex property
104	Adding a vertex label
105	Deleting a vertex label
200	Querying an edge
201	Creating an edge
202	Deleting an edge
203	Modifying an edge property
300	Querying schema details
301	Adding a Label
302	Modifying a Label
303	Querying a Label
304	Modifying a property
400	Querying graph details
401	Clearing a graph
402	Incrementally importing graph data online
403	Creating a graph
405	Deleting a graph
406	Exporting a graph
407	filtered_khop
408	Querying path details
409	Incrementally importing graph data offline
500	Creating a graph backup
501	Restoring a graph from a backup
601	Creating an index.

Type	Name
602	Querying an index
603	Updating an index
604	Deleting an index
700	Running the algorithm
800	Querying an asynchronous task

11 Out-of-Date APIs

11.1 Management Plane APIs (V1)

11.1.1 System Management APIs

11.1.1.1 Viewing Quotas (1.0.0)

Function

This API is used to query tenant quotas.

URI

GET /v1.0/{project_id}/graphs/quotas

Table 11-1 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Request Parameters

Table 11-2 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-3 Response body parameter

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
quotas	GesQuotaResp object	Resource type list. This field is left blank when the request fails.

Table 11-4 GesQuotaResp

Parameter	Type	Description
resources	Array of Quota objects	GES resource quota list

Table 11-5 Quota

Parameter	Type	Description
type	String	Quota type. Available values are as follows: • "graph" • "backup" • "metadata"
available	Integer	Number of available graphs When the value is -1 , the number of available graphs is unlimited.
edgeVolume	Integer	Number of available edges. This parameter is available only when type is graph . When the value is -1 , the number of available edges is unlimited.

Example Request

GET https://{Endpoint}/v1.0/{project_id}/graphs/quotas

Example Response

Status code: 200

OK

```
{
  "quotas": {
    "resources": [ {
      "type": "graph",
      "available": 1,
      "edgeVolume": 178800
    }, {
      "type": "backup",
      "available": 7
    }, {
      "type": "metadata",
      "available": 13
    }
  ]
}
```

Status Codes

Status Code	Description
200	Request sent.
400	Request error.
401	Authorization failed.
403	No operation permissions.

Status Code	Description
404	No resources found.
500	Internal server error.
503	Service unavailable.

11.1.2 Graph Management APIs

11.1.2.1 Querying the Graph List (2.1.18)

Function

This API is used to query all graphs owned by the current tenant.

URI

GET /v1.0/{project_id}/graphs

Table 11-6 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Table 11-7 Query parameters

Parameter	Mandatory	Type	Description
offset	No	Integer	Start position of the request. The default value is 0 .
limit	No	Integer	Maximum number of resources displayed on a single page. The default value is 10 .

Request Parameters

Table 11-8 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200**Table 11-9** Response body parameter

Parameter	Type	Description
graphCount	Integer	Total number of graphs. This parameter is left blank when the request fails.
graphs	Array of graph_1 objects	Graph list. This parameter is left blank when the request fails.
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.

Table 11-10 graph_1

Parameter	Type	Description
id	String	Graph ID
name	String	Graph name
createdBy	String	IAM username
isMultiAz	String	Whether to enable cross-AZ HA

Parameter	Type	Description
regionCode	String	Region code
azCode	String	AZ code
schemaPath	Array of schemaPath_1 objects	Path for storing the metadata file
edgesetPath	Array of edgesetPath_1 objects	OBS path for storing the edge data set
edgesetFormat	String	Format of the edge data file
edgesetDefaultLabel	String	Default label of the edge data file
vertexsetPath	Array of vertexsetPath_1 objects	OBS path for storing the vertex data set
vertexsetFormat	String	Format of the vertex data file
vertexsetDefaultLabel	String	Default label of the vertex data file
dataStoreVersion	String	Graph version
sys_tags	Array of strings	Enterprise project information. If this parameter is not specified, this function is disabled (default).

Parameter	Type	Description
status	String	Status code of a graph: ● 100: Indicates that a graph is being prepared. ● 200: indicates that a graph is running. ● 201: indicates that a graph is upgrading. ● 202: indicates that a graph is being imported. ● 203: indicates that a graph is being rolled back. ● 204: indicates that a graph is being exported. ● 205: indicates that a graph is being cleared. ● 206: indicates that the system is preparing for resize. ● 207: indicates that the resize is in progress. ● 208: Indicates that the resize is being rolled back. ● 210: Preparing for expansion ● 211: Expanding ● 300: indicates that a graph is faulty. ● 303: indicates that a graph fails to be created. ● 400: indicates that a graph is deleted. ● 800: indicates that a graph is frozen. ● 900: indicates that a graph is stopped. ● 901: indicates that a graph is being stopped. ● 920: indicates that a graph is being started.
actionProgress	String	Progress of graph creation in percentage NOTE This field is returned only when status is 100.
graphSizeTypeIndex	String	Graph size type index: ● 0: Ten-thousand-edge ● 1: Million-edge ● 2: Ten-million-edge ● 3: Hundred-million-edge ● 4: Billion-edge ● 5: Ten-billion-edge ● 6: Hundred-billion-edge ● 401: Billion-edge-pro
vpId	String	VPC ID

Parameter	Type	Description
subnetId	String	Subnet ID in the VPC
securityGroupId	String	Security group ID
replication	Integer	Number of replicas. The default value is 1 .
created	String	Time when a graph is created
updated	String	Time when a graph is updated
privateIp	String	Floating IP address of a graph instance. Users can access the instance using the IP address through the ECS deployed on a private network.
publicIp	String	Public network access address of a graph instance. Users can access the instance using the IP address from the Internet.
arch	String	Graph instance's CPU architecture type. The value can be x86_64 or aarch64 .
masterKeyId	String	User master key ID
masterKeyName	String	User master key name
enableRBAC	Boolean	Whether to enable granular permission control
enableFulltextIndex	Boolean	Whether to enable full-text indexes
enableHyG	Boolean	Whether to enable HyG. This parameter is available only for database edition graphs.
trafficIpList	Array of strings	Physical addresses of a graph instance for access from private networks. To prevent service interruption caused by floating IP address switchover, poll the physical IP addresses to access the graph instance.
cryptAlgorithm	String	Graph instance cryptography algorithm. The options are: • generalCipher: Chinese cryptographic algorithm • SMcompatible: Commercial cryptography algorithm (compatible with international ones)
enableHttps	Boolean	Whether to enable the security mode. This mode may damage GES performance greatly.
tags	Array of objects	Tag list. Each tag is in <key,value> format.

Table 11-11 schemaPath_1

Parameter	Type	Description
jobId	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file status: • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Table 11-12 edgesetPath_1

Parameter	Type	Description
jobId	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file status: • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Table 11-13 vertexsetPath_1

Parameter	Type	Description
jobId	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file status: • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Example Request

```
GET https://{Endpoint}/v1.0/{project_id}/graphs?offset=0&limit=10
```

Example Response

Status code: 200

OK

```
{
  "graphCount": 2,
  "graphs": [
    {
      "id": "f1529b88-c958-493e-8452-fccfe932cde1",
      "name": "demo",
      "regionCode": "ap-southeast-1",
      "azCode": "ap-southeast-1a",
      "schemaPath": [
        {
          "path": "ges-graphs/demo_movie/schema.xml",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ],
      "edgesetPath": [
        {
          "path": "ges-graphs/demo_movie/edge.csv",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ],
      "vertexsetPath": [
        {
          "path": "",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ],
      "status": "200",
      "graphSizeTypeIndex": "1",
      "vpId": "2d8af840-fd57-4e3b-a8f1-cda0f55ccd99",
      "subnetId": "dc018ec3-67d1-46c9-b2fc-19d83367f4e2",
      "securityGroupId": "11d27338-8649-4076-8579-5ebc1a60f79e",
      "created": "2018-07-23T04:09:44",
      "updated": "2018-07-23T04:09:44",
      "privateIp": "192.168.0.4",
      "publicIp": "49.4.81.183",
      "dataStoreVersion": "1.0.5",
      "arch": "x86_64",
      "enableFullTextIndex": false
    },
    {
      "id": "53205529-026b-455a-9e07-228fae4b12b9",
      "name": "ges_c5de",
      "regionCode": "ap-southeast-1",
      "azCode": "ap-southeast-1a",
      "schemaPath": [
        {
          "path": "ges-graphs/demo_movie/schema.xml",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ],
      "edgesetPath": [
        {
          "path": "ges-graphs/demo_movie/edge.csv",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ],
      "vertexsetPath": [
        {
          "path": "",
          "jobId": "ff80808167bb90340167bc7445670428",
          "status": "success"
        }
      ]
    }
  ]
}
```

```
    ],
    "status": "200",
    "graphSizeTypeIndex": "2",
    "vpclId": "2d8af840-fd57-4e3b-a8f1-cda0f55ccd99",
    "subnetId": "dc018ec3-67d1-46c9-b2fc-19d83367f4e2",
    "securityGroupId": "11d27338-8649-4076-8579-5ebc1a60f79e",
    "created": "2018-07-18T13:30:16",
    "updated": "2018-07-18T13:30:16",
    "privateIp": "192.168.0.168",
    "dataStoreVersion": "1.0.5",
    "arch": "aarch64",
    "enableFullTextIndex" : false
  }
]
```

Status Codes

Status Code	Description
200	Request sent.
400	Request error.
401	Authorization failed.
403	No operation permissions.
404	No resources found.
500	Internal server error.
503	Service unavailable.

11.1.2.2 Querying Graph Details (1.0.0)

Function

This API is used to query the details about a graph based on the graph ID.

URI

GET /v1.0/{project_id}/graphs/{graph_id}

Table 11-14 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 11-15 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-16 Response body parameter

Parameter	Type	Description
graph	graph_1 object	Graph object. If the request fails, this parameter is left empty.
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.

Table 11-17 graph_1

Parameter	Type	Description
id	String	Graph ID
name	String	Graph name
createdBy	String	IAM username
isMultiAz	String	Whether to enable cross-AZ HA

Parameter	Type	Description
regionCode	String	Region code
azCode	String	AZ code
schemaPath	Array of schemaPath_1 objects	Path for storing the metadata file
edgesetPath	Array of edgesetPath_1 objects	OBS path for storing the edge data set
edgesetFormat	String	Format of the edge data file
edgesetDefaultLabel	String	Default label of the edge data file
vertexsetPath	Array of vertexsetPath_1 objects	OBS path for storing the vertex data set
vertexsetFormat	String	Format of the vertex data file
vertexsetDefaultLabel	String	Default label of the vertex data file
dataStoreVersion	String	Graph version
sys_tags	Array of strings	Enterprise project information. If this parameter is not specified, this function is disabled (default).

Parameter	Type	Description
status	String	Status code of a graph: ● 100: Indicates that a graph is being prepared. ● 200: indicates that a graph is running. ● 201: indicates that a graph is upgrading. ● 202: indicates that a graph is being imported. ● 203: indicates that a graph is being rolled back. ● 204: indicates that a graph is being exported. ● 205: indicates that a graph is being cleared. ● 206: indicates that the system is preparing for resize. ● 207: indicates that the resize is in progress. ● 208: Indicates that the resize is being rolled back. ● 210: Preparing for expansion ● 211: Expanding ● 300: indicates that a graph is faulty. ● 303: indicates that a graph fails to be created. ● 400: indicates that a graph is deleted. ● 800: indicates that a graph is frozen. ● 900: indicates that a graph is stopped. ● 901: indicates that a graph is being stopped. ● 920: indicates that a graph is being started.
actionProgress	String	Progress of graph creation in percentage NOTE This field is returned only when status is 100.
graphSizeTypeIndex	String	Graph size type index: ● 0: Ten-thousand-edge ● 1: Million-edge ● 2: Ten-million-edge ● 3: Hundred-million-edge ● 4: Billion-edge ● 5: Ten-billion-edge ● 401: Billion-edge-pro
vpId	String	VPC ID

Parameter	Type	Description
subnetId	String	Subnet ID in the VPC
securityGroupId	String	Security group ID
replication	Integer	Number of replicas. The default value is 1 .
created	String	Time when a graph is created
updated	String	Time when a graph is updated
privatelp	String	Private network access address of a graph instance. Users can access the instance using the IP address through the ECS deployed on the private network.
publiclp	String	Public network access address of a graph instance. Users can access the instance using the IP address from the Internet.
arch	String	Graph instance's CPU architecture type. The value can be x86_64 or aarch64 .
masterKeyId	String	User master key ID
masterKeyName	String	User master key name
enableRBAC	Boolean	Whether to enable granular permission control
enableFulltextIndex	Boolean	Whether to enable full-text indexes
enableHyG	Boolean	Whether to enable HyG. This parameter is available only for database edition graphs.
trafficIpList	Array of strings	Physical addresses of a graph instance for access from private networks. To prevent service interruption caused by floating IP address switchover, poll the physical IP addresses to access the graph instance.
cryptAlgorithm	String	Graph instance cryptography algorithm. The options are: • generalCipher: Chinese cryptographic algorithm • SMcompatible: Commercial cryptography algorithm (compatible with international ones)
enableHttps	Boolean	Whether to enable the security mode. This mode may damage GES performance greatly.
tags	Array of strings	Tag list. Each tag is in <key,value> format.

Table 11-18 schemaPath_1

Parameter	Type	Description
jobId	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file status: • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Table 11-19 edgesetPath_1

Parameter	Type	Description
jobId	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file status: • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Table 11-20 vertexsetPath_1

Parameter	Type	Description
jobId	String	Job ID corresponding to OBS file import
path	String	OBS storage path, excluding OBS endpoint
status	String	OBS file status: • success: Imported successfully. • partiallyFailed: Partially failed. • failed: Failed to import the file.

Example Request

```
GET https://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}
```

Example Response

Status code: 200

OK

```
{
  "graph": {
    "id": "f1529b88-c958-493e-8452-fccfe932cde1",
    "name": "demo",
    "regionCode": "ap-southeast-1",
    "azCode": "ap-southeast-1a",
    "schemaPath": [
      {
        "path": "ges-graphs/demo_movie/schema.xml",
        "jobId": "ff80808167bb90340167bc7445670428",
        "status": "success"
      }
    ],
    "edgesetPath": [
      {
        "path": "ges-graphs/demo_movie/edge.csv",
        "jobId": "ff80808167bb90340167bc7445670428",
        "status": "success"
      }
    ],
    "vertexsetPath": [
      {
        "path": "",
        "jobId": "ff80808167bb90340167bc7445670428",
        "status": "success"
      }
    ],
    "status": "200",
    "graphSizeTypeIndex": "1",
    "vpclId": "2d8af840-fd57-4e3b-a8f1-cda0f55ccd99",
    "subnetId": "dc018ec3-67d1-46c9-b2fc-19d83367f4e2",
    "securityGroupId": "11d27338-8649-4076-8579-5ebc1a60f79e",
    "created": "2018-07-23T04:09:44",
    "privateIp": "192.168.0.4",
    "publicIp": "49.4.81.183",
    "dataStoreVersion": "1.0.5",
    "arch": "x86_64"
  }
}
```

Status Codes

Status Code	Description
200	Request sent.
400	Request error.
401	Authorization failed.
403	No operation permissions.
404	No resources found.
500	Internal server error.
503	Service unavailable.

11.1.2.3 Creating a Graph (2.2.2)

Function

This API is used to create a graph.

URI

POST /v1.0/{project_id}/graphs

Table 11-21 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Request Parameters

Table 11-22 Request header parameter

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 11-23 Request body parameter

Parameter	Mandatory	Type	Description
graph	Yes	graph object	Graph type

Table 11-24 graph

Parameter	Mandatory	Type	Description
name	Yes	String	A graph name must start with a letter and contains 4 to 50 characters consisting of letters, digits, hyphens (-), and underscores (_). It cannot contain special characters.
graphSizeTypeIndex	Yes	String	Graph size type index: • 0: Ten-thousand-edge • 1: Million-edge • 2: Ten-million-edge • 3: Hundred-million-edge • 4: Billion-edge • 5: Ten-billion-edge • 6: Hundred-billion-edge • 401: Billion-edge-pro
arch	No	String	Graph instance's CPU architecture type. The value can be x86_64 or aarch64 . The default value is x86_64 . • x86_64: x86 64-bit architecture • aarch64: Arm 64-bit architecture
vpcId	Yes	String	VPC ID
subnetId	Yes	String	Subnet ID in the VPC
securityGroupId	Yes	String	Security group ID
publicIp	No	publicIp object	Public IP address. If the parameter is not specified, public connection is not used by default.
enableMultiAZ	No	Boolean	Whether the created graph supports the cross-AZ mode. The default value is false . If the value is true , the system will create the ECSs in the graph in two AZs. If this parameter is not specified when you create a graph, all ECSs in the graph are created in one AZ.

Parameter	Mandatory	Type	Description
ItsOperationTrace	No	ItsOperationTraceReq object	Whether to enable audit logs. This function is disabled by default.
sys_tags	No	Array of SysTagsRes objects	Enterprise project information. If this parameter is not specified, this function is disabled (default).
tags	No	Array of SysTagsRes objects	TMS tags for expenses. This function is disabled by default.
enableRBAC	No	Boolean	Whether to enable granular permission control for the created graph. The default value is false , indicating that granular permission control is disabled. If this parameter is set to true , no user has the permission to access the graph. To access the graph, you need to call the granular permission control API of the service plane to set the required permissions.
enableFullTextIndex	No	Boolean	Whether to enable full-text index control for the created graph. The default value is false, indicating that full-text index control is disabled. If this parameter is set to true, full-text indexes are available for Billion-edge-pro graphs, and a Cloud Search Service (CSS) cluster will be created when you create a graph. NOTE If you enable full-text indexes: If the CSS has been deployed, the system automatically creates a CSS cluster during the creation of the graph instance, which will take a long time. If the CSS is not deployed, the graph creation will fail.
enableHyG	No	Boolean	Whether to enable HyG for the graph. This parameter is available for database edition graphs only.

Parameter	Mandatory	Type	Description
cryptAlgorithm	No	String	Graph instance cryptography algorithm. Available values are as follows: • generalCipher: Chinese cryptographic algorithm • SMcompatible: Commercial cryptography algorithm (compatible with international ones)
enableHttps	No	Boolean	Whether to enable the security mode. This mode may damage GES performance greatly.
tags	No	JsonArray	Tag list. Each tag is in <key,value> format.

Table 11-25 parameters

Parameter	Mandatory	Type	Description
schemaPath	Yes	String	OBS path for storing the metadata file. Only files are supported.
edgesetPath	Yes	String	OBS path for storing the edge file. Only files are supported.
edgesetFormat	No	String	Edge dataset format. Currently, the options are csv and txt . The CSV format is used by default.
edgesetDefaultLabel	No	String	Default label of an edge data set. This parameter is left blank by default.
vertexsetPath	No	String	OBS path for storing the vertex file. Only files are supported.
vertexsetFormat	No	String	Vertex dataset format. Currently, the options are csv and txt . The CSV format is used by default.

Parameter	Mandatory	Type	Description
vertexsetDefaultLabel	No	String	Default label of a vertex data set. This parameter is left blank by default.
logDir	No	String	OBS log storage directory. This directory stores the data that fails to be imported during graph creation and detailed logs.
parallelEdge	No	parallelEdge object	How to process repetitive edges.

 NOTE

- For details about the value validity of the **schemaPath**, **edgesetPath**, **vertexsetPath**, and **logDir** character strings, see the [OBS Object Name Restrictions](#).

Table 11-26 parallelEdge

Parameter	Mandatory	Type	Description
action	No	String	Processing mode of repetitive edges. The value can be allow , ignore , or override . The default value is allow . allow indicates that repetitive edges are allowed. ignore indicates that subsequent repetitive edges are ignored. override indicates that the previous repetitive edges are overwritten.
ignoreLabel	No	Boolean	Whether to ignore labels on repetitive edges. The value can be true or false . The default value is true . true: Indicates that the repetitive edge definition does not contain the label. That is, the <source vertex, target vertex> indicates an edge, excluding the label information. false: Indicates that the repetitive edge definition contains the label. That is, the <source vertex, target vertex, label> indicates an edge.

Table 11-27 publicIp

Parameter	Mandatory	Type	Description
publicBindType	No	String	Binding type of an EIP. The value can be either of the following: • auto_assign • bind_existing
eipId	No	String	EIP ID • If publicBindType is bind_existing, the value is the ID of a created EIP that has not been bound. • If publicBindType is auto_assign, leave this parameter blank.

Table 11-28 ltsOperationTraceReq

Parameter	Mandatory	Type	Description
enableAudit	No	Boolean	Whether to enable graph audit. The value can be true or false . The default value is false .
auditLogGroupName	No	String	LTS log group name

Table 11-29 SysTagsRes

Parameter	Mandatory	Type	Description
key	No	String	Key of the enterprise project. Set this parameter to _sys_enterprise_project_id .
value	No	String	Enterprise project ID. You can obtain it from the enterprise project.

Response Parameters

Status code: 200

Table 11-30 Response body parameters

Parameter	Type	Description
id	String	Graph ID
name	String	Graph name
errorMessage	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.

Example Request

```
POST https://{Endpoint}/v1.0/{project_id}/graphs
{
  "graph":{
    "name":"demo",
    "graphSizeTypeIndex": "1",
    "arch":"x86_64",
    "vpclId":"2d8af840-fd57-4e3b-a8f1-cda0f55ccd99",
    "subnetId":"dc018ec3-67d1-46c9-b2fc-19d83367f4e2",
    "securityGroupId":"11d27338-8649-4076-8579-5ebc1a60f79e",
    "publicIp":{
      "publicBindType":"bind_existing",
      "eipId":"30ef2d58-08a9-4481-b526-b2cbe67d020d"
    },
    "enableMultiAz":false,
    "sys_tags": [{"key": "_sys_enterprise_project_id", "value": "54c0b33c-8627-462f-948e-bae08c0887b4"}],
    "enableRBAC":false,
    "enableFullTextIndex" : false,
    "cryptAlgorithm": "generalCipher",
    "enableHttps": "false"
  }
}
```

Example Response

Status code: 200

OK

```
{
  "id" : "f1529b88-c958-493e-8452-fccfe932cde1",
  "name" : "demo"
}
```

Status Codes

Status Codes	Description
200	Request sent.
400	Request error.
401	Authorization failed.
403	No operation permissions.
404	No resources found.
500	Internal server error.
503	Service unavailable.

11.1.2.4 Stopping a Graph (1.0.0)

Function

This API is used to stop a graph. After the graph is created, you can disable it if it is not used temporarily.

 NOTE

- Instances of stopped graphs are not charged.
- You can stop a graph instance for a maximum of seven days. If you do not restart the instance manually within that time frame, it will automatically restart.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/action

Table 11-31 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 11-32 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID The value can be: • stop

Request Parameters

Table 11-33 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-34 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.

Parameter	Type	Description
jobId	String	ID of the graph stopping job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

Example Request

POST https://{Endpoint}/{v1.0}/{project_id}/graphs/{graph_id}/action?action_id=stop

Example Response

Status code: 200

OK

```
{
  "jobId" : "ff8080816025a0a1016025a5a2700007"
}
```

Status code: 400

Bad Request

```
{
  "errorCode" : "GES.7001",
  "errorMessage" : "The graph is not running."
}
```

Status Codes

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permissions.
404	No resources found.
500	Internal server error.
503	Service unavailable.

11.1.2.5 Starting a Graph (1.0.0)

Function

This API is used to start a graph. You can disable a graph if it is not used temporarily.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/action

Table 11-35 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 11-36 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID Set this parameter to the enumeration value start , which means the graph will be started.

Request Parameters

Table 11-37 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 11-38 Request body parameters

Parameter	Mandatory	Type	Description
graph_backup_id	No	String	Backup ID associated during graph startup. If this parameter is configured, the graph starts from the backup. If this parameter is left blank, the graph starts from the status when it was closed last time. For details about how to back up a graph, see section Adding a Backup (1.0.0) .

Response Parameters

Status code: 200

Table 11-39 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of the graph startup job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

Example Request

```
https://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/action?action_id=start
{
  "graph_backup_id" : "08a898ae-3ff8-40e8-a7ed-03afe05aedbb"
}
```

Example Response

Status code: 200

OK

```
{
  "jobId" : "ff8080816025a0a1016025a5a2700007"
}
```

Status code: 400

Bad Request

```
{
  "errorCode" : "GES.7015",
  "errorMessage" : "The graph is not running or stopped."
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.2.6 Deleting a Graph (1.0.0)

Function

This API is used to delete a graph.

URI

DELETE /v1.0/{project_id}/graphs/{graph_id}

Table 11-40 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 11-41 Query parameters

Parameter	Mandatory	Type	Description
keepBackup	No	Boolean	Whether to retain the backups of a graph after it is deleted. By default, one automatic backup and two manual backups are retained. If this parameter is left empty, no backups are retained.

Request Parameters

Table 11-42 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-43 Response body parameter

Parameter	Type	Description
jobId	String	ID of the graph deletion job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.

Example Request

```
DELETE https://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}
```

Example Response

Example response with status code **200**:

Example response for a successful request

```
{
  "jobId" : "ff8080816025a0a1016025a5a2700007"
}
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
  "error_code": "GES.0047"
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.2.7 Incrementally Importing Data to Graphs (2.1.14)

Function

This API is used to import data to graphs incrementally.

 NOTE

To prevent failures in restoring the imported graph data during system restarting, do not delete the data stored on OBS when the graph is in use.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/action

Table 11-44 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 11-45 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID Set this parameter to the enumeration value import-graph , which means the incremental data will be imported to the target graph.

Request Parameters

Table 11-46 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 11-47 Request body parameters

Parameter	Mandatory	Type	Description
edgesetPath	No	String	Edge file directory or name
edgesetFormat	No	String	Edge dataset format. Currently, the options are csv and txt . The CSV format is used by default.
vertexsetPath	No	String	Vertex file directory or name
vertexsetFormat	No	String	Vertex dataset format. Currently, the options are csv and txt . The CSV format is used by default.
schemaPath	No	String	Path for storing the metadata file of the new data.
logDir	No	String	Directory for storing logs of imported graphs. This directory stores the data that fails to be imported during graph creation and detailed error causes.
parallelEdge	No	Object	Repetitive edge processing This parameter is not available for database edition graphs.
action	No	String	Processing mode of repetitive edges. The value can be allow , ignore , or override . The default value is allow . • allow indicates that repetitive edges are allowed. • ignore indicates that subsequent repetitive edges are ignored. • override indicates that the previous repetitive edges are overwritten. This parameter is not available for database edition graphs.

Parameter	Mandatory	Type	Description
ignoreLabel	No	Boolean	Whether to ignore labels on repetitive edges. The value can be true or false. The default value is true. • true: Indicates that the repetitive edge definition does not contain the label. That is, the <source vertex, target vertex> indicates an edge, excluding the label information. • false: Indicates that the repetitive edge definition contains the label. That is, the <source vertex, target vertex, label> indicates an edge. This parameter is not available for database edition graphs.
targetProperties	No	Array	List of properties used to determine repetitive edges. If this parameter is specified, ignoreLabel is set to false to define repetitive edges that contain properties in the list specified by this parameter. (This parameter is mandatory only when action is set to override.) NOTE • The current version supports only overwriting by property and does not support ignoring by property. In addition, only one property takes effect for each label. • Properties of the non-single type are considered unequal.
delimiter	No	String	Field separator in a CSV file. The default value is comma (,). The default element separator in a field of the list/set type is semicolon (;).

Parameter	Mandatory	Type	Description
trimQuote	No	String	Field quote character in a CSV file. The default value is double quotation marks ("). It is used to enclose a field if the field contains separators or line breaks.
offline	No	Boolean	Whether offline import is selected. The value can be true or false . The default value is false . • true: Offline import is selected. The import speed is high, but the graph is locked and cannot be read or written during the import. • false: Online import is selected. Compared with offline import, online import is slower. However, the graph can be read (cannot be written) during the import.

 NOTE

- For details about the value validity of the **edgesetPath**, **vertexsetPath**, **schemaPath**, and **logDir** character strings, see the [OBS Object Name Restrictions](#).

Response Parameters

Status code: 200

Table 11-48 Response body parameter

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
jobId	String	Indicates the ID of an asynchronous job. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Example Request

```
POST http://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/action?action_id=import-graph
{
  "edgesetPath": "testbucket/demo_movie/edges/",
  "edgesetFormat": "csv",
  "vertexsetPath": "testbucket/demo_movie/vertices/",
  "vertexsetFormat": "csv",
  "schemaPath": "testbucket/demo_movie/incremental_data_schema.xml",
  "logDir": "testbucket/importlogdir",
  "parallelEdge": {
    "action": "override",
    "ignoreLabel": true
  },
  "delimiter": ",",
  "trimQuote": "\"",
  "offline": true
}
```

Example Response

Status code: 200

OK

```
{
  "jobId" : "b4f2e9a0-0439-4edd-a3ad-199bb523b613"
}
```

Status code: 400

Bad Request

```
{
  "errorMessage" : "parameter format error",
  "errorCode" : "GES.8013"
}
```

Status Code

Status Code	Description
200	Request sent.

Status Code	Description
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.2.8 Exporting a Graph (1.0.5)

Function

This API is used to export a graph.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/action

Table 11-49 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 11-50 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID Option: • export-graph

Request Parameters

Table 11-51 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 11-52 Request body parameters

Parameter	Mandatory	Type	Description
graphExportPath	Yes	String	OBS path to which a graph is exported
edgeSetName	Yes	String	Exported edge file name
vertexSetName	Yes	String	Exported vertex file name
schemaName	Yes	String	Name of the exported metadata file

 NOTE

- For details about the value validity of the **graphExportPath** character strings, see the [OBS Object Name Restrictions](#).

Response Parameters

Status code: 200

Table 11-53 Response body parameter

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

 NOTE

- For details about the value validity of the **graphExportPath** character strings, see the [OBS Object Name Restrictions](#).

Example Request

```
POST http://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/action?action_id=export-graph
{
  "graphExportPath" : "demo_movie/",
  "edgeSetName" : "set_edge.csv",
  "vertexSetName" : "set_vertex.csv",
  "schemaName" : "set_schema.xml"
}
```

Example Response

Status code: 200

OK

```
{
  "jobId" : "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"
}
```

Status code: 400

Bad Request

```
{
  "errorMessage" : "graph [demo] is not found",
  "errorCode" : "GES.8011"
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.

Status Code	Description
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.2.9 Clearing a Graph (2.1.2)

Function

This API is used to clear all data in a graph.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/action

Table 11-54 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 11-55 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID Set this parameter to the enumeration value clear-graph , which means that all data in the target graph will be cleared.

Parameter	Mandatory	Type	Description
clear-metadata	No	Boolean	Whether to clear graph metadata. Set this parameter to true . The value can be true or false . The default value is false . • true: The metadata will be cleared. • false: The metadata will not be cleared.

Request Parameters

Table 11-56 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-57 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.

Parameter	Type	Description
jobId	String	ID of an asynchronous job. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Example Request

```
POST http://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/action?action_id=clear-graph&clear-metadata=true
```

Example Response

Status code: 200

OK

```
{
  "jobId" : "ff8080816025a0a1016025a5a2700007"
}
```

Status code: 400

Bad Request

```
{
  "errorMessage" : "graph [demo] is not found",
  "errorCode" : "GES.8012"
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.2.10 Upgrading a Graph (1.0.5)

Function

This API is used to upgrade a graph. The GES version is periodically upgraded. You can upgrade your graphs as required.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/action

Table 11-58 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 11-59 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID Set this parameter to the enumeration value upgrade , which means that the graph will be upgraded.

Request Parameters

Table 11-60 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 11-61 Request body parameters

Parameter	Mandatory	Type	Description
upgradeVersion	Yes	String	Target version, which must be later than the current version

Parameter	Mandatory	Type	Description
forceUpgrade	No	Boolean	Whether to upgrade forcibly. The value is true or false, and the default value is false. • true: forcible upgrades, which will interrupt running tasks, such as long algorithm execution tasks. As a result, a small number of requests may fail. • false: non-forcible upgrades, which will wait for running services to complete. The upgrade process may be slow.

Response Parameters

Status code: 200

Table 11-62 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
jobId	String	ID of an asynchronous job NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs.

Example Request

```
POST http://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/action?action_id=upgrade
{
  "upgradeVersion" : "1.1.8",
```

```
{  
  "forceUpgrade" : false  
}
```

Example Response

Status code: 200

OK

```
{  
  "jobId" : "f99f60f1-bba6-4cde-bd1a-ff4bdd1fd500000168232"  
}
```

Status code: 400

Bad Request

```
{  
  "errorMessage" : "graph [demo] is not found",  
  "errorCode" : "GES.8011"  
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.2.11 Binding an EIP (1.0.6)

Function

This API enables you to access GES by binding an elastic IP (EIP).

URI

POST /v1.0/{project_id}/graphs/{graph_id}/action

Table 11-63 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 11-64 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID Set this parameter to bindEip , which means that an EIP will be bound to the target graph.

Request Parameters

Table 11-65 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 11-66 Request body parameters

Parameter	Mandatory	Type	Description
eipId	Yes	String	ID of the elastic IP address For details about how to query the EIP ID, see Querying an EIP .

Response Parameters

Status code: 200

Table 11-67 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.

Example Request

```
POST http://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/action?action_id=bindEip
{
  "eipId" : "02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042"
}
```

Example Response

```
Status code: 200
OK
{}

Status code: 400
Bad Request
{
  "errorMessage" : "graph [demo] is not found",
  "errorCode" : "GES.8011"
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.

Status Code	Description
503	Service unavailable.

11.1.2.12 Unbinding an EIP (1.0.6)

Function

If you do not need to use the EIP, this API enabled you to unbind the EIP to release network resources.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/action

Table 11-68 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 11-69 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID Set this parameter to unbindEip , which means that the EIP will be unbound from the target graph.

Request Parameters

Table 11-70 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 11-71 Request body parameters

Parameter	Mandatory	Type	Description
eipId	Yes	String	ID of the elastic IP address. For details about how to query the EIP ID, see Querying an EIP .

Response Parameters

Status code: 200

Table 11-72 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.

Example Request

```
POST http://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/action?action_id=unbindEip
```

```
{
  "eipId" : "02bd6dc1-5be8-430e-a4cd-2b0f6d0bb042"
}
```

Example Responses

Status code: 200

OK

```
{ }
```

Status code: 400

Bad Request

```
{
  "errorMessage" : "graph [demo] is not found",
  "errorCode" : "GES.8011"
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.2.13 Resizing a Graph (2.2.21)

Function

This API is used to resize a graph instance.

 NOTE

After the graph is resized, you need to re-create all indexes including composite indexes and full-text indexes.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/resize

Table 11-73 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 11-74 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 11-75 Request body parameters

Parameter	Mandatory	Type	Description
resize	Yes	GraphSizeTypeIndexReq object	Graph specifications after the graph is resized.

Table 11-76 GraphSizeTypeIndexReq

Parameter	Mandatory	Type	Description
graphSizeTypeIndex	Yes	String	Graph flavor. Currently, the value can be 2 , 3 , 4 , or 5 , indicating 10-million-edge , 100-million-edge , 1-billion-edge , or 10-billion-edge , respectively. (graph_size_type_index)

Response Parameters

Status code: 200

Table 11-77 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	Indicates the ID of the resize job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Job Management APIs .

Example Request

```
POST http://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/resize
{
  "resize" : {
    "graphSizeTypeIndex" : "2"
  }
}
```

Example Response

Status code: 200

```
OK
{ }
```

Status code: 400

```
Bad Request
{
  "errorCode" : "GES.7001",
  "errorMessage" : "The graph is not running."
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.2.14 Forcibly Restarting a Graph (2.2.21)

Function

This API is used to forcibly start a graph in the importing, exporting, running, or clearing state. If a graph is forcibly restarted, asynchronous tasks of the graph are failed state and the graph is stopped and started.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/action

Table 11-78 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 11-79 Query parameters

Parameter	Mandatory	Type	Description
action_id	Yes	String	Graph action ID Set this parameter to restart , which means that the target graph will be restarted.

Request Parameters

Table 11-80 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-81 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
jobId	String	ID of a forcible restart job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

Example Request

```
POST https://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/action?action_id=restart
```

Example Response

Status code: 200

OK

```
{ }
```

Status code: 400

Bad Request

```
{
  "errorMessage" : "The request is invalid.",
  "errorCode" : "GES.7016"
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.2.15 Expanding a Graph (2.2.23)

Function

This API is used to expand multiple secondary nodes dynamically. The expanded secondary nodes can process read requests, improving read performance.

 NOTE

- 1. This API is not supported by graphs of the 10,000-edge and 10-billion-edge types.
- 2. Graphs cannot be resized after expansion.
- 3. If you want to resize and expand the graph, resize the graph before you expand it.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/expand

Table 11-82 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 11-83 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 11-84 Request body parameters

Parameter	Mandatory	Type	Description
expand	Yes	ReplicationReq object	Expansion information

Table 11-85 ReplicationReq

Parameter	Mandatory	Type	Description
replication	Yes	String	Number of new nodes to expand

Response Parameters

Status code: 200

Table 11-86 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
jobId	String	ID of the expansion job. This parameter is left blank when the request fails. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

Example Request

```
POST http://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/expand
{
  "expand": {
    "replication": "1"
  }
}
```

Example Responses

Status code: 200

OK

```
{
  "jobId": "ff8080816025a0a1016025a5a2700007"
}
```

Status code: 400

Bad Request

```
{
  "errorCode": "GES.7015",
  "errorMessage": "The graph is not running or stopped."
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.3 Backup Management APIs

11.1.3.1 Viewing the List of All Backups (1.0.0)

Function

This API is used to query the list of all backups.

URI

GET /v1.0/{project_id}/graphs/backups

Table 11-87 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Table 11-88 Query parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of resources displayed on a single page. The default value is 10 .
offset	No	Integer	Start position of the request. The default value is 0 .

Request Parameters

Table 11-89 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200**Table 11-90** Response body parameter

Parameter	Type	Description
errorMessage	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
backupCount	Integer	Total number of backups. This parameter is left blank if the request fails.
backupList	Array of backup objects	List of all backups under the current project ID. This parameter is left blank if the request fails.

Table 11-91 backup

Parameter	Type	Description
id	String	Backup ID
name	String	Backup name

Parameter	Type	Description
backupMethod	String	Backup method. The value can be auto or manual .
graphId	String	ID of the graph associated with the backup
graph_name	String	Name of the graph associated with the backup
graphStatus	String	Status of the graph associated with the backup
graphSizeTypeIndex	String	Size of the graph associated with the backup
dataStoreVersion	String	Version of the graph associated with the backup
arch	String	CPU architecture of the graph node associated with the backup
status	String	Backup status: • backing_up: indicates that a graph is being backed up. • success: indicates that a graph is successfully backed up. • failed: indicates that a graph fails to be backed up.
startTimestamp	Long	Start timestamp of a backup job
startTime	String	Backup start time
endTimestamp	Long	End timestamp of a backup job
endTime	String	Backup end time
size	Long	Backup file size (MB)
duration	Long	Backup duration (seconds)

Example Request

```
GET https://{Endpoint}/v1.0/{project_id}/graphs/backups?offset=0&limit=2
```

Example Response

Example response with status code **200**:

OK

```
{
  "backupCount": 3,
  "backupList": [
    {
      "id": "ada3e720-ab87-48cb-bff7-3ec5ae1a9652",
```

```
{
  "name": "ges060803_nodelete-20210608135513",
  "backupMethod": "manual",
  "graphId": "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
  "graph_name": "ges060803_nodelete",
  "graphStatus": "200",
  "graphSizeTypeIndex": "1",
  "dataStoreVersion": "2.2.21",
  "arch": "x86_64",
  "status": "success",
  "startTimestamp": 1623160513000,
  "startTime": "2021-06-08T13:55:13",
  "endTimestamp": 1623160568000,
  "endTime": "2021-06-08T13:56:08",
  "size": 1,
  "duration": 54,
},
{
  "id": "7ed3f51d-816d-4651-9129-fe21b64b5c91",
  "name": "ges060803_nodelete_20210609203323_auto",
  "backupMethod": "auto",
  "graphId": "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
  "graph_name": "ges060803_nodelete",
  "graphStatus": "200",
  "graphSizeTypeIndex": "1",
  "dataStoreVersion": "2.2.21",
  "arch": "x86_64",
  "status": "success",
  "startTimestamp": 1623242004000,
  "startTime": "2021-06-09T12:33:24",
  "endTimestamp": 1623242004000,
  "endTime": "2021-06-09T12:33:24",
  "size": 1,
  "duration": 0
},
{
  "id": "604bfb46-04dd-45fc-a9ae-df24a0705b9d",
  "name": "ges060802_nodelete-20210608135523",
  "backupMethod": "manual",
  "graphId": "9b9a05c2-0cdb-41ac-b55f-93caffb0519a",
  "graph_name": "ges060802_nodelete",
  "graphStatus": "400",
  "graphSizeTypeIndex": "0",
  "dataStoreVersion": "2.2.23",
  "arch": "x86_64",
  "status": "success",
  "startTimestamp": 1623160524000,
  "startTime": "2021-06-08T13:55:24",
  "endTimestamp": 1623160577000,
  "endTime": "2021-06-08T13:56:17",
  "size": 1,
  "duration": 53
}
]
```

Status code: 400**Bad Request**

```
{
  "errorCode": "GES.7006",
  "errorMessage": "The underlying graph engine has internal error."
}
```


Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.3.2 Viewing the Backup List of a Graph (1.0.0)

Function

This API is used to query the backup list of a graph.

URI

GET /v1.0/{project_id}/graphs/{graph_id}/backups

Table 11-92 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Table 11-93 Query parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of resources displayed on a single page. The default value is 10 .
offset	No	Integer	Start position of the request. The default value is 0 .

Request Parameters

Table 11-94 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-95 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
backupCount	Integer	Total number of backups. This parameter is left blank if the request fails.
backupList	Array of backup objects	List of backups of a specified graph under the current project. This parameter is left blank if the request fails.

Table 11-96 backup

Parameter	Type	Description
id	String	Backup ID
name	String	Backup name

Parameter	Type	Description
backupMethod	String	Backup method. The value can be auto or manual .
graphId	String	ID of the graph associated with the backup
graph_name	String	Name of the graph associated with the backup
graphStatus	String	Status of the graph associated with the backup
graphSizeTypeIndex	String	Size of the graph associated with the backup
dataStoreVersion	String	Version of the graph associated with the backup
arch	String	CPU architecture of the graph node associated with the backup
status	String	Backup status: • backing_up: indicates that a graph is being backed up. • success: indicates that a graph is successfully backed up. • failed: indicates that a graph fails to be backed up.
startTimestamp	Long	Start timestamp of a backup job
startTime	String	Backup start time
endTimestamp	Long	End timestamp of a backup job
endTime	String	Backup end time
size	Long	Backup file size (MB)
duration	Long	Backup duration (seconds)

Example Request

```
GET https://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/backups?offset=0&limit=2
```

Example Response

Example response with status code **200**:

Example response for a successful request

```
{
  "backupCount": 2,
  "backupList": [
    {
      "id": "ada3e720-ab87-48cb-bff7-3ec5ae1a9652",
```

```
    "name": "ges060803_nodelete-20210608135513",
    "backupMethod": "manual",
    "graphId": "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
    "graph_name": "ges060803_nodelete",
    "graphStatus": "200",
    "graphSizeTypeIndex": "1",
    "dataStoreVersion": "2.2.22",
    "arch": "x86_64",
    "status": "success",
    "startTimestamp": 1623160513000,
    "startTime": "2021-06-08T13:55:13",
    "endTimestamp": 1623160568000,
    "endTime": "2021-06-08T13:56:08",
    "size": 1,
    "duration": 54
  },
  {
    "id": "7ed3f51d-816d-4651-9129-fe21b64b5c91",
    "name": "ges060803_nodelete_20210609203323_auto",
    "backupMethod": "auto",
    "graphId": "4c5f882d-a813-4d78-a8e3-6d3212ddd121",
    "graph_name": "ges060803_nodelete",
    "graphStatus": "200",
    "graphSizeTypeIndex": "1",
    "dataStoreVersion": "2.2.21",
    "arch": "x86_64",
    "status": "success",
    "startTimestamp": 1623242004000,
    "startTime": "2021-06-09T12:33:24",
    "endTimestamp": 1623242004000,
    "endTime": "2021-06-09T12:33:24",
    "size": 1,
    "duration": 0
  }
]
```

Status code: 404

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
  "error_code": "GES.0047"
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.3.3 Adding a Backup (1.0.0)

Function

This API is used to add a backup. If data in the current graph is incorrect or faulty, you can start the backup graph to restore the data.

URI

POST /v1.0/{project_id}/graphs/{graph_id}/backups

Table 11-97 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID

Request Parameters

Table 11-98 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-99 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.
jobId	String	ID of the graph backup job. NOTE You can view the job execution status and obtain the return result by querying the job ID. For details, see Task Center APIs .

Example Request

POST https://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/backups

Example Response

Example response with status code **200**:
Example response for a successful request

```
{
  "jobId" : "ff8080815f9a3c84015f9a438ff70001"
}
```

Status code: **404**

Example response for a failed request

```
{
  "error_msg": "Cluster does not exist or has been deleted.",
  "error_code": "GES.0047"
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.

Status Code	Description
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.3.4 Deleting a Backup (1.0.0)

Function

This API is used to delete a backup.

URI

DELETE /v1.0/{project_id}/graphs/{graph_id}/backups/{backup_id}

Table 11-100 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
backup_id	Yes	String	Graph backup ID
graph_id	Yes	String	Graph ID

Request Parameters

Table 11-101 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-102 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. • If execution succeeds, this parameter may be left blank. • If execution fails, this parameter is used to display the error code.

Example Request

DELETE https://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/backups/{backupId}

Example Response

Status code: 200

OK

```
{ }
```

Status code: 400

Bad Request

```
{
  "errorMessage" : "Parameter error!",
  "errorCode" : "GES.0001"
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.4 Metadata Management APIs

11.1.4.1 Constraints

[Table 11-103](#) and [Table 11-104](#) list the metadata types.

Table 11-103 Metadata property constraints

Data Type	Constraints
char	• Less than (<) • Greater than (>) • Equal to (=) • Not equal to '!=' • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
char array	• Less than (<) • Greater than (>) • Equal to (=) • Not equal to '!=' • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
float	• Less than (<) • Greater than (>) • Equal to (=) • Not equal to '!=' • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
double	• Less than (<) • Greater than (>) • Equal to (=) • Not equal to '!=' • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
bool	• Equal to (=) • Not equal to '!='

Data Type	Constraints
long	• Less than (<) • Greater than (>) • Equal to (=) • Not equal to '!=' • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
int	• Less than (<) • Greater than (>) • Equal to (=) • Not equal to '!=' • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
date	• Less than (<) • Greater than (>) • Equal to (=) • Not equal to '!=' • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)
enum	• Equal to (=) • Not equal to '!='
string	• Less than (<) • Greater than (>) • Equal to (=) • Not equal to '!=' • In range (range) • Greater than or equal to (>=) • Less than or equal to (<=)

Table 11-104 Property-level constraints

Property Level	Constraints	Description
Single value/ Multiple values	has	This property is contained.

Property Level	Constraints	Description
Single value/ Multiple values	hasNot	This property is not contained.

11.1.4.2 Listing Metadata Files (1.0.2)

Function

This API is used to list metadata files.

URI

GET /v1.0/{project_id}/graphs/metadatas

Table 11-105 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Table 11-106 Query parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Maximum number of resources displayed on a single page. The default value is 10 .
offset	No	Integer	Start position of the request. The default value is 0 .

Request Parameters

Table 11-107 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-108 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
schemaCount	Integer	Number of returned metadata files. This parameter is left blank if the request fails.
schemaList	Array of metadata objects	List of all metadata files under the current project ID. This parameter is left blank if the request fails.

Table 11-109 metadata

Parameter	Type	Description
id	String	Metadata ID
name	String	Metadata name

Parameter	Type	Description
description	String	Metadata description
status	String	Whether the metadata is available
metadataPath	String	Metadata path
startTime	String	Metadata creation time
lastUpdateTime	String	Last update time of the metadata

Example Request

GET https://{Endpoint}/v1.0/{project_id}/graphs/metadatas?offset=10&limit=100

Example Response

Status code: 200

OK

```
{
  "schemaCount": 1,
  "schemaList": [ {
    "id": "ff7dddc4-6402-43d7-9aed-c5ec677b47fa",
    "name": "schema_demo",
    "description": "",
    "status": "200",
    "metadataPath": "ges-graphs/demo_movie/schema.xml",
    "startTime": "2018-07-23T02:59:41",
    "lastUpdateTime": "2018-07-23T02:59:41"
  } ]
}
```

Status code: 500

Internal Server Error

```
{
  "errorCode": "GES.7006",
  "errorMessage": "The underlying graph engine has internal error."
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.

Status Code	Description
500	Internal service error.
503	Service unavailable.

11.1.4.3 Querying Metadata (1.0.2)

Function

This API is used to query a specific metadata file.

URI

GET /v1.0/{project_id}/graphs/metadatas/{metadata_id}

Table 11-110 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
metadata_id	Yes	String	Metadata ID

Request Parameters

Table 11-111 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-112 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
gesMetadata	GesMetaDat a object	Object for storing metadata message information.

Table 11-113 GesMetaData

Parameter	Type	Description
labels	Array of Label objects	Label data structure set

Table 11-114 Label

Parameter	Type	Description
name	String	Label name
properties	Object	Maps of metadata properties. The key-value pairs are those in the imported metadata.

Example Request

```
GET https://{Endpoint}/v1.0/{project_id}/graphs/metadatas/{metadata_id}
```

Example Response

Example response with status code **200**:

OK

```
{
  "gesMetadata": {
    "labels": [
      {
        "name": "friends",
        "properties": null
      }
    ]
  }
}
```

```
},
{
  "name": "movie",
  "properties": [
    {
      "dataType": "string",
      "name": "ChineseTitle",
      "cardinality": "single"
    },
    {
      "dataType": "int",
      "name": "Year",
      "cardinality": "single"
    },
    {
      "dataType": "string",
      "name": "Genres",
      "cardinality": "set"
    }
  ]
},
{
  "name": "user",
  "properties": [
    {
      "dataType": "string",
      "name": "ChineseTitle",
      "cardinality": "single"
    },
    {
      "dataType": "int",
      "name": "Year",
      "cardinality": "single"
    },
    {
      "dataType": "string",
      "name": "Genres",
      "cardinality": "set"
    },
    {
      "typeName1": "F",
      "typeName2": "M",
      "typeNameCount": "2",
      "dataType": "enum",
      "name": "Gender",
      "cardinality": "single"
    },
    {
      "typeName1": "Under 18",
      "typeName2": "18-24",
      "typeName3": "25-34",
      "typeName4": "35-44",
      "typeNameCount": "7",
      "dataType": "enum",
      "name": "Age",
      "typeName5": "45-49",
      "typeName6": "50-55",
      "cardinality": "single",
      "typeName7": "56+"
    },
    {
      "dataType": "string",
      "name": "Occupation",
      "cardinality": "single"
    },
    {
      "dataType": "char array",
      "name": "Zip-code",
      "maxDataSize": "12",
      "cardinality": "single"
    }
  ]
}
```



```
    }  
  ],  
  {  
    "name": "rate",  
    "properties": [  
      {  
        "dataType": "int",  
        "name": "Score",  
        "cardinality": "single"  
      },  
      {  
        "dataType": "date",  
        "name": "Datetime",  
        "cardinality": "single"  
      }  
    ]  
  }  
]
```

Status code: 500

Internal Server Error

```
{  
  "errorCode" : "GES.7006",  
  "errorMessage" : "The underlying graph engine has internal error."  
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.4.4 Adding Metadata (2.1.18)

Function

This API is used to add the metadata.

URI

POST /v1.0/{project_id}/graphs/metadatas

Table 11-115 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Request Parameters

Table 11-116 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 11-117 Request body parameters

Parameter	Mandatory	Type	Description
metadataPath	Yes	String	Metadata storage address
name	Yes	String	Metadata name, which contains 1 to 64 characters consisting of only letters, digits, and underscores (_)
description	Yes	String	Metadata description
isOverwrite	Yes	Boolean	Whether to overwrite existing files. The value can be true or false . The default value is false . • true: Existing file will be overwritten. • false: Existing file will not be overwritten.
gesMetadata	Yes	Object	Object for storing metadata message information.

Response Parameters

Status code: 200

Table 11-118 Response body parameter

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
id	String	Metadata ID
name	String	Metadata name

Example Request

```
POST https://{Endpoint}/v1.0/{project_id}/graphs/metadatas
{
  "metadataPath": "gesdata/demo_movie/schema.xml",
  "name": "movie_schema",
  "description": "xxxxx",
  "isOverwrite": "true",
  "gesMetadata": {
    "labels": [
      {
        "name": "friends",
        "properties": null
      },
      {
        "name": "movie",
        "properties": [
          {
            "dataType": "string",
            "name": "ChineseTitle",
            "cardinality": "single"
          },
          {
            "dataType": "int",
            "name": "Year",
            "cardinality": "single"
          },
          {
            "dataType": "string",
            "name": "Genres",
            "cardinality": "set"
          }
        ]
      }
    ]
  },
  "name": "user",
```

```
    "properties": [
      {
        "dataType": "string",
        "name": "Name",
        "cardinality": "single"
      },
      {
        "typeName1": "F",
        "typeName2": "M",
        "typeNameCount": "2",
        "dataType": "enum",
        "name": "Gender",
        "cardinality": "single"
      },
      {
        "typeName1": "Under 18",
        "typeName2": "18-24",
        "typeName3": "25-34",
        "typeName4": "35-44",
        "typeNameCount": "7",
        "dataType": "enum",
        "name": "Age",
        "typeName5": "45-49",
        "typeName6": "50-55",
        "cardinality": "single",
        "typeName7": "56+"
      },
      {
        "dataType": "string",
        "name": "Occupation",
        "cardinality": "single"
      },
      {
        "dataType": "char array",
        "name": "Zip-code",
        "maxDataSize": "12",
        "cardinality": "single"
      }
    ]
  },
  {
    "name": "rate",
    "properties": [
      {
        "dataType": "int",
        "name": "Score",
        "cardinality": "single"
      },
      {
        "dataType": "date",
        "name": "Datetime",
        "cardinality": "single"
      }
    ]
  }
]
}
```

Example Response

Example response with status code **200**:

OK

```
{
  "id" : "ff8080815f9a3c84015f9a438ff70001",
  "name" : "movie_schema"
}
```

Status code: 500

Internal Server Error

```
{
  "errorCode" : "GES.2067",
  "errorMessage" : "name: 1 to 64 characters, only letters, digits, and underscores(_) are allowed."
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.4.5 Deleting Metadata (1.0.2)

Function

This API is used to delete the metadata.

URI

DELETE /v1.0/{project_id}/graphs/metadatas/{metadata_id}

Table 11-119 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
metadata_id	Yes	String	Metadata ID

Request Parameters

Table 11-120 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-121 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.

Example Request

```
DELETE https://{Endpoint}/v1.0/{project_id}/graphs/metadatas/{metadata_id}
```

Example Response

Status code: 200

```
OK
{ }
```

Status code: 400

Bad Request

```
{
  "errorCode": "GES.7024",
  "errorMessage": "The metadata is not exist or has been deleted."
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.4.6 Importing Metadata from OBS (1.0.0)

Function

This API is used to import metadata from OBS.

URI

POST /v1.0/{project_id}/graphs/metadata/upload_from_obs

Table 11-122 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Request Parameters

Table 11-123 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Table 11-124 Request body parameters

Parameter	Mandatory	Type	Description
metadataPath	Yes	String	Path for storing the metadata
name	Yes	String	Metadata name
description	No	String	Metadata description

Response Parameters

Status code: 200**Table 11-125** Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt code. - If execution succeeds, this parameter may be left blank. - If execution fails, this parameter is used to display the error code.
id	String	Metadata ID
name	String	Metadata name

Example Request

```
POST https://{Endpoint}/v1.0/{project_id}/graphs/metadata/upload_from_obs
{
  "metadataPath": "devdata/unionsdk/schema.xml",
  "name": "test_schema",
  "description": ""
}
```

Example Response

Example response with status code **200**:

OK

```
{
  "id" : "d30d2e94-f2ee-4344-af49-eb27fd002eea",
  "name" : "test_schema"
}
```

Status code: 404

Internal Server Error

```
{
  "errorCode": "GES.0016",
  "errorMessage": "Resource not found"
}
```

Status Code

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permission.
404	No resources found.
500	Internal service error.
503	Service unavailable.

11.1.5 Task Center APIs

11.1.5.1 Querying Job Status on the Management Plane (1.0.0)

Function

This API is used to query the execution status of a job. Asynchronous APIs that are used to create, stop, start, delete, and import graphs will return job IDs after commands are sent. You can query the job execution status according to the job IDs.

URI

GET /v1.0/{project_id}/graphs/{graph_id}/jobs/{job_id}/status

Table 11-126 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_id	Yes	String	Graph ID
job_id	Yes	String	ID of the asynchronous job

Request Parameters

Table 11-127 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-128 Response body parameters

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.

Parameter	Type	Description
errorCode	String	System prompt. • If the execution succeeds, this parameter may be left blank. • If the execution fails, this parameter is used to display the error code.
jobId	String	Job ID
status	String	Job status. The options are: • pending • running • success • partiallyFailed • failed
jobType	String	Job type
jobName	String	Job name
relatedGraph	String	Associated graph name
beginTime	String	Job start time (UTC). The format is (yyyy-MM-dd HH:mm:ss).
endTime	String	Job end time (UTC). The format is (yyyy-MM-dd HH:mm:ss).
jobDetail	JobDetail object	This parameter is returned only when jobName is set to ImportGraph and is used to display graph import details.
failReason	String	Job failure cause
jobProgress	Double	Job execution progress. It is a reserved field, and not used currently.

Table 11-129 JobDetail

Parameter	Type	Description
schemaPath	Array of schemaPath objects	Path for storing metadata
edgesetPath	Array of edgesetPath objects	Path for storing the edge data set

Parameter	Type	Description
vertexsetPath	Array of vertexsetPath objects	Path for storing the vertex data set

Table 11-130 schemaPath

Parameter	Type	Description
path	String	OBS storage path
log	String	Import log
status	String	OBS file status: • success: Imported successfully. • Failed: Failed to import the file. • partFailed: Partially failed.
cause	String	Import failure cause

Table 11-131 edgesetPath

Parameter	Type	Description
path	String	OBS storage path
log	String	Import log
status	String	OBS file status: • success: Imported successfully. • Failed: Failed to import the file. • partFailed: Partially failed.
cause	String	Import failure cause
totalLines	Long	The total number of imported lines, which is subject to your quota and the specifications of the created graph. The value -1 indicates that this field is not returned in the current version.
failedLines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successfulLines	Long	Lines imported successfully. The value -1 indicates that this field is not returned in the current version.

Table 11-132 vertexsetPath

Parameter	Type	Description
path	String	OBS storage path
log	String	Import log
status	String	OBS file status: ● success: Imported successfully. ● Failed: Failed to import the file. ● partFailed: Partially failed.
cause	String	Import failure cause
totalLines	Long	Total number of imported lines. The value is subject to your quota and specifications of the created graph. If the value is -1 , this parameter is not supported by the current version.
failedLines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successfulLines	Long	Lines imported successfully. The value -1 indicates that this field is not returned in the current version.

Example Request

GET https://{Endpoint}/v1.0/{project_id}/graphs/{graph_id}/jobs/{job_id}/status

Example Response

Status code: 200

OK

```
Http Status Code: 200
{
  "jobId": "ff80808167f09aaa0167f19b35ec0305",
  "status": "success",
  "jobType": "GraphManagement",
  "jobName": "ImportGraph",
  "relatedGraph": "GES_UI_AUTO",
  "beginTime": "2018-11-27T21:39:00",
  "endTime": "2018-11-27T21:39:56",
  "jobDetail": {
    "vertexsetPath": [
      {
        "path": "ges-ui/auDatas/list_set_vertex.csv",
        "log": null,
        "cause": null,
        "status": "success"
      }
    ],
    "edgesetPath": [
      {
        "path": "ges-ui/auDatas/list_set_edge.csv",
```

```
{
  "log": null,
  "cause": null,
  "status": "success"
},
"schemaPath": [
  {
    "path": "ges-ui/auDatas/list_set_schema.xml",
    "log": null,
    "cause": null,
    "status": "success"
  }
],
"jobProgress": 0
}
```

Status code: 400

Bad Request

```
{
  "errorMessage": "can not find job, jobId is ff808081646e81d40164c5fb414b2b1a1",
  "errorCode": "GES.8301"
}
```

Status Codes

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permissions.
404	No resources found.
500	Internal server error.
503	Service unavailable.

11.1.5.2 Querying Job Details in the Job Center (1.1.8)**Function**

This API is used to query asynchronous job details in the job center on the management plane. Asynchronous jobs include creating, closing, starting, deleting, adding, importing, exporting, and upgrading graphs, as well as adding backups.

URI

GET /v1.0/{project_id}/graphs/jobs

Table 11-133 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .

Table 11-134 Query parameters

Parameter	Mandatory	Type	Description
endTime	No	String	Job end date. Currently, only the date is supported. The format is <i>yyyy-MM-dd</i> , for example, 2019-03-27.
graph_name	No	String	Associated graph name
limit	No	String	Maximum number of resources displayed on a single page. The default value is 10 .
offset	No	String	Start position of the request. The default value is 0 .
startTime	No	String	Job start date. Currently, only the date is supported. The format is <i>yyyy-MM-dd</i> , for example, 2019-03-27.
status	No	String	Job status. The options are: • running • waiting • success • partiallyFailed • failed

Request Parameters

Table 11-135 Parameters in the request header

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	User token. It is used to obtain the permission to call APIs. For details about how to obtain the token, see Authentication . The value of X-Subject-Token in the response header is the token.

Response Parameters

Status code: 200

Table 11-136 Response body parameter

Parameter	Type	Description
errorMessage	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. - If the execution succeeds, this parameter may be left blank. - If the execution fails, this parameter is used to display the error code.
jobCount	Integer	Total number of jobs
jobList	Array of Job objects	Task list

Table 11-137 Job

Parameter	Type	Description
jobId	String	Job ID

Parameter	Type	Description
status	String	Job status. The options are: • pending • running • success • partiallyFailed • failed
jobType	String	Task type
jobName	String	Task name
relatedGraph	String	Associated graph name
beginTime	String	Job start time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.
endTime	String	Job end time (UTC). The format is yyyy-MM-dd'T'HH:mm:ss.
jobDetail	JobDetail object	This parameter is returned only when jobName is set to ImportGraph and is used to display graph import details.
failReason	String	Job failure cause
jobProgress	Double	Job execution progress. It is a reserved field, and not used currently.

Table 11-138 JobDetail

Parameter	Type	Description
schemaPath	Array of schemaPath objects	Path for storing metadata
edgesetPath	Array of edgesetPath objects	Path for storing the edge data set
vertexsetPath	Array of vertexsetPath objects	Path for storing the vertex data set

Table 11-139 schemaPath

Parameter	Type	Description
path	String	OBS storage path

Parameter	Type	Description
log	String	Import log
status	String	OBS file status: • success: Imported successfully. • Failed: Failed to import the file. • partFailed: Partially failed.
cause	String	Import failure cause

Table 11-140 edgesetPath

Parameter	Type	Description
path	String	OBS storage path
log	String	Import log
status	String	OBS file status: • success: Imported successfully. • Failed: Failed to import the file. • partFailed: Partially failed.
cause	String	Import failure cause
totalLines	Long	Total number of imported lines. The value -1 indicates that this field is not returned in the current version.
failedLines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successfulLines	Long	Lines imported successfully. The value -1 indicates that this field is not returned in the current version.

Table 11-141 vertexsetPath

Parameter	Type	Description
path	String	OBS storage path
log	String	Import log

Parameter	Type	Description
status	String	OBS file status: ● success: Imported successfully. ● Failed: Failed to import the file. ● partFailed: Partially failed.
cause	String	Import failure cause
totalLines	Long	Total number of imported lines. The value -1 indicates that this field is not returned in the current version.
failedLines	Long	Lines failed to be imported. The value -1 indicates that this field is not returned in the current version.
successfulLines	Long	Lines imported successfully. The value -1 indicates that this field is not returned in the current version.

Example Request

```
GET https://{Endpoint}/v1.0/{project_id}/graphs/jobs?offset=0&limit=100
```

Example Response

Example response with status code **200**:

OK

```
Http Status Code: 200
{
  "jobCount": 136,
  "jobList": [
    {
      "jobId": "ff80808167bb90340167bc3c7b5b026a",
      "status": "success",
      "jobType": "GraphManagement",
      "jobName": "ImportGraph",
      "relatedGraph": "test1217",
      "beginTime": "2018-12-17T12:55:40",
      "endTime": "2018-12-17T12:56:32",
      "jobDetail": {
        "vertexsetPath": null,
        "edgesetPath": [
          {
            "path": "hkmovie/edge.csv",
            "log": null,
            "cause": null,
            "status": "success"
          }
        ],
        "schemaPath": [
          {
            "path": "hkmovie/schema.xml",
            "log": null,
            "cause": null,
            "status": "success"
          }
        ]
      }
    }
  ]
}
```

```
]
},
"jobProgress": 0
},
{
  "jobId": "ff80808167bb90340167bc5d0b1d0358",
  "status": "success",
  "jobType": "GraphManagement",
  "jobName": "DeleteGraph",
  "relatedGraph": "test1218",
  "beginTime": "2018-12-17T13:31:14",
  "endTime": "2018-12-17T13:34:48",
  "jobProgress": 0
}
]
```

Status code: 400

Bad Request

```
{
  "errorMessage": "failed",
  "errorCode": "GES.9999"
}
```

Status Codes

Status Code	Description
200	Request sent.
400	Request error.
401	Authentication failed.
403	No operation permissions.
404	No resources found.
500	Internal server error.
503	Service unavailable.

11.2 Service Plane APIs

11.2.1 Vertex Operation APIs

11.2.1.1 Batch Adding Vertices (2.1.9)

Function

This API is used to add vertices in batches.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-add
- Parameter description

Table 11-142 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example
POST
http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/vertices/action?action_id=batch-add

```
{
  "vertices": [
    {
      "vertex": "150",
      "label": "movie",
      "properties": {
        "movieid": [
          "150"
        ],
        "title": [
          "testmoive"
        ],
        "genres": [
          "Comedy"
        ]
      }
    },
    {
      "vertex": "6",
      "label": "movie",
      "properties": {
        "movieid": [
          "6"
        ],
        "title": [
          "testmoive_exist_id"
        ],
        "genres": [
          "Comedy"
        ]
      }
    }
  ]
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

- Parameter description

Table 11-143 Request body parameter

Parameter	Mandatory	Type	Description
vertices	Yes	Json	Vertex array to be added

Table 11-144 vertices parameter description

Parameter	Mandatory	Type	Description
vertex	Yes	String	Vertex ID
label	Yes	String	Vertex label
properties	No	Json	Value of each property

Response

- Parameter description

Table 11-145 Parameter description

Parameter	Mandatory	Type	Description
errorMessage	No	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.
errorCode	No	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

Http Status Code: 200

```
{  
  "result": "success"  
}
```

- Response example (failed request)

Http Status Code: 400

```
{  
  "errorMessage": "vertex [Lily] already exists",  
  "errorCode": "GES.8000"  
}
```

Return Value

- Normal
200
- Abnormal

Table 11-146 Return code for failed requests

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

11.2.2 Edge Operation APIs

11.2.2.1 Adding an Edge (1.0.6)

Function

This API is used to add an edge.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges
- Parameter description

Table 11-147 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example

POST `http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges`

```
{
  "source": "Lily",
  "target": "Rocky",
  "label": "rate",
  "properties": { "Score": [5], "Datetime": ["2018-01-01 20:30:05"] },
  "parallelEdge": "override"
}
```

 NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

- Request body parameters

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex name
target	Yes	String	Target vertex name
label	Yes	String	Label of a vertex. If no label exists, set it to __DEFAULT__ .
properties	No	Json	Value of each property
parallelEdge	No	String	Processing mode of repetitive edges. The value can be allow , ignore , or override . allow indicates that repetitive edges are allowed. ignore indicates that subsequent repetitive edges are ignored. override indicates that the previous repetitive edges are overwritten. The default value is allow .

Response

- Parameter description

Parameter	Type	Description
errorMessage	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.
errorCode	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.

Parameter	Type	Description
data	String	This data field is included for a successful query, which contains the edge query result.
result	String	If the execution is successful, the value of result is success .

- Response example (successful request)

Http Status Code: 200

```
{
  "result": "success",
  "data": { "index": "0" }
}
```

- Response example (failed request)

Http Status Code: 400

```
{
  "errorMessage": "Edge source vertex [Lily] does not exist",
  "errorCode": "GES.8000"
}
```

Return Value

- Normal
200
- Abnormal

Table 11-148 Return code for failed requests

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

11.2.2.2 Batch Adding Edges (2.1.9)

Function

This API is used to add edges in batches.

URI

- URI format
POST /ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add
- Parameter description

Table 11-149 URI parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see Obtaining a Project ID .
graph_name	Yes	String	Graph name

Request

- Request example

POST `http://{SERVER_URL}/ges/v1.0/{project_id}/graphs/{graph_name}/edges/action?action_id=batch-add`

```
{
  "edges": [
    {
      "source": "46",
      "target": "39",
      "label": "rate",
      "properties": {
        "Rating": [
          5
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    },
    {
      "source": "46",
      "target": "38",
      "label": "rate",
      "properties": {
        "Rating": [
          4
        ],
        "Datetime": [
          "2018-01-0120:30:05"
        ]
      }
    }
  ],
  "parallelEdge": {
    "action": "override",
    "ignoreLabel": true
  }
}
```

NOTE

SERVER_URL: Address for accessing a graph. For details about its value, see [Using Service Plane APIs](#).

- Request body parameters

Parameter	Mandatory	Type	Description
edges	Yes	Json	Edge array to be added

Parameter	Mandatory	Type	Description
parallelEdge	No	Object	Repetitive edge processing
action	No	String	Processing mode of repetitive edges. The value can be allow , ignore , or override . The default value is allow . • allow indicates that repetitive edges are allowed. • ignore indicates that subsequent repetitive edges are ignored. • override indicates that the previous repetitive edges are overwritten.
ignoreLabel	No	Boolean	Whether to ignore labels on repetitive edges. The value can be true or false . The default value is true . • true: Indicates that the repetitive edge definition does not contain the label. That is, the <source vertex, target vertex> indicates an edge, excluding the label information. • false: Indicates that the repetitive edge definition contains the label. That is, the <source vertex, target vertex, label> indicates an edge.

Table 11-150 edges parameter description

Parameter	Mandatory	Type	Description
source	Yes	String	Source vertex of an edge
target	Yes	String	Target vertex of an edge
label	Yes	String	Edge label
properties	No	Json	Value of each property

Response

- Parameter description

Parameter	Mandatory	Type	Description
errorMessage	No	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error message.
errorCode	No	String	System prompt. If execution succeeds, this parameter may be left blank. If execution fails, this parameter is used to display the error code.
result	No	String	If the execution is successful, the value of result is success .

- Response example (successful request)

```
Http Status Code: 200
{
  "result": "success",
  "data": {
    "edges": [
      {
        "index": "7",
        "source": "46",
        "target": "39"
      },
      {
        "index": "0",
        "source": "46",
        "target": "38"
      }
    ]
  }
}
```

- Response example (failed request)

```
Http Status Code: 400
{
  "errorMessage": "Edge source vertex [Lily] does not exist",
  "errorCode": "GES.8000"
}
```

Return Value

- Normal
200
- Abnormal

Table 11-151 Return code for failed requests

Return Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authorization failed.
403 Forbidden	No operation permissions.

Return Value	Description
404 Not Found	No resources found.
500 Internal Server Error	Internal server error.
503 Service Unavailable	Service unavailable.

11.2.3 Algorithm APIs

11.2.3.1 Shortest Paths (1.0.0)

Table 11-152 parameters parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
source	Yes	Source vertex ID of a path	String	-	-
target	Yes	Target vertex ID of a path	String	-	-
weight	No	Weight of an edge	String	Empty or character string - Empty: The default weight and distance of edges are 1. - Character string: The property of the corresponding edge is the weight. If the edge does not have a property, the weight is 1 by default. NOTE The weight of an edge must be greater than 0 .	-
directed	No	Whether to consider the edge direction	Boolean	true or false	false

Table 11-153 response_data parameter description

Parameter	Type	Description
path	List	Shortest path. The format is as follows: [vertexId,...] where vertexId is of the string type.
source	String	Source vertex ID
target	String	Target vertex ID

11.2.3.2 Shortest Path of Vertex Sets (1.0.0)

Table 11-154 Parameter description

Parameter	Mandatory	Description	Type	Value Range	Default Value
sources	Yes	Source vertex ID set	String	The value is in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana . The maximum ID number is 100000.	-
targets	Yes	Target vertex ID set	String	The value is in the standard CSV format. IDs are separated by commas (,), for example, Alice,Nana . The maximum ID number is 100000.	-
directed	No	Whether an edge is directed	Boolean	true or false	false

Table 11-155 response_data parameter description

Parameter	Type	Description
path	List	Shortest path. The format is as follows: [vertexId,...] where vertexId is of the string type.

Parameter	Type	Description
source	String	Source vertex ID
target	String	Target vertex ID

11.2.3.3 Label Propagation (1.0.0)

Table 11-156 parameters parameter descriptions

Parameter	Mandatory	Description	Type	Value Range	Default Value
convergence	No	Convergence	Double	The value is a real number between 0 and 1.	0.00001
max_iterations	No	Maximum iterations	Integer	1 to 2000	1000

Table 11-157 response_data parameter descriptions

Parameter	Type	Description
community	List	Community corresponding to each vertex. The format is as follows: [{vertexId:communityId},...] where vertexId : string type communityId : string type

11.2.3.4 Louvain (1.0.0)

Table 11-158 parameters parameter descriptions

Parameter	Mandatory	Description	Type	Value Range	Default Value
convergence	No	Convergence	Double	The value is a real number between 0 and 1.	0.00001
max_iterations	No	Maximum iterations	Integer	1 to 2000	100

Table 11-159 response_data parameter descriptions

Parameter	Type	Description
modularity	Double	Modularity
community_num	Integer	Number of communities
community	List	Community corresponding to each vertex. The format is as follows: [{vertexId:communityId},...] where vertexId : string type communityId : string type

12 Appendix

12.1 Status Codes

Table 12-1 describes status codes.

Table 12-1 Status codes

Status Code	Message	Description
100	Continue	The client should continue with its request. This interim response is used to inform the client that part of the request has been received and has not yet been rejected by the server.
101	Switching Protocols	The protocol should be switched. The protocol can only be switched to a newer protocol. For example, the current HTTP protocol is switched to a later version of HTTP.
201	Created	The request has been fulfilled and a new resource has been created.
202	Accepted	The request has been received but has not been processed.
203	Non-Authoritative Information	The server has successfully processed the request, but is returning information that may be from another source.
204	NoContent	The request has been fulfilled, but the HTTP response does not contain a response body. The status code is returned in response to an HTTP OPTIONS request.

Status Code	Message	Description
205	Reset Content	The server has successfully processed the request, but does not return any content.
206	Partial Content	The server has successfully processed the partial GET request.
300	Multiple Choices	There are multiple options for the location of the requested resource. The response contains a list of resource characteristics and addresses from which a user terminal (such as a browser) can choose the most appropriate one.
301	Moved Permanently	The requested resource has been assigned a new permanent URI, and the new URI is contained in the response.
302	Found	The requested resource resides temporarily under a different URI.
303	See Other	The response to the request can be found under a different URI. The response to the request can be found under a different URI, and should be retrieved using a GET or POST method.
304	Not Modified	The requested resource has not been modified. In such a case, there is no need to retransmit the resource since the client still has a previously-downloaded copy.
305	Use Proxy	The requested resource is available only through a proxy.
306	Unused	The HTTP status code is no longer used.
400	BadRequest	Invalid request. The client should modify the request instead of re-initiating it.
401	Unauthorized	This status code is returned after the client provides the authentication information, indicating that the authentication information is incorrect or invalid.
402	Payment Required	This status code is reserved for future use.

Status Code	Message	Description
403	Forbidden	The server has received the request and understood it, but the server is refusing to respond to it. The server has received and understood the request; yet it refused to respond, because the request is set to deny access. Do not retry the request before modification.
404	NotFound	The requested resource could not be found. The client should modify the request instead of re-initiating it.
405	MethodNotAllowed	The method specified in the request is not supported by the requested resource. The client should modify the request instead of re-initiating it.
406	Not Acceptable	The server could not fulfill the request according to the content characteristics of the request.
407	Proxy Authentication Required	This code is similar to 401, but indicates that the client must first authenticate itself with the proxy.
408	Request Time-out	The server timed out waiting for the request. The client may repeat the request without modifications at any time later.
409	Conflict	The request could not be processed due to a conflict in the request. This status code indicates that the resource that the client is attempting to create already exists, or that the request has failed to be processed because of the update of the conflict request.
410	Gone	The requested resource cannot be found. The status code indicates that the requested resource has been deleted permanently.
411	Length Required	The server is refusing to process the request without a defined Content-Length .
412	Precondition Failed	The server does not meet one of the preconditions that the requester puts on the request.

Stat us Cod e	Message	Description
413	Request Entity Too Large	The server is refusing to process a request because the request entity is too large for the server to process. The server may disable the connection to prevent the client from sending requests consecutively. If the server is only temporarily unable to process the request, the response will contain a Retry-After header field.
414	Request-URI Too Large	The Request-URI is too long for the server to process.
415	Unsupported Media Type	The server does not support the media type in the request.
416	Requested range not satisfiable	The requested range is invalid.
417	Expectation Failed	The server has failed to meet the requirements of the Expect request-header field.
422	UnprocessableEntity	The request was well-formed but was unable to be followed due to semantic errors.
429	TooManyRequests	The client has sent excessive number of requests to the server within a given time (exceeding the limit on the access frequency of the client), or the server has received an excessive number of requests within a given time (beyond its processing capability). In this case, the client should resend the request after the time specified in the Retry-After header of the response has elapsed.
500	InternalServerError	The server is able to receive the request but unable to understand it.
501	Not Implemented	The server does not support the requested function.
502	Bad Gateway	The server was acting as a gateway or proxy and received an invalid request from the remote server.
503	ServiceUnavailable	The requested service is invalid. The client should modify the request instead of re-initiating it.
504	ServerTimeout	The request cannot be fulfilled within a given time. This status code is returned to the client only when the Timeout parameter is specified in the request.
505	HTTP Version not supported	The server does not support the HTTP protocol version used in the request.

12.2 Error Codes

12.2.1 Error Codes for Management Plane APIs

If an error occurs in API calling, no result is returned. Identify the cause of error based on the error codes of each API. If an error occurs in API calling, HTTP status code 4xx or 5xx is returned. The response body contains the specific error code and information. If you fail to locate the cause of the error, contact technical support and provide the error code for troubleshooting.

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in [API Gateway Error Codes](#).

Table 12-2 Error codes

Status Code	Error Code	Error Message	Description	Solution
400	GES.001	Incorrect parameter.	Incorrect parameter.	1. Check whether the project ID or graph ID in the URL is correct. 2. Check whether the request header is correct, for example, whether X-Auth-Token is correct.
400	GES.016	Resource not found	Resource not found.	1. Check whether the project ID in the URL is the same as the project ID of the token. 2. Check whether the project ID in the URL is the same as the project ID of the graph.
400	GES.7000	The graph does not exist or has been deleted.	The graph does not exist or has been deleted.	1. Call the graph query API to query all graphs. 2. Check whether the project ID or graph ID in the URL is correct.
400	GES.7001	The graph is not running.	The graph is not running.	1. Call the graph query API to query all graphs. 2. View the graph list returned in the preceding step and check whether the graph status corresponding to the graph ID in the URL is 200.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES.7 002	The graph is being backed up.	The graph is being backed up.	1. Call the graph query API to query all graphs. 2. View the graph list returned in the preceding step and check whether the graph status corresponding to the graph ID in the URL is 903 .
400	GES.7 003	The graph is being stopped or is stopped.	The graph is being stopped or is stopped.	1. Call the graph query API to query all graphs. 2. View the graph list returned in the preceding step and check whether the graph status corresponding to the graph ID in the URL is 900 or 901 .
400	GES.7 004	Components at the IaaS layer are faulty.	Components at the IaaS layer are faulty.	Check whether the components at the IaaS layer, such as IAM, VPC, ECS, and OBS, are faulty.
408	GES.7 005	The underlying service of the graph engine is unavailable.	The underlying service of the graph engine is unavailable.	Try again later or contact technical personnel.
400	GES.7 006	An internal error occurs in the underlying service of the graph engine.	An internal error occurs in the underlying service of the graph engine.	Try again later or contact technical support.
400	GES.7 007	The job does not exist.	The job does not exist.	Check whether the job ID in the URL is correct.
400	GES.7 008	The job is stopped.	The job is stopped.	Jobs cannot be stopped repeatedly.
400	GES.7 009	The job operation is not supported.	The job operation is not supported.	The job operation is not supported.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES.7 010	Failed to verify the schema and data files.	Failed to verify the schema and data files.	Check whether the schema file matches the edge and vertex data files.
400	GES.7 011	The path or name of the schema or data file is invalid.	The path or name of the schema or data file is invalid.	Check whether the name of the scheme, vertex, or edge data file is valid. Only letters, numbers, underscores (_), exclamation marks (!), hyphens (-), periods (.), asterisks (*), left brackets, right brackets, and slashes (/) are allowed.
400	GES.7 012	Failed to verify the graph name.	Failed to verify the graph name.	Check the graph name. The name can contain 4 to 64 characters and must start with a letter. Only letters, numbers, and underscores (_) are allowed.
400	GES.7 013	The graph name already exists.	The graph name already exists.	1. Call the graph query API to query all graphs. 2. Query the graph list returned in the preceding step and check whether the name field in the request body already exists.
400	GES.7 014	An error is reported when the metadata verification API is called.	An error is reported when the metadata verification API is called.	Check whether the value of action_id is check-schema .
400	GES.7 015	The graph is not running or is stopped.	The graph is not running or is stopped.	1. Call the graph query API to query all graphs. 2. View the graph list returned in the preceding step and check whether the graph corresponding to the graph ID in the URL exists or is in the 900 status.
400	GES.7 016	The request body or header is invalid.	The request body or header is invalid.	Check the API reference and ensure that every configuration item in the request body and header is correctly configured.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES.7 017	The object does not exist. Check whether the bucket or object name is correct.	The object does not exist. Check whether the bucket or object name is correct.	Check whether the schema, vertex, and edge data files in the request body exist on OBS.
400	GES.7 018	The number of graphs or edges reaches the upper limit.	The number of graphs or edges reaches the upper limit.	Call the quota query API to check whether graphs have available quotas.
400	GES.7 019	The number of graph backups reaches the upper limit.	The number of graph backups reaches the upper limit.	Call the quota query API to check whether graph backups have available quotas.
400	GES.7 020	The VPC does not exist.	The VPC does not exist.	Check whether the VPC ID in the request body exists.
400	GES.7 021	The subnet cannot be found in the specified VPC.	The subnet cannot be found in the specified VPC.	Check whether the subnet ID in the request body exists or belongs to the preceding VPC.
400	GES.7 022	The security group does not exist.	The security group does not exist.	Check whether the security group ID in the request body exists.
400	GES.7 023	The graph size index is invalid.	The graph size index is invalid.	Check whether the graph size index in the request body is valid.
400	GES.7 024	The graph backup does not exist or has been deleted.	The graph backup does not exist or has been deleted.	1. Call the backup query API to query all backups of a specified graph. 2. Check whether the backup ID or graph ID in the URL is correct.

Status Code	Error Code	Error Message	Description	Solution
400	GES.7027	Failed to create an agency.	Failed to create an agency.	1. Assign the security administrator role to the user group to which the user belongs. For details, see Dealing with Insufficient Permission for Creating Agencies. 2. If the fault persists, report the error information in errorMessage to technical support.
400	GES.7028	Failed to authorize an agency.	Failed to authorize an agency.	1. Assign the security administrator role to the user group to which the user belongs. For details, see Dealing with Insufficient Permission for Creating Agencies. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES.7029	The agency resource exceeds the quota limit.	The agency resource exceeds the quota limit.	Check whether the agency resource reaches the quota limit on the IAM page.
400	GES.7030	Agency query error.	Agency query error.	Check the error message for detailed information.
400	GES.7031	Invalid binding type of an EIP.	Invalid binding type of an EIP.	Confirm the EIP binding type. The value can be either of the following: • auto_assign • bind_existing
400	GES.7032	The EIP resource exceeds the quota limit.	The EIP resource exceeds the quota limit.	Check whether the EIP resource reaches the quota limit on the VPC page.
400	GES.7033	Invalid EIP ID.	Invalid EIP ID.	If the EIP binding type is set to bind_existing , ensure that the EIP ID exists.

Status Code	Error Code	Error Message	Description	Solution
400	GES.7034	Resources in the current AZ have been sold out.	Resources in the current AZ have been sold out.	Switch to another AZ and try again.
400	GES.7035	Invalid region code.	Invalid region code.	Enter the correct region code.
400	GES.7036	The target version is earlier than the current version.	The target version is earlier than the current version.	A graph can only be upgraded to a later version.
400	GES.7037	The graph is not in the Stopped state.	The graph is not in the Stopped state.	Check whether the graph is in the Stopped state.
400	GES.7038	You cannot bind an EIP to a graph that has bound an EIP.	You cannot bind an EIP to a graph that has bound an EIP.	You cannot bind an EIP to a graph that has bound an EIP.
400	GES.7039	You cannot unbind an EIP from a graph that has not bound an EIP.	You cannot unbind an EIP from a graph that has not bound an EIP.	You cannot unbind an EIP from a graph that has not bound an EIP.
400	GES.7040	Failed to back up a graph.	Failed to back up a graph.	Failed to restore a graph from the backup you select.
400	GES.7041	Insufficient permission.	Insufficient permissions.	Insufficient permissions.
400	GES.7042	The graph is being created.	The graph is being created.	The graph is being created.
400	GES.7048	Invalid graph operation.	Invalid graph operation.	Check if the value of action_id is start , stop , import-graph , export-graph , clear-graph , or upgrade .

Status Code	Error Code	Error Message	Description	Solution
400	GES.7049	The parameter does not exist.	The parameter does not exist.	Check whether the request body is consistent with that in the API reference. Mandatory parameters must be set.
400	GES.7050	The parameter is empty.	The parameter is empty.	Check whether the request body is consistent with that in the API reference. Mandatory parameters must be set.
400	GES.7051	Components at the IaaS layer are faulty.	Components at the IaaS layer are faulty.	1. If the network fluctuates, try again later. 2. If the fault persists, obtain logs and send them to technical support.
400	GES.7052	Invalid CPU architecture of the graph instance.	Invalid CPU architecture of the graph instance.	Check whether the value of arch is set to x86_64 or aarch64 when the graph is created.
400	GES.7054	The graph is being deleted or has been deleted.	The graph is being deleted or has been deleted.	This error occurs in concurrent deletion scenarios. Generally, a message indicating that the graph does not exist is displayed when you try again later.
400	GES.7056	The graph of the current flavor cannot be scaled out.	Current graph cannot be resized.	Currently, Ten-thousand-edge and Ten-billion-edge graphs cannot be resized. Check whether the graph is a Ten-thousand-edge or Ten-billion-edge one.
400	GES.7057	Invalid graph flavor for scale-out.	Invalid graph size.	graphSizeTypeIndex in the resize request body can be set to 2 , 3 , 4 , or 5 , indicating the Ten-million-edge, Hundred-million-edge, Billion-edge, or Ten-billion-edge graph.
400	GES.7059	The IaaS resources of the graph flavor to be scaled out are insufficient.	There are insufficient IaaS resources for graph resize.	Check whether the compute resources are sufficient. For details about the IaaS resources required by each GES flavor, see the LLD.

Status Code	Error Code	Error Message	Description	Solution
400	GES.7061	Failed to create the ECS because the resources are insufficient.	Failed to create the ECS because the resources are insufficient.	Check whether the compute resources are sufficient. For details about the IaaS resources required by each GES flavor, see the LLD.
400	GES.7062	Failed to create the data disk.	Failed to create the data disk.	Check the FusionStorage capacity or obtain the detailed error information from the returned errorMessage .
400	GES.7063	Failed to create the system disk.	Failed to create the system disk.	Check the FusionStorage capacity or obtain the detailed error information from the returned errorMessage .
400	GES.7064	Failed to create the ECS.	Failed to create the ECS.	See the returned errorMessage or download the microservice log on the management plane, search for the ID of failed task in the log, and contact IaaS O&M personnel.
400	GES.7065	Failed to query the image because the image does not exist.	Failed to query the image because the image does not exist.	Check whether the image ID configured on the GES management plane exists.
400	GES.7066	Failed to query the flavor.	Failed to query the flavor.	Check whether the flavor ID configured on the GES management plane exists.
400	GES.7067	Insufficient ECS quota.	Insufficient ECS quota.	Check whether the flavor ID configured on the GES management plane exists.
400	GES.7068	Invalid request parameters.	Invalid request parameters.	During graph creation, parameters in the request for calling the IaaS API are invalid. Obtain detailed information based on the returned error message, and contact technical support..

Status Code	Error Code	Error Message	Description	Solution
400	GES.7069	The metadata file is too large.	The metadata file is too large.	The metadata file on OBS or that in the request body exceeds 10 MB.
400	GES.7070	Failed to parse the metadata file.	Failed to parse the metadata file.	When creating metadata, the metadata file on OBS or in the request body does not comply with the metadata standards. Create a metadata file or request body correctly.

12.2.2 Error Codes for Service Plane APIs

If an error occurs in API calling, no result is returned. Identify the cause of error based on the error codes of each API. If an error occurs in API calling, HTTP status code 4xx or 5xx is returned. The response body contains the specific error code and information. If you fail to locate the cause of the error, contact technical support and provide the error code for troubleshooting.

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in [API Gateway Error Codes](#).

Table 12-3 Error codes

Status Code	Error Code	Error Message	Description	Solution
400	GES.8000	Incorrect parameter format.	Incorrect parameter format.	Check whether the URL and request body are the same as those described in the document.
400	GES.8001	Failed to query graph statistics.	Failed to query graph statistics.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.

Status Code	Error Code	Error Message	Description	Solution
500	GES.8002	Graph statistics query error.	Graph statistics query error.	1. Check whether the token has expired. If it is expired, obtain a new one. 2. If the fault persists, report the error information in errorMessage to technical support.
400	GES.8005	Incorrect parameter.	Incorrect parameter.	1. Check whether the project ID in the URL is correct. 2. Check whether the request header is correct, for example, whether X-Auth-Token is correct.
400	GES.8006	Invalid resource access.	Invalid resource access.	1. AK/SK-based authentication: Verify if the region code or name is correct. 2. Token-based authentication: Verify if the project corresponding to the token is correct.
400	GES.8007	Invalid token.	Invalid token.	Check whether the token is correct.
400	GES.8008	An error occurs in the underlying authentication system.	An error occurs in the underlying authentication system.	Try again later or contact technical support.
400	GES.8011	Failed to export a graph.	Failed to export a graph.	1. Check whether the graph name is correct. 2. Check whether the export path is correct. 3. Check whether the account has the OBS write permission.
400	GES.8012	Failed to clear a graph.	Failed to clear a graph.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical support.

Status Code	Error Code	Error Message	Description	Solution
400	GES.8013	Failed to incrementally import data to the graph.	Failed to incrementally import data to the graph.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical support.
400	GES.8020	The current user does not have permission.	The current user does not have the required permission for granular permission control.	Grant permissions as the Security Administrator.
400	GES.8101	Invalid filter criteria for edge queries.	Invalid filter criteria for edge queries.	Check whether format of the filter criteria for edge queries is correct.
400	GES.8102	Invalid label for edge filtering queries.	Invalid label for edge filtering queries.	Check whether the labels are in the correct JSON format.
400	GES.8103	Both the condition and label of edge filtering queries are empty.	Both the condition and label of edge filtering queries are empty.	The condition and label of edge filtering queries cannot be both empty.
400	GES.8104	Invalid edge filtering query sequence.	Invalid edge filtering query sequence.	Check whether the edge filtering query sequence is valid.
400	GES.8105	Failed to query edges that meet filter criteria.	Failed to query edges that meet filter criteria.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.

Status Code	Error Code	Error Message	Description	Solution
400	GES.8106	The source vertex or target vertex in the edge details is empty.	The source vertex or target vertex in the edge details is empty.	Ensure that the source vertex or target vertex in the edge details cannot be empty.
400	GES.8107	Failed to query edge details.	Failed to query edge details.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
500	GES.8108	Edge details query error.	Edge details query error.	Try again later or contact technical personnel.
400	GES.8109	Invalid edge filtering query operator.	Invalid edge filtering query operator.	Ensure that the values of edge filtering query operators are in , out , both , and edge .
400	GES.8110	Parameter edges cannot be left blank.	Parameter edges cannot be left blank.	Check whether the value of edges in the batch edge query request body is empty.
400	GES.8201	Invalid label for vertex filtering queries.	Invalid label for vertex filtering queries.	Check whether the labels are in the correct JSON format.
400	GES.8202	Invalid filter criteria for vertex queries.	Invalid filter criteria for vertex queries.	1. Check whether propertyName of the vertex query API is left blank. 2. Check whether values of the vertex query API is left blank. 3. If the fault persists, report the error information in errorMessage to technical personnel.

Status Code	Error Code	Error Message	Description	Solution
400	GES.8203	Both the condition and label of vertex filtering queries are empty.	Both the condition and label of vertex filtering queries are empty.	Ensure that the condition and label of vertex filtering queries are not both empty.
400	GES.8204	Failed to query vertices that meet filter criteria.	Failed to query vertices that meet filter criteria.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES.8205	Invalid vertex filtering query sequence.	Invalid vertex filtering query sequence.	In the vertex filtering query API, orderValue must be set to incr or decr .
400	GES.8206	Both vertexid and vertexids exist.	Both vertexid and vertexids exist.	vertexid and vertexids cannot coexist.
400	GES.8207	Both vertexid and vertexids are empty.	Both vertexid and vertexids are empty.	The vertexid or vertexids parameter is empty.
400	GES.8208	Incorrect vertexids format.	Incorrect vertexids format.	Check whether vertexids is a JSON array.
400	GES.8209	Failed to query vertex details.	Failed to query vertex details.	Check whether the graph name exists.
500	GES.8210	Vertex details query error.	Vertex details query error.	Try again later or contact technical personnel.
400	GES.8211	Invalid vertex filtering query operator.	Invalid vertex filtering query operator.	Ensure that values of vertex filtering query operators are inV , outV , bothV , and vertex .

Status Code	Error Code	Error Message	Description	Solution
400	GES.8212	Failed to delete the vertex label.	Failed to delete the vertex label.	Check whether the label exists.
400	GES.8213	Failed to add the vertex label.	Failed to add the vertex label.	Check whether the label exists.
400	GES.8214	Parameter vertices cannot be left blank.	Parameter vertices cannot be left blank.	Check whether the value of vertices in the batch vertex query request body is empty.
400	GES.8220	Failed to update the vertex properties.	Failed to update the vertex properties.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES.8221	Failed to update the edge properties.	Failed to update the edge properties.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES.8301	Failed to query a job.	Failed to query a job.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
500	GES.8302	Job query error.	Job query error.	Try again later or contact technical personnel.
400	GES.8303	Failed to terminate a job.	Failed to terminate a job.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
500	GES.8304	Job termination error.	Job termination error.	Try again later or contact technical personnel.

Statu s Code	Error Code	Error Message	Description	Solution
400	GES.84 01	The algorithm or graph name cannot be empty.	The algorithm or graph name cannot be empty.	Ensure that the algorithm or graph name in not empty.
400	GES.84 02	Failed to run the algorithm.	Failed to run the algorithm.	1. If the network fluctuates, try again later. 2. Check whether the graph name in the algorithm running API is correct. 3. If the fault persists, report the error information in errorMessage to technical personnel.
500	GES.84 03	Algorithm running error.	Algorithm running error.	Try again later or contact technical personnel.
400	GES.84 04	Invalid algorithm running format.	Invalid algorithm running format.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES.85 01	The Gremlin command is not supported.	The Gremlin command is not supported.	Replace the unsupported Gremlin statements: tryNext, explain, and tree.
400	GES.85 02	Failed to find the Gremlin configuration file.	Failed to find the Gremlin configuratio n file.	Try again later or contact technical personnel.
400	GES.85 03	Gremlin query failed.	Gremlin query failed.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
500	GES.85 04	Gremlin query error.	Gremlin query error.	Try again later or contact technical personnel.

Status Code	Error Code	Error Message	Description	Solution
400	GES.8505	The Gremlin query statement does not contain the command field.	The Gremlin query statement does not contain the command field.	Ensure that the Gremlin query statement does not contain the command field.
400	GES.8506	The size of the Gremlin query request statements exceeds the upper limit.	The size of the Gremlin query request statements exceeds the upper limit.	The current limit is 64 MB.
500	GES.8601	Gremlin service unavailable.	Gremlin service unavailable.	Try again later or contact technical personnel.
500	GES.8602	Engine service unavailable.	Engine service unavailable.	Try again later or contact technical personnel.
400	GES.8603	Failed to create an index	Failed to create an index	1. Check whether the index name contains only letters, numbers, hyphens (-), and underscores (_). 2. Check whether the index parameter type complies with that specified by the API.
400	GES.8604	Failed to delete an index	Failed to delete an index	1. Check whether the graph name is correct. 2. Check whether the index name is correct. 3. Check whether Method type of the request is delete.
400	GES.8605	Failed to query an index	Failed to query an index	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.

Status Code	Error Code	Error Message	Description	Solution
400	GES.8609	The request body for querying path details is invalid.	The request body for querying path details is invalid.	1. Check whether the graph name is correct. 2. Check whether the parameter format of the API for querying path details is correct. 3. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES.8610	The path parameter of the request body for querying path details is invalid.	The path parameter of the request body for querying path details is invalid.	1. Check whether the parameter format of the API for querying path details is correct. 2. Check whether the mandatory parameters of the API for querying path details are set. 3. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES.8611	Failed to query path details.	Failed to query path details.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES.8612	The operation of querying path details is not supported.	The operation of querying path details is not supported.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.
400	GES.8801	Failed to add a label to metadata.	Failed to add a label to metadata.	1. Check whether the label to be added already exists. 2. Check whether the format of the parameter for adding the label is correct. 3. Check whether the mandatory parameters for adding the label are set.

Status Code	Error Code	Error Message	Description	Solution
400	GES.8803	Failed to query the metadata.	Failed to query the metadata.	1. Check whether the graph to be queried exists. 2. Check whether the value of graph_name in the API for querying graph metadata is correct.
500	GES.8804	Metadata query error.	Metadata query error.	Try again later or contact technical personnel.
400	GES.8806	K-Hop query with filter criteria failed.	K-Hop query with filter criteria failed.	1. If the network fluctuates, try again later. 2. If the fault persists, report the error information in errorMessage to technical personnel.

12.3 Obtaining a Project ID

Obtaining a Project ID by Calling an API

You can obtain a project ID by calling the API used to [query project information based on the specified criteria](#).

The API for obtaining a project ID is **GET https://{Endpoint}/v3/projects**. **{Endpoint}** indicates the endpoint of IAM, which can be obtained from [Regions and Endpoints](#).

For details about API authentication, see [Making a Management Plane API Request](#).

The following is an example response. The value of **id** under **projects** is the project ID. The following is an example response. If GES is deployed in the **ap-southeast-1** region, the value of **name** in the response body is **ap-southeast-1**, and the value of **id** in **projects** is the project ID.

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "ap-southeast-1",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "id": "a4a5d4098fb4474fa22cd05f897d6b99",
      "enabled": true
    }
  ]
}
```

```
}  
],  
"links": {  
  "next": null,  
  "previous": null,  
  "self": "https://www.example.com/v3/projects"  
}  
}
```

Obtaining a Project ID from the Console

A project ID is required for some URIs when an API is called.

To obtain a project ID, perform the following operations:

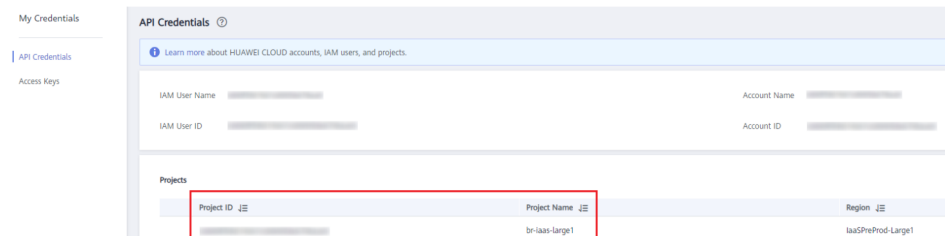
1. Log in to the [management console](#).
2. In the upper right corner of the page, click the username and choose **My Credentials** from the drop-down list. The **My Credentials** page is displayed.

NOTE

If you have logged in to the Huawei Cloud official website instead of the management console, after clicking your username in the upper right corner, choose **My Account** from the drop-down list. On the **Basic Information** page, click **Manage** following **Security Credentials**. Then, the **My Credentials** page is displayed.

3. On the **API Credentials** page, obtain the project ID and name from the project list.

Figure 12-1 Checking the project ID



If there are multiple projects in one region, expand **Region** and view subproject IDs in the **Project ID** column.

12.4 Obtaining the Account Name and Account ID

An account ID is required for some requests for calling APIs. To obtain an account ID, perform the following operations:

1. Log in to the [management console](#).
2. Hover the cursor on the username and select **My Credentials** from the drop-down list.

On the **API Credentials** page, view the account name and ID.

Figure 12-2 Viewing the account name and ID

