

代码检查

API 参考

文档版本 01
发布日期 2025-08-19



版权所有 © 华为技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

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

如企业客户须获取漏洞信息，请参见如下网址：

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

目录

1 使用前必读	1
2 API 概览	3
3 如何调用 API	4
3.1 构造请求	4
3.2 认证鉴权	6
3.3 返回结果	11
4 API	13
4.1 任务管理	13
4.1.1 新建检查任务	13
4.1.2 删除检查任务	24
4.1.3 查询任务列表	28
4.1.4 执行检查任务	33
4.1.5 终止检查任务	37
4.1.6 查询任务执行状态	41
4.1.7 历史扫描结果查询	45
4.1.8 查询任务的已选规则集列表 v2	51
4.1.9 查询任务的已选规则集列表 v3	55
4.1.10 查询任务规则集的检查参数 v2	61
4.1.11 查询任务规则集的检查参数 v3	64
4.1.12 配置任务检查参数	67
4.1.13 修改任务规则集	76
4.1.14 查询任务检查失败日志	83
4.1.15 获取任务的目录树	89
4.1.16 任务配置屏蔽目录	94
4.1.17 查询任务的高级配置	100
4.1.18 任务配置高级配置	103
4.1.19 获取代码仓所有文件列表	109
4.1.20 责任人精确匹配开关	111
4.1.21 取消检查	114
4.1.22 根据任务 id 获取任务 WebhookUrl 以及开关状态	117
4.2 缺陷管理	121
4.2.1 查询缺陷概要	121

4.2.2 查询代码度量概要.....	130
4.2.3 查询缺陷详情.....	140
4.2.4 查询缺陷详情的统计.....	149
4.2.5 修改缺陷状态.....	154
4.3 规则管理.....	159
4.3.1 获取规则列表接口.....	159
4.3.2 创建自定义规则集.....	165
4.3.3 查询规则集列表.....	176
4.3.4 查看规则集的规则列表.....	182
4.3.5 删除自定义规则集.....	189
4.3.6 设置每个项目对应语言的默认规则集配置.....	194
5 场景化示例.....	200
5.1 示例 1: 查询任务列表.....	200
5.2 示例 2: 查询缺陷概要.....	202
6 附录.....	204
6.1 状态码.....	204
6.2 错误码.....	206
6.3 获取项目 ID.....	208
6.3.1 获取 IAM 项目 ID.....	208
6.3.2 获取 CodeArts 项目 ID.....	209
6.4 获取账号 ID.....	209
6.5 Cmetrics 工具规格说明.....	210
7 修订记录.....	217

1 使用前必读

欢迎使用代码检查服务（CodeArts Check）。代码检查是基于云端实现代码质量管理的服务，软件开发者可在编码完成后执行多语言的代码静态检查和安全检查，获取全面的质量报告，并提供缺陷的分组查看与改进建议，有效管控代码质量，帮助产品成功。

您可以使用本文档提供API对代码检查进行相关操作，如创建、删除、查询任务等。支持的全部操作请参见[API概览](#)。

在调用代码检查API之前，请确保已经充分了解代码检查[基本概念](#)。

终端节点

终端节点即调用API的[请求地址](#)，不同服务不同区域的终端节点不同，服务的终端节点信息可参考[地区和终端节点](#)。

代码检查的终端节点如[表1-1](#)所示，请您根据业务需要选择对应区域的终端节点。

表 1-1 代码检查的终端节点

区域名称	区域	终端节点（Endpoint）
拉美-墨西哥城二	la-north-2	devcloud.la-north-2.myhuaweicloud.com
拉美-圣保罗一	sa-brazil-1	devcloud.sa-brazil-1.myhuaweicloud.com
亚太-新加坡	ap-southeast-3	devcloud.ap-southeast-3.myhuaweicloud.com

基本概念

- 账号
用户注册华为云时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用他们进行日常管理工作。

- 用户
由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。
在[我的凭证](#)下，您可以查看账号ID和用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。
- 区域（Region）
从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region。
 - 通用Region指面向公共租户提供通用云服务的Region。
 - 专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。
- 可用区（AZ，Availability Zone）
一个AZ是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。
- 企业项目
企业项目是项目的升级版，针对企业不同项目间资源的分组和管理，是逻辑隔离。企业项目中可以包含多个区域的资源，且项目中的资源可以迁入迁出。
关于企业项目ID的获取及企业项目特性的详细信息，请参见《[企业管理服务用户指南](#)》。

2 API 概览

表 2-1 代码检查 API 概览

类型	说明
任务管理	任务管理包括：新建、执行、查询、终止任务等。
缺陷管理	缺陷管理包括：问题概述、问题状态、圈复杂度、代码重复率等缺陷详情。
规则管理	规则管理包括：获取规则列表接口、创建自定义规则集、查询规则集列表等。

3 如何调用 API

3.1 构造请求

本节介绍REST API请求的组成，并以调用IAM服务的[管理员创建IAM用户](#)接口为例说明如何调用API。

请求 URI

尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

请求URI由四部分构成：**{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}**

各参数含义如下：

参数	说明
URI-scheme	表示用于传输请求的协议，当前所有API均采用HTTPS协议。
Endpoint	指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从 地区和终端节点 中获取。 例如，IAM服务在亚太-新加坡区域的Endpoint为iam.ap-southeast-3.myhuaweicloud.com。
resource-path	资源路径，即API访问路径，从具体API的URI模块获取。例如“管理员创建IAM用户”API的resource-path为“/v3.0/OS-USER/users”。
query-string	查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个?，形式为 参数名=参数取值 。例如， limit=10 表示查询不超过10条数据。

例如，您需要创建IAM用户，由于IAM为全局服务，则使用任一区域的Endpoint即可，比如“亚太-新加坡”区域的Endpoint（iam.ap-

southeast-3.myhuaweicloud.com)，并在[管理员创建IAM用户](#)的URI部分找到 resource-path (/v3.0/OS-USER/users)，拼接起来如下所示。

```
https://iam.ap-southeast-3.myhuaweicloud.com/v3.0/OS-USER/users
```

图 3-1 URI 示意图



说明

为查看方便，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，而Endpoint在同一个区域也相同，所以简洁起见将这两部分省略。

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

- **GET**：请求服务器返回指定资源。
- **PUT**：请求服务器更新指定资源。
- **POST**：请求服务器新增资源或执行特殊操作。
- **DELETE**：请求服务器删除指定资源，如删除对象等。
- **HEAD**：请求服务器资源头部。
- **PATCH**：请求服务器更新资源的部分内容。当资源不存在的时候，PATCH可能会去创建一个新的资源。

在[管理员创建IAM用户](#)的URI部分，您可以看到其请求方法为“POST”，则其请求为：

```
POST https://iam.ap-southeast-3.myhuaweicloud.com/v3.0/OS-USER/users
```

请求消息头

附加请求头字段，如指定的URI和HTTP方法所要求的字段。例如定义消息体类型的请求头“Content-Type”，请求鉴权信息等。

如下公共消息头需要添加到请求中。

- **Content-Type**：消息体的类型（格式），必选，默认取值为“application/json”，有其他取值时会在具体接口中专门说明。
- **Authorization**：签名认证信息，可选，当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。AK/SK认证的详细说明请参考[AK/SK认证](#)。
- **X-Sdk-Date**：请求发送的时间，可选，当使用AK/SK方式认证时，使用SDK对请求进行签名的过程中会自动填充该字段。AK/SK认证的详细说明请参考[AK/SK认证](#)。

- **X-Auth-Token**: 用户Token，可选，当使用Token方式认证时，必须填充该字段。用户Token也就是调用**获取用户Token**接口的响应值，该接口是唯一不需要认证的接口。
- **X-Project-ID**: 子项目ID，可选，在多项目场景中使用。如果云服务资源创建在子项目中，AK/SK认证方式下，操作该资源的接口调用需要在请求消息头中携带X-Project-ID。
- **X-Domain-ID**: 账号ID，可选。AK/SK认证方式下，全局服务的接口调用时，需在请求消息头中携带X-Domain-ID。

对于**管理员创建IAM用户**接口，使用AK/SK方式认证时，添加消息头后的请求如下所示。

```
POST https://iam.ap-southeast-3.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****
```

请求消息体

请求消息体通常以结构化格式发出，与请求消息头中Content-type对应，传递除请求消息头之外的内容。

每个接口的请求消息体内容不同，也并不是每个接口都需要有请求消息体（或者说消息体为空），GET、DELETE操作类型的接口就不需要消息体，消息体具体内容需要根据具体接口而定。

对于**获取用户Token**获取请求认证接口，您可以从接口的请求部分看到所需的请求参数及参数说明。将消息体加入后的请求如下所示，加粗的斜体字段需要根据实际值填写。

- **accountid**为IAM用户所属的账号ID。
- **username**为要创建的IAM用户名。
- **email**为IAM用户的邮箱。
- *********为IAM用户的登录密码。

```
POST https://iam.ap-southeast-3.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****

{
  "user": {
    "domain_id": "accountid",
    "name": "username",
    "password": "*****",
    "email": "email",
    "description": "IAM User Description"
  }
}
```

到这里为止这个请求需要的内容就具备齐全了，您可以使用curl、Postman或直接编写代码等方式发送请求调用API。

3.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- Token认证：通过Token认证通用请求。
- AK/SK认证：通过AK (Access Key ID) /SK (Secret Access Key)加密调用请求。

Token 认证

📖 说明

- Token的有效期为24小时，需要使用一个Token鉴权时，可以先缓存起来，避免频繁调用。
- 使用Token前请确保Token离过期有足够的时间，防止调用API的过程中Token过期导致调用API失败。

Token在计算机系统中代表令牌（临时）的意思，拥有Token就代表拥有某种权限。Token认证就是在调用API的时候将Token加到请求消息头，从而通过身份认证，获得操作API的权限。

Token可通过调用[获取用户Token](#)接口获取，调用本服务API需要project级别的Token，即调用接口时，请求body中auth.scope的取值需要选择project，如下所示。

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

获取Token后，再调用其他接口时，您需要在请求消息头中添加“X-Auth-Token”，其值即为Token。例如Token值为“ABCDEFJ....”，则调用接口时将“X-Auth-Token: ABCDEFJ....”加到请求消息头即可，如下所示。

```
GET https://iam.ap-southeast-3.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK 认证

通过使用Access Key ID (AK) /Secret Access Key (SK) 加密的方法来验证某个请求发送者身份。当您使用AK/SK认证方式完成认证鉴权时，需要通过请求签名流程获取签名并增加到业务接口请求消息头。

说明

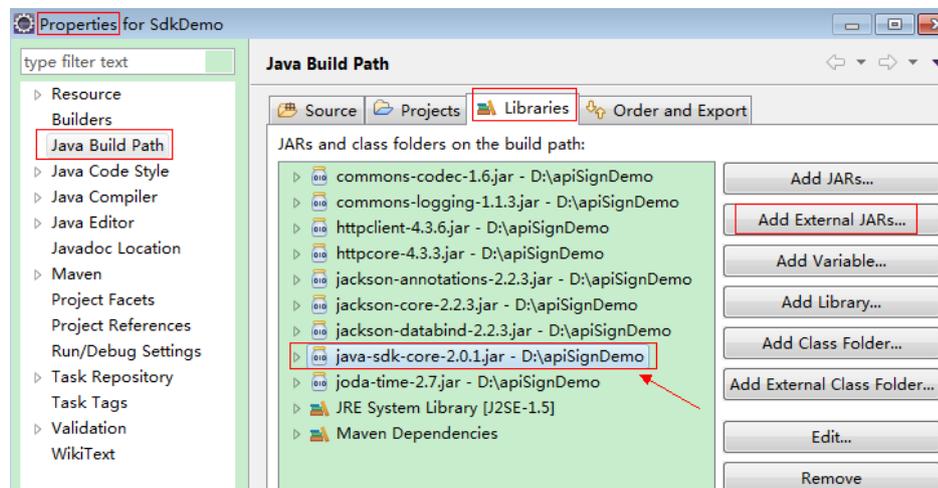
AK(Access Key ID): 访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和私有访问密钥一起使用，对请求进行加密签名。

SK(Secret Access Key): 与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。

以下结合一个Demo来介绍如何对一个请求进行签名，并通过HTTP Client发送一个HTTPS请求的过程。

Demo下载地址：<https://github.com/api-gate-way/SdkDemo>。

解压下载的压缩包，得到一个jar文件。将解压出来的jar文件引用到依赖路径中。如下图所示：



步骤1 生成AK/SK。如果已生成过AK/SK，则可跳过该步骤，找到原来已下载的AK/SK文件，文件名一般为：credentials.csv。

1. 登录管理控制台。
2. 单击用户名，在下拉列表中单击“我的凭证”。
3. 单击“管理访问密钥”。
4. 单击“新增访问密钥”，进入“新增访问密钥”页面。
5. 输入当前用户的登录密码。
6. 通过邮箱或者手机进行验证，输入对应的验证码。

说明

在统一身份服务中创建的用户，如果创建时未填写邮箱或者手机号，则只需校验登录密码。

7. 单击“确定”，下载访问密钥。

说明

为防止访问密钥泄露，建议您将其保存到安全的位置。

步骤2 获取示例代码，解压缩。

步骤3 通过import方式将示例工程导入到Eclipse。

图 3-2 选择已存在的工程

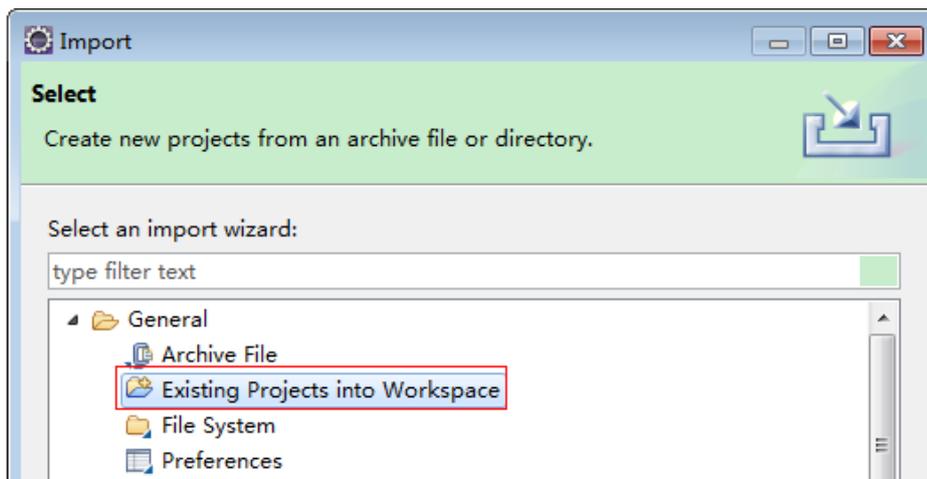


图 3-3 选择解压后的示例代码

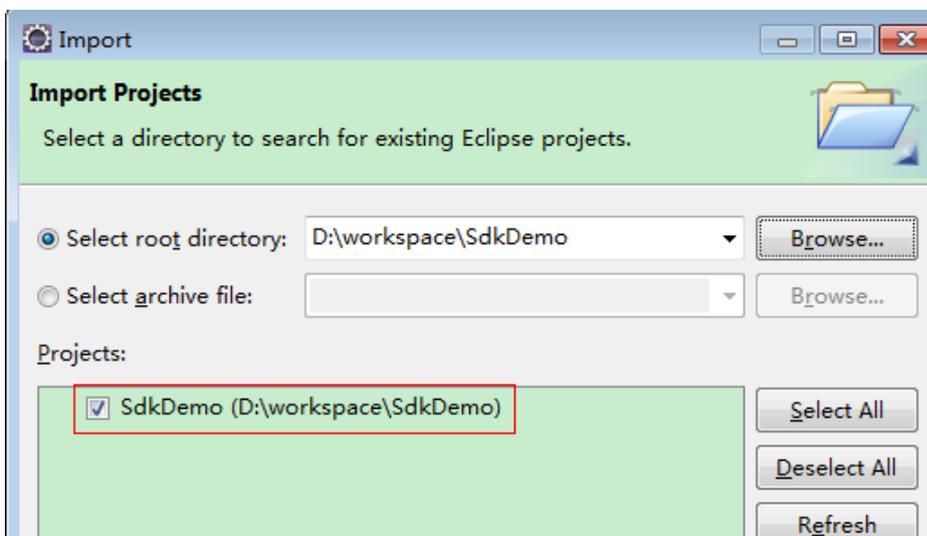
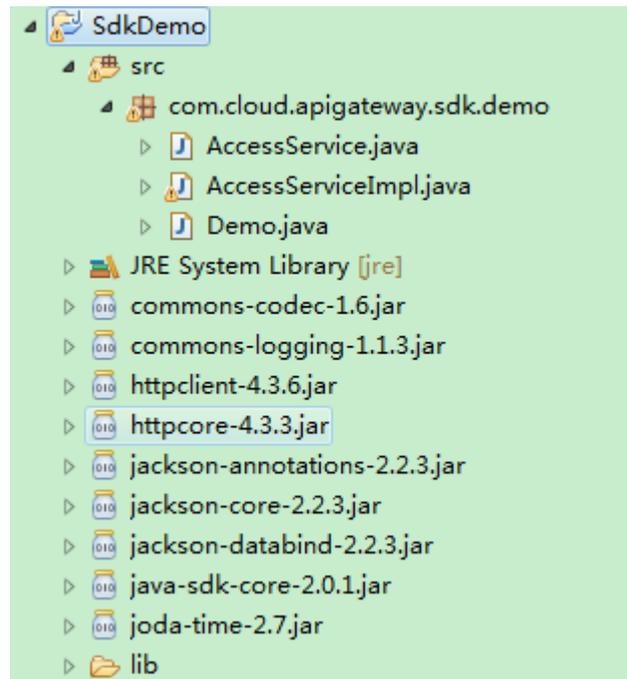


图 3-4 导入成功后工程结构示例



步骤4 对请求进行签名。

签名方法集成在**步骤3**引入的jar文件中。发送请求前，需要对请求内容进行签名，得到的签名结果将作为http头部信息一起发送。

Demo代码分成三个类进行演示：

- AccessService:抽象类，将GET/POST/PUT/DELETE归一成access方法。
- Demo:运行入口，模拟用户进行GET/POST/PUT/DELETE请求。
- AccessServiceImpl:实现access方法，具体与API网关通信的代码都在access方法中。

1. （可选）添加请求消息头。

请在“AccessServiceImpl.java”文件中找到如下行，取消代码行屏蔽，并且替换成实际的子项目ID和账号ID。

```
//TODO: Add special headers.
//request.addHeader("X-Project-Id", "xxxxx");
//request.addHeader("X-Domain-Id", "xxxxx");
```

2. 编辑“Demo.java”文件中的main方法，将以下内容替换为实际获取到的值。

如下所示，加粗部分需要替换成实际值，如果调用其他方法，如POST，PUT，DELETE等，请参考对应注释方法。

注意替换 region、serviceName、AK/SK 和 URL，Demo中使用了获取VPC的URL，请替换为您需要的URL，URL中project_id获取请参见[获取IAM项目ID](#)，Endpoint获取请参见[地区和终端节点](#)。

```
//TODO: Replace region with the name of the region in which the service to be accessed is located.
private static final String region = "";

//TODO: Replace serviceName with the name of the service you want to access. For example, ecs,
vpc, iam, and elb.
private static final String serviceName = "";

public static void main(String[] args) throws UnsupportedOperationException
{
```

```
//TODO: Replace the AK and SK with those obtained on the My Credential page.
String ak = "ZIRRKMTWP*****1WKNKB";
String sk = "Us0mdMNHk*****YrRCnW0ecfzl";

//TODO: To specify a project ID (multi-project scenarios), add the X-Project-Id header.
//TODO: To access a global service, such as IAM, DNS, CDN, and TMS, add the X-Domain-Id header
to specify an account ID.
//TODO: To add a header, find "Add special headers" in the AccessServiceImple.java file.

//TODO: Test the API
String url = "https://{Endpoint}/v1/{project_id}/vpcs";
get(ak, sk, url);

//TODO: When creating a VPC, replace {project_id} in postUrl with the actual value.
//String postUrl = "https://serviceEndpoint/v1/{project_id}/cloudservers";
//String postbody = "{\"vpc\": {\"name\": \"vpc\", \"cidr\": \"192.168.0.0/16\"}}";
//post(ak, sk, postUrl, postbody);

//TODO: When querying a VPC, replace {project_id} in url with the actual value.
//String url = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//get(ak, sk, url);

//TODO: When updating a VPC, replace {project_id} and {vpc_id} in putUrl with the actual values.
//String putUrl = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//String putbody = "{\"vpc\": {\"name\": \"vpc1\", \"cidr\": \"192.168.0.0/16\"}}";
//put(ak, sk, putUrl, putbody);

//TODO: When deleting a VPC, replace {project_id} and {vpc_id} in deleteUrl with the actual values.
//String deleteUrl = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//delete(ak, sk, deleteUrl);
}
```

3. 编译与运行接口调用。

在左侧“Package Explorer”中找到“Demo.java”，右键选择“Run AS >Java Application”并单击运行。

可在控制台查看调用日志。

----结束

3.3 返回结果

状态码

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

状态码是一组从1xx到5xx的数字代码，状态码表示了请求响应的状态，完整的状态码列表请参见[状态码](#)。

对于[管理员创建IAM用户](#)接口，如果调用后返回状态码为“201”，则表示请求成功。

响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

对于[管理员创建IAM用户](#)接口，返回如[图3-5](#)所示的消息头。

图 3-5 管理员创建 IAM 用户响应消息头

```
"X-Frame-Options": "SAMEORIGIN",
"X-IAM-ETag-id": "2562365939-d8f6f12921974cb097338ac11fcec8a",
"Transfer-Encoding": "chunked",
"Strict-Transport-Security": "max-age=31536000; includeSubdomains;",
"Server": "api-gateway",
"X-Request-Id": "af2953f2bcc67a42325a69a19e6c32a2",
"X-Content-Type-Options": "nosniff",
"Connection": "keep-alive",
"X-Download-Options": "noopen",
"X-XSS-Protection": "1; mode=block;",
"X-IAM-Trace-Id": "token_█_null_af2953f2bcc67a42325a69a19e6c32a2",
"Date": "Tue, 21 May 2024 09:03:40 GMT",
"Content-Type": "application/json; charset=utf8"
```

响应消息体

响应消息体通常以结构化格式返回，与响应消息头中Content-type对应，传递除响应消息头之外的内容。

对于[管理员创建IAM用户](#)接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

```
{
  "user": {
    "id": "c131886aec...",
    "name": "IAMUser",
    "description": "IAM User Description",
    "areacode": "",
    "phone": "",
    "email": "****@***.com",
    "status": null,
    "enabled": true,
    "pwd_status": false,
    "access_mode": "default",
    "is_domain_owner": false,
    "xuser_id": "",
    "xuser_type": "",
    "password_expires_at": null,
    "create_time": "2024-05-21T09:03:41.000000",
    "domain_id": "d78cbac1.....",
    "xdomain_id": "30086000.....",
    "xdomain_type": "",
    "default_project_id": null
  }
}
```

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

```
{
  "error_msg": "Request body is invalid.",
  "error_code": "IAM.0011"
}
```

其中，error_code表示错误码，error_msg表示错误描述信息。

4 API

4.1 任务管理

4.1.1 新建检查任务

功能介绍

新建检查任务但是不执行。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/task

表 4-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释: 项目的32位uuid, 项目唯一标识, 通过 查询项目列表 接口查询项目列表获取。 约束限制: 不涉及。 取值范围: 32位项目ID。 默认取值: 不涉及。

请求参数

表 4-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

表 4-3 请求 Body 参数

参数	是否必选	参数类型	描述
git_url	是	String	<p>参数解释: 代码仓地址。</p> <p>约束限制: 不涉及</p> <p>取值范围: repo仓传以git开头的地址；非repo仓传以http开头的地址。</p> <p>默认取值: 不涉及。</p>
git_branch	是	String	<p>参数解释: 计划创建任务的对应代码仓的分支。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

参数	是否必选	参数类型	描述
language	否	Array of strings	<p>参数解释: 启用的检查语言，数组格式，支持的语言请关注服务更新特性文档。</p> <p>约束限制: 与rule_sets 2选1使用。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
rule_sets	否	Array of RuleSetV2 objects	<p>参数解释: 启用的规则集，数组格式，当只传语言，不传规则集id时，与language参数等效，启用这些语言在项目下的默认规则集，支持的语言请关注服务更新特性文档。</p> <p>约束限制: 与language 2选1使用。</p>
task_type	否	String	<p>参数解释: 任务类型。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> ● full: 代码仓分支对应的全量任务。 ● inc: MR增量任务。 <p>默认取值: full</p>

参数	是否必选	参数类型	描述
username	否	String	<p>参数解释: 用户名, 与access_token联合使用, 用于在创建三方仓时, 由Check服务创建1个通用git扩展点, 从而保存代码仓信息。</p> <p>约束限制: username、access_token组合与endpoint_id 2选1使用。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
access_token	否	String	<p>参数解释: 密码或访问密钥/令牌, username, 用于在创建三方仓时, 由Check服务创建1个通用git扩展点, 从而保存代码仓信息。</p> <p>约束限制: username、access_token组合与endpoint_id 2选1使用。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
endpoint_id	否	String	<p>参数解释: 用户预先创建的项目下的代码源扩展点ID, 保存了代码仓信息。</p> <p>约束限制: username、access_token组合与endpoint_id 2选1使用。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
inc_config	否	IncConfigV2 object	<p>参数解释: 基于某一任务创建分支任务、MR任务时使用的参数。</p> <p>约束限制: 不涉及。</p>

参数	是否必选	参数类型	描述
resource_pool_id	否	String	<p>参数解释: 用户购买的资源池ID，创建任务时如传此参数，任务将使用对应资源池进行检查。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

表 4-4 RuleSetV2

参数	是否必选	参数类型	描述
ruleset_id	否	String	<p>参数解释: 规则集ID。</p> <p>约束限制: 32位UUID。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 语言在项目下的默认规则集。</p>
language	是	String	<p>参数解释: 检查语言，支持的语言请关注服务更新特性文档。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

表 4-5 IncConfigV2

参数	是否必选	参数类型	描述
parent_task_id	否	String	参数解释: 需要关联的任务ID。 约束限制: 32位UUID。 取值范围: 不涉及。 默认取值: 不涉及。
git_source_branch	否	String	参数解释: MR增量任务的源分支。 约束限制: 不涉及。 取值范围: 不涉及。 默认取值: 不涉及。
git_target_branch	否	String	参数解释: MR增量任务的目标分支。 约束限制: 不涉及。 取值范围: 不涉及。 默认取值: 不涉及。
merge_id	否	String	参数解释: MR编号。 约束限制: 不涉及。 取值范围: 不涉及。 默认取值: 不涉及。

参数	是否必选	参数类型	描述
event_type	否	String	参数解释: 创建MR增量任务时需传。 约束限制: 不涉及。 取值范围: merge_request。 默认取值: 不涉及。
action	否	String	参数解释: 创建MR增量任务时需传。 约束限制: 不涉及。 取值范围: open。 默认取值: 不涉及。
title	否	String	参数解释: 创建MR增量任务时需传，MR标题。 约束限制: 不涉及。 取值范围: 不涉及。 默认取值: 不涉及。

响应参数

状态码：200

表 4-6 响应 Body 参数

参数	参数类型	描述
task_id	String	参数解释: 任务ID。 取值范围: 全量任务为32位UUID，MR增量任务为"MR_"+32位UUID。

状态码：400

表 4-7 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-8 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

新建检查任务：检查任务的检查类型为source，对应代码仓URL根据实际情况显示，代码仓分支为master，语言类型为cpp

POST https://{endpoint}/v2/{project_id}/task

```
{
  "check_type" : [ "source" ],
  "git_url" : "git@code*****958.git",
  "git_branch" : "master",
  "language" : [ "cpp", "js" ]
}
```

响应示例

状态码：200

Request succeeded!

```
{
  "task_id" : "d161fd0*****cb3c1d6a783e"
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CC.00000000",
  "error_msg" : "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code" : "CC.00000003",
}
```

```
"error_msg" : "认证信息过期"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

新建检查任务：检查任务的检查类型为source，对应代码仓URL根据实际情况显示，代码仓分支为master，语言类型为cpp

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;  
import com.huaweicloud.sdk.codecheck.v2.*;  
import com.huaweicloud.sdk.codecheck.v2.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class CreateTaskSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new BasicCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        CodeCheckClient client = CodeCheckClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CreateTaskRequest request = new CreateTaskRequest();  
        request.withProjectId("{project_id}");  
        CreateTaskRequestV2 body = new CreateTaskRequestV2();  
        List<String> listbodyLanguage = new ArrayList<>();  
        listbodyLanguage.add("cpp");  
        listbodyLanguage.add("js");  
        List<String> listbodyCheckType = new ArrayList<>();  
        listbodyCheckType.add("source");  
        body.withLanguage(listbodyLanguage);  
        body.withGitBranch("master");  
        body.withGitUrl("git@code*****958.git");  
        body.withCheckType(listbodyCheckType);  
        request.withBody(body);  
        try {  
            CreateTaskResponse response = client.createTask(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
        System.out.println(e.getStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

新建检查任务：检查任务的检查类型为source，对应代码仓URL根据实际情况显示，代码仓分支为master，语言类型为cpp

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcodecheck.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = CodeCheckClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateTaskRequest()
        request.project_id = "{project_id}"
        listLanguagebody = [
            "cpp",
            "js"
        ]
        listCheckTypebody = [
            "source"
        ]
        request.body = CreateTaskRequestV2(
            language=listLanguagebody,
            git_branch="master",
            git_url="git@code*****958.git",
            check_type=listCheckTypebody
        )
        response = client.create_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

新建检查任务：检查任务的检查类型为source，对应代码仓URL根据实际情况显示，代码仓分支为master，语言类型为cpp

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := codecheck.NewCodeCheckClient(
        codecheck.CodeCheckClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateTaskRequest{}
    request.ProjectId = "{project_id}"
    var listLanguagebody = []string{
        "cpp",
        "js",
    }
    var listCheckTypebody = []string{
        "source",
    }
    request.Body = &model.CreateTaskRequestV2{
        Language: listLanguagebody,
        GitBranch: "master",
        GitUrl: "git@code*****958.git",
        CheckType: &listCheckTypebody,
    }
    response, err := client.CreateTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.2 删除检查任务

功能介绍

删除检查任务，任务删除后无法再查看。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/tasks/{task_id}

表 4-9 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	参数解释： 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 4-10 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

Request succeeded!

状态码：400

表 4-11 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-12 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

```
DELETE https://{endpoint}/v2/tasks/{task_id}
```

响应示例

状态码：200

Request succeeded!

```
null
```

状态码：400

Bad Request

```
{
  "error_code" : "CC.00000000",
  "error_msg" : "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code" : "CC.00000003",
  "error_msg" : "认证信息过期"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;
import com.huaweicloud.sdk.codecheck.v2.*;
import com.huaweicloud.sdk.codecheck.v2.model.*;

public class DeleteTaskSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CodeCheckClient client = CodeCheckClient.newBuilder()
            .withCredential(auth)
            .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteTaskRequest request = new DeleteTaskRequest();
        request.withTaskId("{task_id}");
        try {
            DeleteTaskResponse response = client.deleteTask(request);
        }
    }
}
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcodecheck.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = CodeCheckClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteTaskRequest()
        request.task_id = "{task_id}"
        response = client.delete_task(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := codecheck.NewCodeCheckClient(
    codecheck.CodeCheckClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.DeleteTaskRequest{}
request.TaskId = "{task_id}"
response, err := client.DeleteTask(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.3 查询任务列表

功能介绍

根据project_id查询该项目下的任务列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/tasks

表 4-13 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	<p>参数解释: 项目的32位uuid，项目唯一标识，通过查询项目列表接口查询项目列表获取。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位项目ID。</p> <p>默认取值: 不涉及。</p>

表 4-14 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	<p>参数解释: 页码，类似page_num的作用。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 0-999999</p>
limit	否	Integer	<p>参数解释: 每页数量，类似page_size的作用。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 1-100</p>

请求参数

表 4-15 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

表 4-16 响应 Body 参数

参数	参数类型	描述
tasks	Array of SimpleTaskInfoV2 objects	<p>参数解释: 任务信息。</p> <p>取值范围: 不涉及。</p>
total	Integer	<p>参数解释: 项目下的任务总数。</p> <p>取值范围: 不涉及。</p>

表 4-17 SimpleTaskInfoV2

参数	参数类型	描述
task_id	String	参数解释: 任务下的默认分支任务的id。 取值范围: 不涉及。
task_name	String	参数解释: 任务名称。 取值范围: 不涉及。
creator_id	String	参数解释: 任务创建者的用户id。 取值范围: 不涉及。
git_url	String	参数解释: 代码仓地址。 取值范围: 不涉及。
git_branch	String	参数解释: 任务的默认分支，非代码仓默认分支。 取值范围: 不涉及。
created_at	String	参数解释: 任务创建时间。 取值范围: UTC格式。
last_check_time	String	参数解释: 任务下所有分支任务、增量任务的最近检查时间。 取值范围: UTC格式。

状态码：400

表 4-18 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码: 401

表 4-19 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

GET https://{endpoint}/v2/{task_id}/tasks

响应示例

状态码: 200

Request succeeded!

```
{
  "tasks": [ {
    "task_id": "435b58e*****d01c94a6",
    "task_name": "CSharp_02201119",
    "creator_id": "ecadebb5041***9780f3d905e20",
    "git_url": "git@*****_02201119.git",
    "git_branch": "master",
    "created_at": "2020-02-20 11:21:11",
    "last_check_time": "2020-02-20 11:37:46"
  }, {
    "task_id": "435b58343tg5g36907c1384d01c94a6",
    "task_name": "CSharp_066",
    "creator_id": "ecadebb3664*****",
    "git_url": "git@****_02201119.git",
    "git_branch": "master",
    "created_at": "2020-02-20 11:21:11",
    "last_check_time": "2020-02-20 11:37:46"
  } ],
  "total": 2
}
```

状态码: 400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙, 请稍后再试"
}
```

状态码: 401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.4 执行检查任务

功能介绍

执行检查任务。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/tasks/{task_id}/run

表 4-20 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

请求参数

表 4-21 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

表 4-22 请求 Body 参数

参数	是否必选	参数类型	描述
username	否	String	废弃参数。
access_token	否	String	废弃参数。
git_url	否	String	废弃参数。

参数	是否必选	参数类型	描述
git_branch	否	String	废弃参数。
ref	否	String	<p>参数解释: 代码仓拉取mr代码的方式，推荐在检查阶段拉取mr代码失败时使用此参数进行拉取方式的指定。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> pull: 以pull/xx/xx方式进行拉取。 merge_request: 以merge_request/xx/xx方式进行拉取。 <p>默认取值: 不传时，后台自动识别代码仓类型以进行mr代码的拉取，gitee仓以pull/xx/xx方式进行拉取，非gitee仓以merge_request/xx/xx方式进行拉取。</p>

响应参数

状态码：200

表 4-23 响应 Body 参数

参数	参数类型	描述
exec_id	String	<p>参数解释: 1次代码检查对应的执行记录id。</p> <p>取值范围: 32位UUID。</p>

状态码：400

表 4-24 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误描述

状态码：401

表 4-25 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

执行检查任务：如需自行指定增量代码下载方式，则传递参数ref

```
POST https://{endpoint}/v2/tasks/{task_id}/run
{
  "ref": "merge_request"
}
```

响应示例

状态码：200

Request succeeded!

```
{
  "exec_id": "d163535d0*****81d6a7654443e"
}
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.5 终止检查任务

功能介绍

根据任务ID终止检查任务。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/tasks/{task_id}/stop

表 4-26 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID

请求参数

表 4-27 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码：200

Request succeeded!

状态码：400

表 4-28 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-29 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

```
POST https://{endpoint}/v2/tasks/2b31ed520xxxxxbedb6e57xxxxxxx/stop
```

响应示例

状态码：200

Request succeeded!

```
null
```

状态码：400

Bad Request

```
{  
  "error_code": "CC.00000000",  
  "error_msg": "网络繁忙, 请稍后再试"  
}
```

状态码：401

Unauthorized

```
{  
  "error_code": "CC.00000003",  
  "error_msg": "认证信息过期"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;
import com.huaweicloud.sdk.codecheck.v2.*;
import com.huaweicloud.sdk.codecheck.v2.model.*;

public class StopTaskByIdSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CodeCheckClient client = CodeCheckClient.newBuilder()
            .withCredential(auth)
            .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
            .build();
        StopTaskByIdRequest request = new StopTaskByIdRequest();
        request.withTaskId("{task_id}");
        try {
            StopTaskByIdResponse response = client.stopTaskById(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcodecheck.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]

credentials = BasicCredentials(ak, sk)

client = CodeCheckClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = StopTaskByIdRequest()
    request.task_id = "{task_id}"
    response = client.stop_task_by_id(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := codecheck.NewCodeCheckClient(
        codecheck.CodeCheckClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.StopTaskByIdRequest{}
    request.TaskId = "{task_id}"
    response, err := client.StopTaskById(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.6 查询任务执行状态

功能介绍

根据任务ID查询任务执行状态。任务状态：0表示检查中，1表示检查失败，2表示检查成功，3表示任务中止。只有正在检查中才有进度的详细信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/tasks/{task_id}/progress

表 4-30 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

请求参数

表 4-31 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

表 4-32 响应 Body 参数

参数	参数类型	描述
task_status	Integer	<p>参数解释: 检查执行状态。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • 0: 检查中。 • 1: 检查失败。 • 2: 检查成功。 • 3: 任务中止。 • null: 未检查。
progress	ProgressDetailV2 object	<p>参数解释: 检查执行进度。</p> <p>取值范围: 不涉及。</p>

表 4-33 ProgressDetailV2

参数	参数类型	描述
ratio	String	<p>参数解释: 检查进度百分比。</p> <p>取值范围: 检查中、检查成功时, 为0%-100%; 检查失败、任务中止、未检查时, 为 null。</p>
info	String	<p>参数解释: 检查执行状态。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • running: 检查中。 • success: 检查成功。 • null: 检查失败、任务中止、未检查。

状态码: 400

表 4-34 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误描述

状态码：401

表 4-35 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

```
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxxebdb6e57xxxxxxx/progress
```

响应示例

状态码：200

Request succeeded!

```
{
  "task_status" : 0,
  "progress" : {
    "ratio" : "50%",
    "info" : "running"
  }
}
```

状态码：400

Bad Request

```
{
  "error_code" : "CC.00000000",
  "error_msg" : "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code" : "CC.00000003",
  "error_msg" : "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!

状态码	描述
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.7 历史扫描结果查询

功能介绍

查询某个任务在时间范围内每次扫描各个问题级别的问题数量。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/tasks/{task_id}/checkrecord

表 4-36 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释： 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制： 不涉及。</p> <p>取值范围： 不涉及。</p> <p>默认取值： 不涉及。</p>

参数	是否必选	参数类型	描述
project_id	是	String	<p>参数解释: 项目的32位uuid，项目唯一标识，通过查询项目列表接口查询项目列表获取。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位项目ID。</p> <p>默认取值: 不涉及。</p>

表 4-37 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	<p>参数解释: 页码，类似page_num的作用。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 0-999999</p>
limit	否	Integer	<p>参数解释: 每页数量，类似page_size的作用。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 1-1000</p>
start_time	否	String	<p>参数解释: 任务检查开始时间。使用UTC时间，格式为yyyy-MM-ddTHH:mm:ssZ。例如：2020-09-25T12:05:00Z。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 2000-01-01T00:00:00Z</p>

参数	是否必选	参数类型	描述
end_time	否	String	<p>参数解释: 任务检查结束时间。使用UTC时间，格式为yyyy-MM-ddTHH:mm:ssZ。例如：2020-09-25T12:05:00Z。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 当前时间。</p>

请求参数

表 4-38 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

表 4-39 响应 Body 参数

参数	参数类型	描述
data	Array of CheckRecordDataInfo objects	参数解释: 每次扫描的各个问题级别的问题数量。 取值范围: 不涉及。
total	Integer	参数解释: 扫描次数总数。 取值范围: 不涉及。

表 4-40 CheckRecordDataInfo

参数	参数类型	描述
jobId	String	参数解释: 执行ID。版本级不传，门禁级必传。 约束限制: 不涉及。 取值范围: 不涉及。 默认取值: 不涉及。
check_time	String	参数解释: 检查任务执行开始时间，使用UTC时间，格式为yyyy-MM-ddTHH:mm:ssZ。 取值范围: 不涉及。
check_end_time	String	参数解释: 检查任务执行结束时间，使用UTC时间，格式为yyyy-MM-ddTHH:mm:ssZ。 取值范围: 不涉及。
issue_counts	CheckRecordIssueCountsInfo object	参数解释: 各个问题级别的问题数量。 取值范围: 不涉及。

表 4-41 CheckRecordIssueCountsInfo

参数	参数类型	描述
critical	Integer	参数解释: 致命问题数量。 取值范围: 不涉及。
serious	Integer	参数解释: 严重问题数量。 取值范围: 不涉及。
normal	Integer	参数解释: 一般问题数量。 取值范围: 不涉及。
prompt	Integer	参数解释: 提示问题数量。 取值范围: 不涉及。

状态码：400

表 4-42 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-43 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

在ID为e04b357xxxxx5ed832xxxxxx的项目下，查询ID为2b31ed520xxxxxebedb6e57xxxxxxx的任务在UTC时间2025-06-01T00:00:00Z到2025-06-10T00:00:00Z之间的所有扫描结果数据

```
GET https://{endpoint}/v2/363540xxxxxxxx5105099944xxxxxxxx/tasks/2b31ed520xxxxxebedb6e57xxxxxxx/checkrecord?start_time=2025-06-01T00:00:00Z&end_time=2025-06-10T00:00:00Z
null
```

响应示例

状态码：200

Request succeeded!

```
{
  "total": 100,
  "data": [ {
    "check_time": "2023-11-29T05:28:41Z",
    "check_end_time": "2023-11-29T05:30:38Z",
    "job_id": "84xxxxxxxxxxxxxxxxxxxxxxxxxxxx6f",
    "issue_counts": {
      "critical": 0,
      "serious": 1,
      "normal": 2,
      "prompt": 3
    }
  }, {
    "check_time": "2023-09-29T05:28:41Z",
    "check_end_time": "2023-09-29T05:30:38Z",
    "job_id": "a4xxxxxxxxxxxxxxxxxxxxxxxxxxxxb1",
    "issue_counts": {
      "critical": 0,
      "serious": 1,
      "normal": 2,
      "prompt": 3
    }
  }
]
}
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!

状态码	描述
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.8 查询任务的已选规则集列表 v2

功能介绍

查询任务的已选规则集列表v2。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/tasks/{task_id}/rulesets

表 4-44 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	<p>参数解释： 项目的32位uuid，项目唯一标识，通过查询项目列表接口查询项目列表获取。</p> <p>约束限制： 不涉及。</p> <p>取值范围： 32位项目ID。</p> <p>默认取值： 不涉及。</p>

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

表 4-45 Query 参数

参数	是否必选	参数类型	描述
needExtendInfo	否	String	<p>参数解释: 是否需要扩展信息：未启用语言对应的可选规则集、规则集的名称和使用状态。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • true：需要。 • false：不需要。

请求参数

表 4-46 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

表 4-47 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of TaskRulesetInfo objects	<p>参数解释: 规则集信息的数组。</p> <p>取值范围: 不涉及。</p>

表 4-48 TaskRulesetInfo

参数	参数类型	描述
template_id	String	<p>参数解释: 规则集ID。</p> <p>取值范围: 32位规则集ID。</p>

参数	参数类型	描述
language	String	参数解释: 规则集语言。 取值范围: 不涉及。
template_name	String	参数解释: 规则集名称。 取值范围: 不涉及。
type	String	参数解释: 规则集选取状态。 取值范围: <ul style="list-style-type: none"> selected: 已选。 optional: 可选, 指未进行检查语言在当前项目下的默认规则集。
status	String	参数解释: 规则集属性。 取值范围: <ul style="list-style-type: none"> 0: 用户规则集。 1: 系统默认规则集。

状态码: 400

表 4-49 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码: 401

表 4-50 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

GET https://{endpoint}/v2/{project_id}/tasks/{task_id}/rulesets

响应示例

状态码：200

Request succeeded!

```
[ {  
  "template_id" : "396e1c2511744f6fa199d33b26038edd",  
  "language" : "Java",  
  "template_name" : "Java_cmetrics",  
  "type" : "selected",  
  "status" : "0"  
}, {  
  "template_id" : "4e988c481b4c46d0a3297fbe343a662d",  
  "language" : "html",  
  "template_name" : "通用检查规则集",  
  "type" : "optional",  
  "status" : "1"  
}]
```

状态码：400

Bad Request

```
{  
  "error_code" : "CC.00000000",  
  "error_msg" : "网络繁忙，请稍后再试"  
}
```

状态码：401

Unauthorized

```
{  
  "error_code" : "CC.00000003",  
  "error_msg" : "认证信息过期"  
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.9 查询任务的已选规则集列表 v3

功能介绍

查询任务的已选规则集列表v3。

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/tasks/{task_id}/rulesets

表 4-51 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释: 项目的32位uuid, 项目唯一标识, 通过 查询项目列表 接口查询项目列表获取。 约束限制: 不涉及。 取值范围: 32位项目ID。 默认取值: 不涉及。
task_id	是	String	参数解释: 任务ID, 即新建检查任务接口的返回值, 每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。 约束限制: 不涉及。 取值范围: 不涉及。 默认取值: 不涉及。

表 4-52 Query 参数

参数	是否必选	参数类型	描述
needExtendInfo	否	String	<p>参数解释: 是否需要扩展信息：未启用语言对应的可选规则集、规则集的名称和使用状态。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • true: 需要。 • false: 不需要。
offset	否	Integer	<p>参数解释: 页码，类似page_num的作用。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 0-999999</p>
limit	否	Integer	<p>参数解释: 每页数量，类似page_size的作用。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 1-1000</p>

请求参数

表 4-53 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

表 4-54 响应 Body 参数

参数	参数类型	描述
data	Array of TaskRulesetInfo objects	<p>参数解释: 规则集信息。</p> <p>取值范围: 不涉及。</p>
total	Integer	<p>参数解释: 总数。</p> <p>取值范围: 不涉及。</p>

表 4-55 TaskRulesetInfo

参数	参数类型	描述
template_id	String	参数解释: 规则集ID。 取值范围: 32位规则集ID。
language	String	参数解释: 规则集语言。 取值范围: 不涉及。
template_name	String	参数解释: 规则集名称。 取值范围: 不涉及。
type	String	参数解释: 规则集选取状态。 取值范围: <ul style="list-style-type: none"> selected: 已选。 optional: 可选, 指未进行检查语言在当前项目下的默认规则集。
status	String	参数解释: 规则集属性。 取值范围: <ul style="list-style-type: none"> 0: 用户规则集。 1: 系统默认规则集。

状态码: 400

表 4-56 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码: 401

表 4-57 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

GET https://{endpoint}/v3/{project_id}/tasks/{task_id}/rulesets

响应示例

状态码：200

Request succeeded!

```
{
  "total": 1,
  "data": [{
    "template_id": "111111xxxx3246",
    "language": "java",
    "type": "selected",
    "template_name": "通用检查规则集",
    "status": "1"
  }]
}
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.10 查询任务规则集的检查参数 v2

功能介绍

查询任务规则集的检查参数v2

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/tasks/{task_id}/ruleset/{ruleset_id}/check-parameters

表 4-58 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，每个CodeArts首页下的项目都有1个唯一的项目ID
task_id	是	String	任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID
ruleset_id	是	String	规则集ID

表 4-59 Query 参数

参数	是否必选	参数类型	描述
language	是	String	规则集语言

请求参数

表 4-60 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码：200

表 4-61 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of TaskCheckParameters objects	

表 4-62 TaskCheckParameters

参数	参数类型	描述
check_id	Integer	检查工具ID
name	String	编译参数名称
checker_configs	Array of CheckConfigInfo objects	检查参数配置信息

表 4-63 CheckConfigInfo

参数	参数类型	描述
name	String	检查参数名称
cfg_key	String	检查参数对应的key值
default_value	String	检查参数默认值
option_value	String	检查参数可选项
is_required	Integer	0: 非必填, 1: 必填
description	String	检查参数说明
type	Integer	参数类型, 0: 文本, 2: 有可选项
status	String	参数状态, on: 启用, off: 未启用
value	String	检查参数值

状态码：400

表 4-64 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码: 401

表 4-65 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

GET https://{endpoint}/v2/{project_id}/tasks/{task_id}/ruleset/{ruleset_id}/check-parameters

响应示例

状态码: 200

Request succeeded!

```
[{
  "check_id": 0,
  "name": "compile_tool",
  "checker_configs": [{
    "value": "",
    "name": "编译命令",
    "cfg_key": "",
    "default_value": "",
    "option_value": "",
    "is_required": 0,
    "description": "编译命令",
    "type": 0,
    "status": "on"
  }]
}]
```

状态码: 400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙, 请稍后再试"
}
```

状态码: 401

Unauthorized

```
{
  "error_code": "CC.00000003",
```

```
"error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.11 查询任务规则集的检查参数 v3

功能介绍

查询任务规则集的检查参数v3

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project_id}/tasks/{task_id}/ruleset/{ruleset_id}/check-parameters

表 4-66 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，每个CodeArts首页下的项目都有1个唯一的项目ID
task_id	是	String	任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID
ruleset_id	是	String	规则集ID

表 4-67 Query 参数

参数	是否必选	参数类型	描述
language	是	String	规则集语言

参数	是否必选	参数类型	描述
offset	否	Integer	分页索引, 偏移量, 非必填
limit	否	Integer	每页显示的数量, 非必填

请求参数

表 4-68 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码：200

表 4-69 响应 Body 参数

参数	参数类型	描述
data	Array of TaskCheckParameters objects	历史记录数据
total	Integer	总数

表 4-70 TaskCheckParameters

参数	参数类型	描述
check_id	Integer	检查工具ID
name	String	编译参数名称
checker_configs	Array of CheckConfigInfo objects	检查参数配置信息

表 4-71 CheckConfigInfo

参数	参数类型	描述
name	String	检查参数名称
cfg_key	String	检查参数对应的key值
default_value	String	检查参数默认值
option_value	String	检查参数可选项
is_required	Integer	0: 非必填, 1: 必填
description	String	检查参数说明
type	Integer	参数类型, 0: 文本, 2: 有可选项
status	String	参数状态, on: 启用, off: 未启用
value	String	检查参数值

状态码: 400

表 4-72 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码: 401

表 4-73 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

```
GET https://{endpoint}/v3/{project_id}/tasks/{task_id}/ruleset/{ruleset_id}/check-parameters
```

响应示例

状态码: 200

Request succeeded!

```
{
  "total": 1,
```

```
"data": [{
  "check_id": 0,
  "name": "compile_tool",
  "checker_configs": [{
    "value": "",
    "name": "编译命令",
    "cfg_key": "",
    "default_value": "",
    "option_value": "",
    "is_required": 0,
    "description": "编译命令",
    "type": 0,
    "status": "on"
  }]
}]
}
```

状态码: 400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙, 请稍后再试"
}
```

状态码: 401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.12 配置任务检查参数

功能介绍

基于规则集配置对应的任务配置检查参数

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/tasks/{task_id}/config-parameters

表 4-74 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	<p>参数解释: 项目的32位uuid, 项目唯一标识, 通过查询项目列表接口查询项目列表获取。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位项目ID。</p> <p>默认取值: 不涉及。</p>
task_id	是	String	<p>参数解释: 任务ID, 即新建检查任务接口的返回值, 每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

请求参数

表 4-75 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

表 4-76 请求 Body 参数

参数	是否必选	参数类型	描述
check_id	是	Integer	<p>参数解释: 参数已废弃，默认填写0即可。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
ruleset_id	是	String	<p>参数解释: 规则集ID，通过查询规则集列表接口可获取到规则集ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位规则集ID。</p> <p>默认取值: 不涉及。</p>

参数	是否必选	参数类型	描述
language	是	String	<p>参数解释: 规则集对应语言，通过查询规则集列表接口可获取到规则集ID对应的语言。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 与ruleset_id规则集的语言对应。</p> <p>默认取值: 不涉及。</p>
status	是	String	<p>参数解释: 当前规则集对应的编译配置总开关。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • off: 关闭。 • on: 打开。 <p>默认取值: 不涉及。</p>
task_check_settings	是	Array of TaskCheckSettingsItem objects	<p>参数解释: 数组类型，当前规则集对应编译配置具体详情，可基于web页面查看当前规则集所涉及的编译配置，多个参数配置多条即可。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

表 4-77 TaskCheckSettingsItem

参数	是否必选	参数类型	描述
cfg_key	是	String	<p>参数解释: 规则集对应的编译配置条件键。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • compileTool: 编译工具。 • buildToolVersion: 编译工具版本。 • compileScriptCmd: 编译命令。 <p>默认取值: 不涉及。</p>
status	是	String	<p>参数解释: 当前检查参数key对应的开启状态, 填写"on"即可。</p> <p>约束限制: 不涉及。</p> <p>取值范围: "on"。</p> <p>默认取值: 不涉及</p>
cfg_value	否	String	<p>参数解释: 规则集对应的编译配置值。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 可选值可以在web页面-任务设置-规则集-编译配置直接获取。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码: 200

Request succeeded!

状态码: 400

表 4-78 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码: 401

表 4-79 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

任务配置编译参数: 语言类型为java, 状态为on

POST https://{endpoint}/v2/{project_id}/tasks/{task_id}/config-parameters

```
{
  "check_id" : 0,
  "ruleset_id" : "b113bxxxxxxxx5738bee86ecxxxxxxxx",
  "language" : "java",
  "status" : "on",
  "task_check_settings" : [ {
    "cfg_value" : "maven",
    "cfg_key" : "compileTool",
    "status" : "on"
  }, {
    "cfg_value" : "maven3.8.5-jdk17",
    "cfg_key" : "buildToolVersion",
    "status" : "on"
  }, {
    "cfg_value" : "mvn clean package -Dmaven.test.skip=true -U -e -X -B",
    "cfg_key" : "compileScriptCmd",
    "status" : "on"
  }
  ]
}
```

响应示例

状态码: 200

Request succeeded!

null

状态码: 400

Bad Request

```
{
  "error_code" : "CC.00000000",
```

```
"error_msg": "网络繁忙，请稍后再试"  
}
```

状态码：401

Unauthorized

```
{  
  "error_code": "CC.00000003",  
  "error_msg": "认证信息过期"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

任务配置编译参数：语言类型为java，状态为on

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;  
import com.huaweicloud.sdk.codecheck.v2.*;  
import com.huaweicloud.sdk.codecheck.v2.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class ListTaskParameterSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new BasicCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        CodeCheckClient client = CodeCheckClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListTaskParameterRequest request = new ListTaskParameterRequest();  
        request.withProjectId("{project_id}");  
        request.withTaskId("{task_id}");  
        ConfigTaskParameterBody body = new ConfigTaskParameterBody();  
        List<TaskCheckSettingsItem> listbodyTaskCheckSettings = new ArrayList<>();  
        listbodyTaskCheckSettings.add(  
            new TaskCheckSettingsItem()  
                .withCfgKey("compileTool")  
                .withStatus("on")  
                .withCfgValue("maven")  
        );  
        listbodyTaskCheckSettings.add(  
            new TaskCheckSettingsItem()  
                .withCfgKey("buildToolVersion")
```

```
        .withStatus("on")
        .withCfgValue("maven3.8.5-jdk17")
    );
    listbodyTaskCheckSettings.add(
        new TaskCheckSettingsItem()
            .withCfgKey("compileScriptCmd")
            .withStatus("on")
            .withCfgValue("mvn clean package -Dmaven.test.skip=true -U -e -X -B")
    );
    body.withTaskCheckSettings(listbodyTaskCheckSettings);
    body.withStatus("on");
    body.withLanguage("java");
    body.withRulesetId("b113bxxxxxx5738bee86ecxxxxxx");
    body.withCheckId(0);
    request.withBody(body);
    try {
        ListTaskParameterResponse response = client.listTaskParameter(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

任务配置编译参数：语言类型为java，状态为on

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcodecheck.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = CodeCheckClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListTaskParameterRequest()
        request.project_id = "{project_id}"
        request.task_id = "{task_id}"
        listTaskCheckSettingsbody = [
            TaskCheckSettingsItem(
                cfg_key="compileTool",
                status="on",
                cfg_value="maven"
            )
        ]
```

```
    ),
    TaskCheckSettingsItem(
        cfg_key="buildToolVersion",
        status="on",
        cfg_value="maven3.8.5-jdk17"
    ),
    TaskCheckSettingsItem(
        cfg_key="compileScriptCmd",
        status="on",
        cfg_value="mvn clean package -Dmaven.test.skip=true -U -e -X -B"
    )
]
request.body = ConfigTaskParameterBody(
    task_check_settings=listTaskCheckSettingsbody,
    status="on",
    language="java",
    ruleset_id="b113bxxxxxxxx5738bee86ecxxxxxxxx",
    check_id=0
)
response = client.list_task_parameter(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

任务配置编译参数：语言类型为java，状态为on

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := codecheck.NewCodeCheckClient(
        codecheck.CodeCheckClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListTaskParameterRequest{}
    request.ProjectId = "{project_id}"
    request.TaskId = "{task_id}"
    cfgValueTaskCheckSettings:= "maven"
    cfgValueTaskCheckSettings1:= "maven3.8.5-jdk17"
    cfgValueTaskCheckSettings2:= "mvn clean package -Dmaven.test.skip=true -U -e -X -B"
    var listTaskCheckSettingsbody = []model.TaskCheckSettingsItem{
        {
```

```
    CfgKey: "compileTool",
    Status: "on",
    CfgValue: &cfgValueTaskCheckSettings,
  },
  {
    CfgKey: "buildToolVersion",
    Status: "on",
    CfgValue: &cfgValueTaskCheckSettings1,
  },
  {
    CfgKey: "compileScriptCmd",
    Status: "on",
    CfgValue: &cfgValueTaskCheckSettings2,
  },
}
request.Body = &model.ConfigTaskParameterBody{
  TaskCheckSettings: listTaskCheckSettingsbody,
  Status: "on",
  Language: "java",
  RulesetId: "b113bxxxxxxx5738bee86ecxxxxxxx",
  CheckId: int32(0),
}
response, err := client.ListTaskParameter(request)
if err == nil {
  fmt.Printf("%+v\n", response)
} else {
  fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.13 修改任务规则集

功能介绍

修改任务规则集。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/tasks/{task_id}/ruleset

表 4-80 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

请求参数

表 4-81 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

表 4-82 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of UpdateTaskRulesetItem objects	参数解释: 规则集信息。 约束限制: 不涉及。

表 4-83 UpdateTaskRulesetItem

参数	是否必选	参数类型	描述
language	是	String	参数解释: 检查语言，支持的语言请关注服务更新特性文档。 约束限制: 不涉及。 取值范围: 不涉及。 默认取值: 不涉及。
rule_set_id	是	String	参数解释: 规则集ID。 约束限制: 32位规则集ID。 取值范围: 不涉及。 默认取值: 不涉及。
if_use	是	String	参数解释: 是否启用。 约束限制: 不涉及。 取值范围: <ul style="list-style-type: none"> • 0: 不启用。 • 1: 启用。 默认取值: 不涉及。

参数	是否必选	参数类型	描述
status	是	String	参数解释: 新/老数据。 约束限制: 不涉及。 取值范围: 1 默认取值: 不涉及。

响应参数

状态码: 200

Request succeeded!

状态码: 400

表 4-84 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码: 401

表 4-85 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

修改任务的规则集：规则集的语言类型为cpp，规则设置ID根据实际显示，规则状态为1

PUT https://{endpoint}/v2/tasks/{task_id}/ruleset

```
[ {
  "language": "cpp",
  "rule_set_id": "7be4cfxxxxxxxxfd8d07077cxxxxxxxx",
  "if_use": "1",
```

```
"status" : "1"  
}]
```

响应示例

状态码：200

Request succeeded!

```
null
```

状态码：400

Bad Request

```
{  
  "error_code" : "CC.00000000",  
  "error_msg" : "网络繁忙，请稍后再试"  
}
```

状态码：401

Unauthorized

```
{  
  "error_code" : "CC.00000003",  
  "error_msg" : "认证信息过期"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改任务的规则集：规则集的语言类型为cpp，规则设置ID根据实际显示，规则状态为1

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;  
import com.huaweicloud.sdk.codecheck.v2.*;  
import com.huaweicloud.sdk.codecheck.v2.model.*;  
  
import java.util.List;  
import java.util.ArrayList;  
  
public class UpdateTaskRulesetSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new BasicCredentials()  
            .withAk(ak)  
            .withSk(sk);
```

```
CodeCheckClient client = CodeCheckClient.newBuilder()
    .withCredential(auth)
    .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateTaskRulesetRequest request = new UpdateTaskRulesetRequest();
request.withTaskId("{task_id}");
List<UpdateTaskRulesetItem> listbodyUpdateTaskRuleset = new ArrayList<>();
listbodyUpdateTaskRuleset.add(
    new UpdateTaskRulesetItem()
        .withLanguage("cpp")
        .withRuleSetId("7be4cfxxxxxxxxfd8d07077cxxxxxxxx")
        .withIfUse("1")
        .withStatus("1")
);
request.withBody(listbodyUpdateTaskRuleset);
try {
    UpdateTaskRulesetResponse response = client.updateTaskRuleset(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

修改任务的规则集：规则集的语言类型为cpp，规则设置ID根据实际显示，规则状态为1

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcodecheck.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = CodeCheckClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateTaskRulesetRequest()
        request.task_id = "{task_id}"
        listUpdateTaskRulesetbody = [
            UpdateTaskRulesetItem(
                language="cpp",
                rule_set_id="7be4cfxxxxxxxxfd8d07077cxxxxxxxx",
            )
        ]
        request.with_body(listUpdateTaskRulesetbody)
        response = client.update_task_ruleset(request)
        print(response.to_dict())
    except exceptions.ApiError as e:
        print(e.to_dict())
```

```
        if_use="1",
        status="1"
    )
]
request.body = listUpdateTaskRulesetbody
response = client.update_task_ruleset(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

修改任务的规则集：规则集的语言类型为cpp，规则设置ID根据实际显示，规则状态为1

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := codecheck.NewCodeCheckClient(
        codecheck.CodeCheckClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateTaskRulesetRequest{}
    request.TaskId = "{task_id}"
    var listUpdateTaskRulesetbody = []model.UpdateTaskRulesetItem{
        {
            Language: "cpp",
            RuleSetId: "7be4cfxxxxxxxxfd8d07077cxxxxxxxx",
            IfUse: "1",
            Status: "1",
        },
    }
    request.Body = &listUpdateTaskRulesetbody
    response, err := client.UpdateTaskRuleset(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.14 查询任务检查失败日志

功能介绍

查询任务检查失败日志，不传execute_id则查询最近一次的检查日志。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/tasks/{task_id}/log-detail

表 4-86 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释： 项目的32位uuid，项目唯一标识，通过 查询项目列表 接口查询项目列表获取。 约束限制： 不涉及。 取值范围： 32位项目ID。 默认取值： 不涉及。

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

表 4-87 Query 参数

参数	是否必选	参数类型	描述
execute_id	否	String	<p>参数解释: 执行记录ID，即每一次执行版本级或门禁级扫描的唯一ID。通过执行检查任务接口的返回值可获取执行记录ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位UUID。</p> <p>默认取值: 无，不传时查询任务的最近一次的检查日志。</p>

请求参数

表 4-88 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

表 4-89 响应 Body 参数

参数	参数类型	描述
param_info	ParamInfo object	<p>参数解释: 任务基本信息。</p> <p>取值范围: 不涉及。</p>
log_info	Array of LogInfo objects	<p>参数解释: 失败日志详情。</p> <p>取值范围: 不涉及。</p>

表 4-90 ParamInfo

参数	参数类型	描述
url	String	参数解释: 代码仓地址。 取值范围: 不涉及。
branch	String	参数解释: 任务本次扫描的代码仓分支。 取值范围: 不涉及。
language	String	参数解释: 任务本次扫描的规则语言。 取值范围: 不涉及。
exclude_dir	String	参数解释: 任务配置的排除目录信息。 取值范围: 不涉及。
encode	String	参数解释: 废弃字段。 取值范围: 不涉及。
compile_config	String	参数解释: 任务的部分检查参数json格式信息。 取值范围: 不涉及。
rule_template	String	参数解释: 废弃字段。 取值范围: 不涉及。

表 4-91 LogInfo

参数	参数类型	描述
display_name	String	参数解释: 检查日志页面所显示的子任务阶段名称。 取值范围: 不涉及。
log	String	参数解释: 扫描日志内容。 取值范围: 不涉及。
level	String	参数解释: 废弃字段。 取值范围: 不涉及。
analysis	String	参数解释: 任务失败的简要原因。 取值范围: 不涉及。
faq	String	参数解释: 废弃字段。 取值范围: 不涉及。

状态码：400

表 4-92 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-93 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

GET https://{endpoint}/v2/{project_id}/tasks/{task_id}/log-detail

响应示例

状态码：200

Request succeeded!

```
{
  "param_info": {
    "url": "git@xxxxxx_only00001/file.git",
    "branch": "master",
    "language": "JAVA",
    "exclude_dir": "aa",
    "encode": "",
    "compile_config": "",
    "rule_template": ""
  },
  "log_info": [ {
    "log": "2023-01-04 14:31:33 Running on server:10.75.***.***\n2023-01-04 14:31:34 waiting subJob to execute\n2023-01-04 14:33:38 flush redis cache successfully!\n2023-01-04 14:33:38 all subJob finish!",
    "level": "",
    "analysis": "2023-01-04 14:31:33 Running on server:10.75.***.***\n2023-01-04 14:31:34 waiting subJob to execute\n2023-01-04 14:33:38 flush redis cache successfully!\n2023-01-04 14:33:38 all subJob finish!",
    "faq": "",
    "display_name": "sonarqube"
  } ]
}
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!

状态码	描述
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.15 获取任务的目录树

功能介绍

获取任务的目录树

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/tasks/{task_id}/listpathtree

表 4-94 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，每个CodeArts首页下的项目都有1个唯一的项目ID
task_id	是	String	任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID

表 4-95 Query 参数

参数	是否必选	参数类型	描述
current_path	否	String	目录或文件的路径
offset	否	Integer	分页索引，偏移量
limit	否	Integer	每页显示的数量,每页最多显示1000条

请求参数

表 4-96 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码：200

表 4-97 响应 Body 参数

参数	参数类型	描述
info	Array of TreeNode objects	任务的目录树信息
total	Integer	数目

表 4-98 TreeNode

参数	参数类型	描述
file_name	String	目录或文件名
file_path	String	目录或文件路径
is_leaf	Boolean	是否为叶子节点，true是，false不是
checkbox_status	String	屏蔽状态，包括unchecked(不屏蔽)、all(全屏蔽)、half(半屏蔽)

状态码：400

表 4-99 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-100 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

```
GET https://{endpoint}/v2/{project_id}/tasks/{task_id}/listpathtree
```

响应示例

状态码：200

Request succeeded!

```
{
  "info": [ {
    "file_name": ".LAST_RELEASE",
    "file_path": ".LAST_RELEASE",
    "is_leaf": true,
    "checkbox_status": "unchecked"
  } ],
  "total": 1
}
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;
import com.huaweicloud.sdk.codecheck.v2.*;
import com.huaweicloud.sdk.codecheck.v2.model.*;
```

```
public class ShowTaskPathTreeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CodeCheckClient client = CodeCheckClient.newBuilder()
            .withCredential(auth)
            .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowTaskPathTreeRequest request = new ShowTaskPathTreeRequest();
        request.withProjectId("{project_id}");
        request.withTaskId("{task_id}");
        try {
            ShowTaskPathTreeResponse response = client.showTaskPathTree(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcodecheck.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk)

    client = CodeCheckClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowTaskPathTreeRequest()
```

```

request.project_id = "{project_id}"
request.task_id = "{task_id}"
response = client.show_task_path_tree(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

```

Go

```

package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := codecheck.NewCodeCheckClient(
        codecheck.CodeCheckClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowTaskPathTreeRequest{}
    request.ProjectId = "{project_id}"
    request.TaskId = "{task_id}"
    response, err := client.ShowTaskPathTree(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}

```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Request succeeded!
400	Bad Request

状态码	描述
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.16 任务配置屏蔽目录

功能介绍

任务配置屏蔽目录。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/tasks/{task_id}/config-ignorepath

表 4-101 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	<p>参数解释： 项目的32位uuid，项目唯一标识，通过查询项目列表接口查询项目列表获取。</p> <p>约束限制： 不涉及。</p> <p>取值范围： 32位项目ID。</p> <p>默认取值： 不涉及。</p>

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

请求参数

表 4-102 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

表 4-103 请求 Body 参数

参数	是否必选	参数类型	描述
ignore_path_settings	是	Array of IgnorePathSettingItem objects	<p>参数解释: 屏蔽目录的节点信息列表。</p> <p>约束限制: 不涉及。</p>

表 4-104 IgnorePathSettingItem

参数	是否必选	参数类型	描述
file_path	是	String	<p>参数解释: 需要屏蔽的目录或文件路径。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
checkbox_status	是	String	<p>参数解释: 屏蔽状态。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • unchecked: 取消屏蔽 • all: 全部屏蔽 • half: 半屏蔽, 用于目录, 表示目录下部分子目录、文件需要屏蔽 <p>默认取值: 不涉及。</p>

响应参数

状态码: 200

Request succeeded!

状态码: 400

表 4-105 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码: 401

表 4-106 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

任务配置屏蔽目录：文件路径为.LAST_RELEASE，检查状态为all

```
POST https://{endpoint}/v2/{project_id}/tasks/{task_id}/config-ignorepath
{
  "ignore_path_settings": [ {
    "file_path": ".LAST_RELEASE",
    "checkbox_status": "all"
  } ]
}
```

响应示例

状态码：200

Request succeeded!

```
null
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

SDK 代码示例

SDK代码示例如下。

Java

任务配置屏蔽目录：文件路径为.LAST_RELEASE，检查状态为all

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;
import com.huaweicloud.sdk.codecheck.v2.*;
import com.huaweicloud.sdk.codecheck.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateIgnorePathSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CodeCheckClient client = CodeCheckClient.newBuilder()
            .withCredential(auth)
            .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateIgnorePathRequest request = new UpdateIgnorePathRequest();
        request.withProjectId("{project_id}");
        request.withTaskId("{task_id}");
        UpdateIgnorePathRequestBody body = new UpdateIgnorePathRequestBody();
        List<IgnorePathSettingItem> listbodyIgnorePathSettings = new ArrayList<>();
        listbodyIgnorePathSettings.add(
            new IgnorePathSettingItem()
                .withFilePath(".LAST_RELEASE")
                .withCheckboxStatus("all")
        );
        body.withIgnorePathSettings(listbodyIgnorePathSettings);
        request.withBody(body);
        try {
            UpdateIgnorePathResponse response = client.updateIgnorePath(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

任务配置屏蔽目录：文件路径为.LAST_RELEASE，检查状态为all

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcodecheck.v2 import *

if __name__ == "__main__":
```

```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]

credentials = BasicCredentials(ak, sk)

client = CodeCheckClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = UpdateIgnorePathRequest()
    request.project_id = "{project_id}"
    request.task_id = "{task_id}"
    listIgnorePathSettingsbody = [
        IgnorePathSettingItem(
            file_path=".LAST_RELEASE",
            checkbox_status="all"
        )
    ]
    request.body = UpdateIgnorePathRequestBody(
        ignore_path_settings=listIgnorePathSettingsbody
    )
    response = client.update_ignore_path(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

任务配置屏蔽目录：文件路径为.LAST_RELEASE，检查状态为all

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := codecheck.NewCodeCheckClient(
        codecheck.CodeCheckClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```

request := &model.UpdateIgnorePathRequest{}
request.ProjectId = "{project_id}"
request.TaskId = "{task_id}"
var listIgnorePathSettingsbody = []model.IgnorePathSettingItem{
    {
        FilePath: ".LAST_RELEASE",
        CheckboxStatus: "all",
    },
}
request.Body = &model.UpdateIgnorePathRequestBody{
    IgnorePathSettings: listIgnorePathSettingsbody,
}
response, err := client.UpdateIgnorePath(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}

```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.17 查询任务的高级配置

功能介绍

查询任务的高级配置

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/tasks/{task_id}/settings

表 4-107 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，每个CodeArts首页下的项目都有1个唯一的项目ID
task_id	是	String	任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID

表 4-108 Query 参数

参数	是否必选	参数类型	描述
config_list	否	String	配置项名称列表，当前可选：customImage,includePaths,authId,reviewData,taskName，不建议同时查询多个配置项
offset	否	Integer	分页索引，偏移量，非必填
limit	否	Integer	每页显示的数量，非必填

请求参数

表 4-109 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码：200

表 4-110 响应 Body 参数

参数	参数类型	描述
info	Array of TaskAdvancedSettings objects	高级配置的相关信息

参数	参数类型	描述
total	Integer	总数

表 4-111 TaskAdvancedSettings

参数	参数类型	描述
key	String	高级配置对应的名称
value	String	高级配置对应的取值
option_value	String	高级配置对应的可选项
description	String	高级配置对应的中文描述

状态码：400

表 4-112 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-113 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

GET https://{endpoint}/v2/{project_id}/tasks/{task_id}/settings

响应示例

状态码：200

Request succeeded!

```
{
  "info": [ {
    "key": "customImage",
    "value": "xxxxx",
```

```
"description": "自定义镜像",
"option_value": ""
}],
"total": 1
}
```

状态码：400

Bad Request

```
{
"error_code": "CC.00000000",
"error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
"error_code": "CC.00000003",
"error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.18 任务配置高级配置

功能介绍

任务配置高级配置，如自定义镜像

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/tasks/{task_id}/settings

表 4-114 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID，每个CodeArts首页下的项目都有1个唯一的项目ID
task_id	是	String	任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID

请求参数

表 4-115 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	设置媒体类型和编码格式

表 4-116 请求 Body 参数

参数	是否必选	参数类型	描述
task_advanced_settings	是	Array of TaskAdvancedSettingsItem objects	高级配置参数的相关信息

表 4-117 TaskAdvancedSettingsItem

参数	是否必选	参数类型	描述
key	是	String	高级配置对应的名称
value	是	String	高级配置对应的取值

响应参数

状态码：200

Request succeeded!

状态码：400

表 4-118 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-119 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

任务配置高级配置，如自定义镜像等

POST https://{endpoint}/v2/{project_id}/tasks/{task_id}/settings

```
{
  "task_advanced_settings": [{
    "key": "customImage",
    "value": "xxxx"
  }]
}
```

响应示例

状态码：200

Request succeeded!

null

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

SDK 代码示例

SDK代码示例如下。

Java

任务配置高级配置，如自定义镜像等

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;
import com.huaweicloud.sdk.codecheck.v2.*;
import com.huaweicloud.sdk.codecheck.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateTaskSettingsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CodeCheckClient client = CodeCheckClient.newBuilder()
            .withCredential(auth)
            .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateTaskSettingsRequest request = new UpdateTaskSettingsRequest();
        request.withProjectId("{project_id}");
        request.withTaskId("{task_id}");
        UpdateTaskSettingsRequestBody body = new UpdateTaskSettingsRequestBody();
        List<TaskAdvancedSettingsItem> listbodyTaskAdvancedSettings = new ArrayList<>();
        listbodyTaskAdvancedSettings.add(
            new TaskAdvancedSettingsItem()
                .withKey("customImage")
                .withValue("xxxx")
        );
        body.withTaskAdvancedSettings(listbodyTaskAdvancedSettings);
        request.withBody(body);
        try {
            UpdateTaskSettingsResponse response = client.updateTaskSettings(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}
```

Python

任务配置高级配置，如自定义镜像等

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcodecheck.v2 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
  
    credentials = BasicCredentials(ak, sk)  
  
    client = CodeCheckClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = UpdateTaskSettingsRequest()  
        request.project_id = "{project_id}"  
        request.task_id = "{task_id}"  
        listTaskAdvancedSettingsbody = [  
            TaskAdvancedSettingsItem(  
                key="customImage",  
                value="xxxx"  
            )  
        ]  
        request.body = UpdateTaskSettingsRequestBody(  
            task_advanced_settings=listTaskAdvancedSettingsbody  
        )  
        response = client.update_task_settings(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

任务配置高级配置，如自定义镜像等

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := codecheck.NewCodeCheckClient(
    codecheck.CodeCheckClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateTaskSettingsRequest{}
request.ProjectId = "{project_id}"
request.TaskId = "{task_id}"
var listTaskAdvancedSettingsbody = []model.TaskAdvancedSettingsItem{
    {
        Key: "customImage",
        Value: "xxxx",
    },
}
request.Body = &model.UpdateTaskSettingsRequestBody{
    TaskAdvancedSettings: listTaskAdvancedSettingsbody,
}
response, err := client.UpdateTaskSettings(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.19 获取代码仓所有文件列表

功能介绍

获取代码仓所有文件的列表

调用方法

请参见[如何调用API](#)。

URI

GET /v4/tasks/{task_id}/file-list

表 4-120 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及</p> <p>取值范围: 不涉及</p> <p>默认值: 不涉及</p>

请求参数

表 4-121 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务 获取用户Token 接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码：200

表 4-122 响应 Body 参数

参数	参数类型	描述
data	Array of strings	参数解释: 代码仓文件列表 取值范围: 不涉及
total	Integer	参数解释: 总数 取值范围: 0-2147483647

状态码：400

表 4-123 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-124 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

GET https://{endpoint}/v4/tasks/897acf5exxx48a5013cd/file-list

响应示例

状态码：200

Request succeeded!

```
{
  "data" : [ "README.md" ],
  "total" : 1
}
```

状态码：400

Bad Request

```
{  
  "error_code": "CC.00000000",  
  "error_msg": "网络繁忙，请稍后再试"  
}
```

状态码：401

Unauthorized

```
{  
  "error_code": "CC.00000003",  
  "error_msg": "认证信息过期"  
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.20 责任人精确匹配开关

功能介绍

查询责任人精确匹配开关

调用方法

请参见[如何调用API](#)。

URI

GET /v1/tasks/{task_id}/owner-matching-switch

表 4-125 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及</p> <p>取值范围: 不涉及</p> <p>默认值: 不涉及</p>

请求参数

表 4-126 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务 获取用户Token 接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码：200

表 4-127 响应 Body 参数

参数	参数类型	描述
result	String	<p>参数解释: 接口响应成功。</p> <p>约束限制: 不涉及。</p>

状态码：400

表 4-128 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-129 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

```
GET https://{endpoint}/v1/tasks/897acf5exxx48a5013cd/owner-matching-switch
```

响应示例

状态码：200

Request succeeded!

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

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!

状态码	描述
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.21 取消检查

功能介绍

取消检查。

调用方法

请参见[如何调用API](#)。

URI

PUT /v1/tasks/{task_id}/stop

表 4-130 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及</p> <p>取值范围: 不涉及</p> <p>默认值: 不涉及</p>

表 4-131 Query 参数

参数	是否必选	参数类型	描述
job_id	否	String	参数解释: 此次执行记录的job_id 约束限制: 不涉及 取值范围: 不涉及 默认值: 不涉及

请求参数

表 4-132 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。通过调用IAM服务 获取用户Token 接口获取（响应消息头中X-Subject-Token的值）。
operator	否	String	参数解释: 操作者 约束限制: 不涉及 取值范围: 不涉及 默认值: 不涉及

响应参数

状态码：200

表 4-133 响应 Body 参数

参数	参数类型	描述
result	String	参数解释: 接口响应成功。 约束限制: 不涉及。

状态码: 400

表 4-134 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码: 401

表 4-135 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

```
PUT https://{endpoint}/v1/tasks/897acf5exxx48a5013cd/stop
```

响应示例

状态码: 200

Request succeeded!

```
{
  "result": "任务取消成功"
}
```

状态码: 400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙, 请稍后再试"
}
```

状态码: 401

Unauthorized

```
{  
  "error_code": "CC.00000003",  
  "error_msg": "认证信息过期"  
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.1.22 根据任务 id 获取任务 WebhookUrl 以及开关状态

功能介绍

根据任务id获取任务WebhookUrl以及开关状态

调用方法

请参见[如何调用API](#)。

URI

GET /v4/tasks/{task_id}/task-webhook-info

表 4-136 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及</p> <p>取值范围: 不涉及</p> <p>默认值: 不涉及</p>

请求参数

无

响应参数

状态码：200

表 4-137 响应 Body 参数

参数	参数类型	描述
data	Array of TaskWebhookInfo objects	<p>参数解释: webhoo节点列表</p> <p>取值范围: 不涉及</p>
total	Integer	<p>参数解释: 总数</p> <p>取值范围: 0-2147483647</p>

表 4-138 TaskWebhookInfo

参数	参数类型	描述
id	String	参数解释: 主键32位UUID 取值范围: 不涉及
taskId	String	参数解释: 任务id 取值范围: 不涉及
webhookUrl	String	参数解释: 第三方平台推送消息的url 取值范围: 不涉及
status	String	参数解释: 推送消息开关,off表示关,on表示开 取值范围: on,off
type	String	参数解释: 第三方类型,比如: 钉钉、企业微信等 取值范围: 不涉及
reviewResultFilter	String	参数解释: 根据门禁结果过滤 取值范围: 不涉及
now	String	参数解释: 创建时间 取值范围: 不涉及
userIds	String	参数解释: 用户id列表 取值范围: 不涉及

参数	参数类型	描述
notificationMenu	String	参数解释: 需要通知的菜单项 取值范围: 不涉及
secret	String	参数解释: webhook加签需要的密钥 取值范围: 不涉及

状态码：400

表 4-139 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-140 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

GET https://{endpoint}/v4/tasks/ac26d65ef4ef45c2986611XXXXXXffa/task-webhook-info

响应示例

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{  
  "error_code": "CC.00000003",  
  "error_msg": "认证信息过期"  
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.2 缺陷管理

4.2.1 查询缺陷概要

功能介绍

根据检查任务ID查询缺陷结果的概要。包括问题概述、问题状态、圈复杂度、代码重复率等。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/tasks/{task_id}/defects-summary

表 4-141 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

表 4-142 Query 参数

参数	是否必选	参数类型	描述
job_id	否	String	<p>参数解释: 执行记录ID，即每一次执行版本级或门禁级扫描的唯一ID，主要用于查询门禁级任务。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
merge_id	否	String	<p>参数解释: 合并请求ID或Commit ID，主要用于查询门禁级任务。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

请求参数

表 4-143 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

表 4-144 响应 Body 参数

参数	参数类型	描述
taskId	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
task_name	String	<p>参数解释: 任务名称。</p> <p>取值范围: 不涉及。</p>

参数	参数类型	描述
creator_id	String	参数解释: 任务创建者用户ID。 取值范围: 不涉及。
git_url	String	参数解释: 代码仓地址。 取值范围: 不涉及。
git_branch	String	参数解释: 代码仓分支, 如果是MR模式, 为源分支。 取值范围: 不涉及。
last_check_time	String	参数解释: 上一次检查时间。 取值范围: 不涉及。
code_line_total	Integer	参数解释: 原始代码行数, 包含空行和注释行。 取值范围: 不涉及。
code_line	Integer	参数解释: 有效代码行数, 不包含空行和注释行。 取值范围: 不涉及。
code_quality	Number	参数解释: 代码质量。 取值范围: 不涉及。
issue_count	Integer	参数解释: 问题总数。 取值范围: 不涉及。

参数	参数类型	描述
risk_coefficient	Number	参数解释: 危险系数。 取值范围: 不涉及。
duplication_ratio	String	参数解释: 代码重复率。 取值范围: 不涉及。
complexity_count	Integer	参数解释: 圈复杂度总数。 取值范围: 不涉及。
duplicated_lines	Integer	参数解释: 重复行数总数。 取值范围: 不涉及。
comment_lines	Integer	参数解释: 注释行数。 取值范围: 不涉及。
comment_ratio	String	参数解释: 注释率。 取值范围: 不涉及。
duplicated_blocks	Integer	参数解释: 重复块个数。 取值范围: 不涉及。
last_exec_time	String	参数解释: 上次执行时间。 取值范围: 不涉及。
check_type	String	参数解释: 检查类型。 取值范围: 不涉及。

参数	参数类型	描述
created_at	String	参数解释: 创建时间。 取值范围: 不涉及。
cyclomatic_complexity_per_method	String	参数解释: 平均圈复杂度（方法）。 取值范围: 不涉及。
cyclomatic_complexity_per_file	String	参数解释: 平均圈复杂度（文件）。 取值范围: 不涉及。
critical_count	String	参数解释: 致命问题数。 取值范围: 不涉及。
major_count	String	参数解释: 严重问题数。 取值范围: 不涉及。
minor_count	String	参数解释: 一般问题数。 取值范围: 不涉及。
suggestion_count	String	参数解释: 提示问题数。 取值范围: 不涉及。
is_access	String	参数解释: 门禁质量是否通过。 取值范围: 不涉及。
review_result	String	参数解释: 门禁质量是否通过。 取值范围: 不涉及。

参数	参数类型	描述
trigger_type	String	参数解释: 任务触发方式。 取值范围: 不涉及。
file_duplication_ratio	String	参数解释: 文件重复率。 取值范围: 不涉及。
new_count	Integer	参数解释: 待处理问题数。 取值范围: 不涉及。
solve_count	Integer	参数解释: 已解决问题数。 取值范围: 不涉及。
dismissed_count	Integer	参数解释: 已忽略问题数。 取值范围: 不涉及。
duplicated_files	Integer	参数解释: 重复文件数量。 取值范围: 不涉及。
new_critical_count	String	参数解释: 新增致命问题数。 取值范围: 不涉及。
new_major_count	String	参数解释: 新增严重问题数。 取值范围: 不涉及。
new_minor_count	String	参数解释: 新增一般问题数。 取值范围: 不涉及。

参数	参数类型	描述
new_suggestion_count	String	参数解释: 新增提示问题数。 取值范围: 不涉及。

状态码: 400

表 4-145 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码: 401

表 4-146 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

查询ID为2b31ed520xxxxxbedb6e57xxxxxxx的任务的缺陷概要

```
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxbedb6e57xxxxxxx/defects-summary
```

```
null
```

响应示例

状态码: 200

Request succeeded!

```
{
  "task_id": "435b58ecf7f54c45907c1384d01c94a6",
  "task_name": "CSharp_02201119",
  "creator_id": "ecadebb5041146cc96f9780f3d905e20",
  "git_url": "git@***_02201119.git",
  "git_branch": "master",
  "last_check_time": "2020-02-20 11:37:46",
  "code_line_total": 20,
  "code_line": 156,
  "code_quality": 147.4,
  "issue_count": 3,
```

```

"risk_coefficient" : 23,
"duplication_ratio" : "0.0%",
"complexity_count" : 0,
"duplicated_lines" : 0,
"comment_lines" : 0,
"comment_ratio" : "0",
"duplicated_blocks" : 0,
"last_exec_time" : "2 分 28 秒",
"check_type" : "source",
"created_at" : "2020-02-20 11:21:11",
"cyclomatic_complexity_per_method" : "84.0",
"cyclomatic_complexity_per_file" : "112.0",
"critical_count" : 2,
"major_count" : 1,
"minor_count" : 1,
"suggestion_count" : 1,
"is_access" : 0,
"trigger_type" : 1,
"file_duplication_ratio" : "33.3%",
"new_count" : 0,
"solve_count" : 0,
"duplicated_files" : 0,
"new_critical_count" : "0",
"new_major_count" : "0",
"new_minor_count" : "0",
"new_suggestion_count" : "0"
}

```

状态码：400

Bad Request

```

{
  "error_code" : "CC.00000000",
  "error_msg" : "网络繁忙，请稍后再试"
}

```

状态码：401

Unauthorized

```

{
  "error_code" : "CC.00000003",
  "error_msg" : "认证信息过期"
}

```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.2.2 查询代码度量概要

功能介绍

根据检查任务ID查询代码度量概要

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/tasks/{task_id}/metrics-summary

表 4-147 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释： 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制： 不涉及。</p> <p>取值范围： 不涉及。</p> <p>默认取值： 不涉及。</p>
project_id	是	String	<p>参数解释： 项目的32位uuid，项目唯一标识，通过查询项目列表接口查询项目列表获取。</p> <p>约束限制： 不涉及。</p> <p>取值范围： 32位项目ID。</p> <p>默认取值： 不涉及。</p>

请求参数

表 4-148 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

表 4-149 响应 Body 参数

参数	参数类型	描述
task_id	String	<p>参数解释: 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
task_name	String	<p>参数解释: 任务名称。</p> <p>取值范围: 不涉及。</p>

参数	参数类型	描述
creator_id	String	参数解释: 任务创建者用户ID。 取值范围: 不涉及。
git_url	String	参数解释: 代码仓地址。 取值范围: 不涉及。
git_branch	String	参数解释: 代码仓分支。 取值范围: 不涉及。
last_check_time	String	参数解释: 上一次检查时间。格式 2025-07-21T03:38:53Z。 取值范围: 不涉及。
last_exec_time	String	参数解释: 上一次检查时间。格式 2025-07-21T03:38:53Z。 取值范围: 不涉及。
last_check_finish_time	String	参数解释: 上次执行结束时间。格式 2025-07-21T03:38:53Z。 取值范围: 不涉及。
check_type	String	参数解释: 检查类型。 取值范围: 不涉及。
created_at	String	参数解释: 任务创建时间。格式 2025-07-21T03:38:53Z。 取值范围: 不涉及。
metric_info	MetricInfo object	度量项

表 4-150 MetricInfo

参数	参数类型	描述
code_size	String	参数解释: 有效代码行数，不包含空行和注释行。 取值范围: 不涉及。
raw_lines	String	参数解释: 原始代码行数，包含空行和注释行。 取值范围: 不涉及。
methods_total	String	参数解释: 函数总数。 取值范围: 不涉及。
cyclomatic_complexity_total	String	参数解释: 圈复杂度总数。 取值范围: 不涉及。
cyclomatic_complexity_per_method	String	参数解释: 平均圈复杂度。 取值范围: 不涉及。
maximum_cyclomatic_complexity	String	参数解释: 最大圈复杂度。 取值范围: 不涉及。
huge_cyclomatic_complexity_total	String	参数解释: 超大圈复杂度函数总数。 取值范围: 不涉及。
huge_cyclomatic_complexity_ratio	String	参数解释: 超大圈复杂度函数占比。 取值范围: 不涉及。

参数	参数类型	描述
cca_cyclomatic_complexity_total	String	参数解释: CCA圈复杂度总数。 取值范围: 不涉及。
cca_cyclomatic_complexity_per_method	String	参数解释: CCA平均圈复杂度。 取值范围: 不涉及。
maximum_cca_cyclomatic_complexity	String	参数解释: CCA最大圈复杂度。 取值范围: 不涉及。
huge_cca_cyclomatic_complexity_total	String	参数解释: 超大CCA圈复杂度函数总数。 取值范围: 不涉及。
cyclomatic_complexity_adequacy	String	参数解释: 圈复杂度阈值遵从度。 取值范围: 不涉及。
maximum_depth	String	参数解释: 最大深度。 取值范围: 不涉及。
huge_depth_total	String	参数解释: 超大深度个数。 取值范围: 不涉及。
huge_depth_ratio	String	参数解释: 超大深度占比。 取值范围: 不涉及。
method_lines	String	参数解释: 函数总行数。 取值范围: 不涉及。

参数	参数类型	描述
lines_per_method	String	参数解释: 平均函数代码行。 取值范围: 不涉及。
huge_method_total	String	参数解释: 超大代码行函数数。 取值范围: 不涉及。
huge_method_ratio	String	参数解释: 超大代码行函数占比。 取值范围: 不涉及。
files_total	String	参数解释: 文件总数。 取值范围: 不涉及。
folders_total	String	参数解释: 目录总数。 取值范围: 不涉及。
lines_per_file	String	参数解释: 平均文件代码行。 取值范围: 不涉及。
huge_headerfile_total	String	参数解释: 超大头文件数。 取值范围: 不涉及。
huge_headerfile_ratio	String	参数解释: 超大头文件占比。 取值范围: 不涉及。
huge_non_headerfile_total	String	参数解释: 超大源文件数。 取值范围: 不涉及。

参数	参数类型	描述
huge_non_headerfile_ratio	String	参数解释: 超大源文件占比。 取值范围: 不涉及。
huge_folder_total	String	参数解释: 超大目录数。 取值范围: 不涉及。
huge_folder_ratio	String	参数解释: 超大目录占比。 取值范围: 不涉及。
file_duplication_total	String	参数解释: 重复文件数。 取值范围: 不涉及。
file_duplication_ratio	String	参数解释: 文件重复率。 取值范围: 不涉及。
non_hfile_duplication_total	String	参数解释: 重复源文件数。 取值范围: 不涉及。
non_hfile_duplication_ratio	String	参数解释: 源文件重复率。 取值范围: 不涉及。
code_duplication_total	String	参数解释: 代码重复数。 取值范围: 不涉及。
code_duplication_ratio	String	参数解释: 代码重复率。 取值范围: 不涉及。

参数	参数类型	描述
non_hfile_code_duplication_total	String	参数解释: 源文件代码重复数。 取值范围: 不涉及。
non_hfile_code_duplication_ratio	String	参数解释: 源文件代码重复率。 取值范围: 不涉及。
unsafe_functions_total	String	参数解释: 危险函数总数。 取值范围: 不涉及。
unsafe_functions_kloc	String	参数解释: 危险函数密度。 取值范围: 不涉及。
redundant_code_total	String	参数解释: 冗余代码数。 取值范围: 不涉及。
redundant_code_kloc	String	参数解释: 冗余代码密度。 取值范围: 不涉及。
warning_suppression_total	String	参数解释: 抑制告警数。 取值范围: 不涉及。
warning_suppression_kloc	String	参数解释: 抑制告警密度。 取值范围: 不涉及。

状态码: 400

表 4-151 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-152 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

在ID为e04b357xxxxx5ed832xxxxxx的项目下，查询ID为2b31ed520xxxxxebedb6e57xxxxxxx的任务的代码度量概要

```
GET https://{endpoint}/v2/e04b357xxxxx5ed832xxxxxx/tasks/2b31ed520xxxxxebedb6e57xxxxxxx/metrics-summary
null
```

响应示例

状态码：200

Request succeeded!

```
{
  "task_id": "435b58ecf7f54c45907c1384d01c94a6",
  "task_name": "CSharp_02201119",
  "creator_id": "ecadebb5041146cc96f9780f3d905e20",
  "git_url": "git@****_02201119.git",
  "git_branch": "master",
  "last_check_time": "2020-02-20 11:37:46",
  "last_exec_time": "2020-02-20 11:37:46",
  "last_check_finish_time": "2020-02-20 11:53:46",
  "check_type": "source",
  "created_at": "2020-02-20 11:21:11",
  "metric_info": {
    "code_size": "133426.00",
    "raw_lines": "182082.00",
    "methods_total": "6840.00",
    "cyclomatic_complexity_total": "27435.00",
    "cyclomatic_complexity_per_method": "7.43",
    "maximum_cyclomatic_complexity": "346.00",
    "huge_cyclomatic_complexity_total": "167.00",
    "huge_cyclomatic_complexity_ratio": "4.13",
    "cca_cyclomatic_complexity_total": "26693.00",
    "cca_cyclomatic_complexity_per_method": "7.25",
    "maximum_cca_cyclomatic_complexity": "342.00",
    "huge_cca_cyclomatic_complexity_total": "148.00",
    "cyclomatic_complexity_adequacy": "21.00",
```

```

"maximum_depth": "333.00",
"huge_depth_total": "8.85",
"huge_depth_ratio": "91546.00",
"method_lines": "24.91",
"lines_per_method": "8.04",
"huge_method_total": "12300",
"huge_method_ratio": "12300",
"files_total": "629.00",
"folders_total": "426.00",
"lines_per_file": "12300",
"huge_headerfile_total": "12300",
"huge_headerfile_ratio": "12300",
"huge_non_headerfile_total": "12300",
"huge_non_headerfile_ratio": "12300",
"huge_folder_total": "12300",
"huge_folder_ratio": "12300",
"file_duplication_total": "12300",
"file_duplication_ratio": "12300",
"non_hfile_duplication_total": "12300",
"non_hfile_duplication_ratio": "12300",
"code_duplication_total": "3.28",
"code_duplication_ratio": "11782.00",
"non_hfile_code_duplication_total": "8.59",
"non_hfile_code_duplication_ratio": "11782.00",
"unsafe_functions_total": "17.17",
"unsafe_functions_kloc": "0.00",
"redundant_code_total": "0.00",
"redundant_code_kloc": "674.00",
"warning_suppression_total": "17.00",
"warning_suppression_kloc": "0.45"
}
}

```

状态码: 400

Bad Request

```

{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙, 请稍后再试"
}

```

状态码: 401

Unauthorized

```

{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}

```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.2.3 查询缺陷详情

功能介绍

根据检查任务ID分页查询任务检查结果的缺陷列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/tasks/{task_id}/defects-detail

表 4-153 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	参数解释： 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

表 4-154 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	参数解释： 页码，类似page_num的作用。 约束限制： 不涉及。 取值范围： 0-999999

参数	是否必选	参数类型	描述
limit	否	Integer	<p>参数解释: 每页数量, 类似page_size的作用。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 1-200</p>
status_ids	否	String	<p>参数解释: 问题状态, 多个以","分隔。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • 0, 待处理。 • 1, 已解决+已修复。 • 2, 已忽略。 <p>默认取值: 0</p>
severity	否	String	<p>参数解释: 问题级别, 多个以","分隔。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • 0, 致命。 • 1, 严重。 • 2, 一般。 • 3, 提示。 <p>默认取值: 不涉及。</p>
delay_status	否	String	<p>参数解释: 延迟状态, 多个以","分隔。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • 0, 正式问题。 • 1, 延迟上线问题。 <p>默认取值: 不涉及。</p>

请求参数

表 4-155 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

表 4-156 响应 Body 参数

参数	参数类型	描述
defects	Array of DefectInfoV2 objects	<p>参数解释: 缺陷详情列表。</p>
total	Integer	<p>参数解释: 总数。</p> <p>取值范围: 不涉及。</p>

表 4-157 DefectInfoV2

参数	参数类型	描述
defect_id	String	参数解释: MergeKey, 问题ID。 取值范围: 不涉及。
defect_checker_name	String	参数解释: 规则名称。 取值范围: 不涉及。
defect_status	String	参数解释: 问题状态。 取值范围: <ul style="list-style-type: none">• 0, 待处理。• 1, 已解决+已修复。• 2, 已忽略。
rule_system_tags	String	参数解释: 标签, 多个标签用逗号隔开。 取值范围: 不涉及。
rule_id	String	参数解释: 规则id。 取值范围: 不涉及。
rule_name	String	参数解释: 规则名称。 取值范围: 不涉及。
line_number	String	参数解释: 问题所在文件行号。 取值范围: 不涉及。
defect_content	String	参数解释: 问题详细描述。 取值范围: 不涉及。

参数	参数类型	描述
defect_level	String	参数解释: 问题级别。 取值范围: <ul style="list-style-type: none"> • 0, 致命。 • 1, 严重。 • 2, 一般。 • 3, 提示。
file_path	String	参数解释: 问题文件路径。 取值范围: 不涉及。
created_at	String	参数解释: 问题发现时间, 格式 2025-07-21T03:38:53Z。 取值范围: 不涉及。
issue_key	String	参数解释: MergeKey, 问题ID。 取值范围: 不涉及。
fragment	Array of DefectFragment V2 objects	参数解释: 代码片段内容。
events	Array of DefectEvents objects	参数解释: 调用链信息。

表 4-158 DefectFragmentV2

参数	参数类型	描述
line_num	String	参数解释: 行号。 取值范围: 不涉及。

参数	参数类型	描述
line_content	String	参数解释: 行代码内容。 取值范围: 不涉及。
start_offset	Integer	参数解释: 废弃字段。 取值范围: 不涉及。
end_offset	Integer	参数解释: 废弃字段。 取值范围: 不涉及。

表 4-159 DefectEvents

参数	参数类型	描述
events	Array of DefectEvents objects	参数解释: 调用链信息。
description	String	参数解释: 问题详细描述。 取值范围: 不涉及。
fix_suggestions	Array of strings	参数解释: 修复建议。 取值范围: 不涉及。
line	Integer	参数解释: 问题所在文件行号。 取值范围: 不涉及。
end_line	Integer	参数解释: 废弃字段。 取值范围: 不涉及。

参数	参数类型	描述
main	Boolean	参数解释: 废弃字段。 取值范围: 不涉及。
path	String	参数解释: 文件路径。 取值范围: 不涉及。
tag	String	参数解释: 废弃字段。 取值范围: 不涉及。
main_buggy_code	String	参数解释: 问题主事件对应的代码片段内容。 取值范围: 不涉及。
code_context	String	参数解释: 代码上下文。 取值范围: 不涉及。
code_context_start_line	Integer	参数解释: 代码上下文起始行号。 取值范围: 不涉及。

状态码：400

表 4-160 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-161 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

查询ID为2b31ed520xxxxxebedb6e57xxxxxxx的任务下的待处理问题详情

```
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxebedb6e57xxxxxxx/defects-detail?status_ids=0
```

```
null
```

响应示例

状态码: 200

Request succeeded!

```
{
  "defects": [ {
    "line_number": "81",
    "defect_id": "5b71a4594bcb4a69a33f576c3564e6cc",
    "rule_system_tags": "fossscan",
    "rule_id": "111",
    "rule_name": "FossScan.1 OpenSource Software",
    "defect_checker_name": "Disallow fallthrough of `case` statements",
    "defect_status": "0",
    "defect_content": "Update or refactor this function so that its implementation doesn't duplicate the one
on line 60.",
    "defect_level": "1",
    "file_path": "code/types/functions.ts",
    "created_at": "2020-07-21T02:12:01Z",
    "issue_key": "AA76E96XXXXXXXXXXXXXXXXXXCAAEE5",
    "fragment": [ {
      "line_num": "78",
      "line_content": "  export function padding(all: number);",
      "start_offset": -1,
      "end_offset": -1
    }, {
      "line_num": "79",
      "line_content": "  export function padding(topAndBottom: number, leftAndRight: number);",
      "start_offset": -1,
      "end_offset": -1
    }, {
      "line_num": "80",
      "line_content": "  export function padding(top: number, right: number, bottom: number, left:
number);",
      "start_offset": -1,
      "end_offset": -1
    }, {
      "line_num": "81",
      "line_content": "  export function padding(a: number, b?: number, c?: number, d?: number) {}",
      "start_offset": 11,
      "end_offset": 19
    }, {
      "line_num": "82",
      "line_content": "    if (b === undefined && c === undefined && d === undefined) {}",
      "start_offset": -1,
      "end_offset": -1
    }
  ]
}
```

```
}, {
  "line_number": "190",
  "defect_id": "5b71a4594wtrfsgrew546t466c3564e6cc",
  "rule_system_tags": "fosscan",
  "rule_id": "112",
  "rule_name": "FossScan.1 OpenSource Software",
  "defect_checker_name": "Disallow fallthrough of `case` statements",
  "defect_status": "0",
  "defect_content": "Update or refactor this function so that its implementation doesn't duplicate the one
on line 173.",
  "defect_level": "1",
  "file_path": "code/types/types.ts",
  "created_at": "2020-07-21T02:12:01Z",
  "issue_key": "AA76E96XXXXXXXXXXXXXXXXXXXXFE3B6",
  "fragment": [ {
    "line_num": "187",
    "line_content": "  var _value;",
    "start_offset": -1,
    "end_offset": -1
  }, {
    "line_num": "188",
    "line_content": "  function getOrSet(): number;",
    "start_offset": -1,
    "end_offset": -1
  }, {
    "line_num": "189",
    "line_content": "  function getOrSet(value: number);",
    "start_offset": -1,
    "end_offset": -1
  }, {
    "line_num": "190",
    "line_content": "  function getOrSet(value?: number) {\\"",
    "start_offset": 4,
    "end_offset": 12
  }, {
    "line_num": "191",
    "line_content": "    if (value === undefined) {\\"",
    "start_offset": -1,
    "end_offset": -1
  }
  ]
}, {
  "total": 2
}
```

状态码: 400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙, 请稍后再试"
}
```

状态码: 401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!

状态码	描述
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.2.4 查询缺陷详情的统计

功能介绍

根据检查任务ID查询问题详情的统计

调用方法

请参见[如何调用API](#)。

URI

GET /v2/tasks/{task_id}/defects-statistic

表 4-162 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	<p>参数解释： 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。</p> <p>约束限制： 不涉及。</p> <p>取值范围： 不涉及。</p> <p>默认取值： 不涉及。</p>

表 4-163 Query 参数

参数	是否必选	参数类型	描述
time_info_switch	否	String	<p>参数解释: 是否查询检查开始、结束时间。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • true, 查询检查开始、结束时间。 • false, 不查询检查开始、结束时间。 <p>默认取值: false</p>

请求参数

表 4-164 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

表 4-165 响应 Body 参数

参数	参数类型	描述
severity	StatisticSeverityV2 object	参数解释: 各问题级别的问题数统计。
status	StatisticStatusV2 object	参数解释: 各问题状态的问题数统计。
last_check_time	String	参数解释: 检查开始时间, 格式 2025-07-21T03:38:53Z。 取值范围: 不涉及。
last_check_finish_time	String	参数解释: 检查结束时间, 格式 2025-07-21T03:38:53Z。 取值范围: 不涉及。

表 4-166 StatisticSeverityV2

参数	参数类型	描述
critical	Integer	参数解释: 致命问题数。 取值范围: 不涉及。
major	Integer	参数解释: 严重问题数。 取值范围: 不涉及。
minor	Integer	参数解释: 一般问题数。 取值范围: 不涉及。
suggestion	Integer	参数解释: 提示问题数。 取值范围: 不涉及。

表 4-167 StatisticStatusV2

参数	参数类型	描述
unresolved	Integer	参数解释: 待处理问题数。 取值范围: 不涉及。
resolved	Integer	参数解释: 已解决+已修复问题数。 取值范围: 不涉及。
dismissed	Integer	参数解释: 已忽略问题数。 取值范围: 不涉及。
fixed	Integer	参数解释: 已修复问题数。 取值范围: 不涉及。
solved	Integer	参数解释: 已解决问题数。 取值范围: 不涉及。

状态码：400

表 4-168 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-169 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误描述

请求示例

查询ID为2b31ed520xxxxxebedb6e57xxxxxxx的任务问题详情的统计

```
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxebedb6e57xxxxxxx/defects-statistic
```

```
null
```

响应示例

状态码：200

Request succeeded!

```
{
  "severity": {
    "critical": 120,
    "major": 877,
    "minor": 79,
    "suggestion": 3
  },
  "status": {
    "unresolved": 877,
    "resolved": 79,
    "dismissed": 5,
    "fixed": 70,
    "solved": 9
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.2.5 修改缺陷状态

功能介绍

根据问题MergeKey，修改问题状态

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/tasks/{task_id}/defect-status

表 4-170 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	参数解释： 任务ID，即新建检查任务接口的返回值，每个CodeArtsCheck任务及其下的分支任务、通过API创建的增量任务都有1个唯一的任务ID。 约束限制： 不涉及。 取值范围： 不涉及。 默认取值： 不涉及。

请求参数

表 4-171 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

表 4-172 请求 Body 参数

参数	是否必选	参数类型	描述
defect_id	是	String	<p>参数解释: MergeKey，问题ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
defect_status	是	String	<p>参数解释: 问题状态。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • 0，待处理。 • 1，已解决。 • 2，已忽略。 <p>默认取值: 不涉及。</p>

响应参数

状态码：200

Request succeeded!

状态码：400

表 4-173 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-174 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

修改问题的状态：将ID为897acf5exxx48a5013cd的任务下MergeKey为363540xxxxxxxxx5105099944xxxxxxxx的问题状态修改为已解决

```
PUT https://{endpoint}/v2/tasks/897acf5exxx48a5013cd/defect-status
{
  "defect_id": "363540xxxxxxxxx5105099944xxxxxxxx",
  "defect_status": "1"
}
```

响应示例

状态码：200

Request succeeded!

null

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code" : "CC.00000003",
  "error_msg" : "认证信息过期"
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改问题的状态：将ID为897acf5exxx48a5013cd的任务下MergeKey为363540xxxxxxxx5105099944xxxxxxxx的问题状态修改为已解决

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;
import com.huaweicloud.sdk.codecheck.v2.*;
import com.huaweicloud.sdk.codecheck.v2.model.*;

public class UpdateDefectStatusSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CodeCheckClient client = CodeCheckClient.newBuilder()
            .withCredential(auth)
            .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateDefectStatusRequest request = new UpdateDefectStatusRequest();
        request.withTaskId("{task_id}");
        UpdateDefectRequestBody body = new UpdateDefectRequestBody();
        body.withDefectStatus("1");
        body.withDefectId("363540xxxxxxxx5105099944xxxxxxxx");
        request.withBody(body);
        try {
            UpdateDefectStatusResponse response = client.updateDefectStatus(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}
```

Python

修改问题的状态：将ID为897acf5exxx48a5013cd的任务下MergeKey为363540xxxxxxxx5105099944xxxxxxxx的问题状态修改为已解决

```
# coding: utf-8  
  
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkcodecheck.v2 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
  
    credentials = BasicCredentials(ak, sk)  
  
    client = CodeCheckClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = UpdateDefectStatusRequest()  
        request.task_id = "{task_id}"  
        request.body = UpdateDefectRequestBody(  
            defect_status="1",  
            defect_id="363540xxxxxxxx5105099944xxxxxxxx"  
        )  
        response = client.update_defect_status(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

修改问题的状态：将ID为897acf5exxx48a5013cd的任务下MergeKey为363540xxxxxxxx5105099944xxxxxxxx的问题状态修改为已解决

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
```

```

example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := codecheck.NewCodeCheckClient(
    codecheck.CodeCheckClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.UpdateDefectStatusRequest{
    request.TaskId = "{task_id}"
    defectStatusUpdateDefectRequestBody:= "1"
    defectIdUpdateDefectRequestBody:= "363540xxxxxxxx5105099944xxxxxxxx"
    request.Body = &model.UpdateDefectRequestBody{
        DefectStatus: &defectStatusUpdateDefectRequestBody,
        DefectId: &defectIdUpdateDefectRequestBody,
    }
}
response, err := client.UpdateDefectStatus(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
    
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.3 规则管理

4.3.1 获取规则列表接口

功能介绍

根据语言、问题级别等条件查询规则列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/rules

表 4-175 Query 参数

参数	是否必选	参数类型	描述
rule_languages	否	String	<p>参数解释: 规则的语言。</p> <p>约束限制: 需要传当前环境支持的语言。</p> <p>取值范围: 只能传一种语言。当前支持的语言有：ARKTS,C#,C++,CSS,GO,HTML,JAVA,JAVASCRIPT,KOTLIN,LUA,PHP,PYTHON,RUST,SCALA,SHELL,SQL,TYPESCRIPT。</p> <p>默认取值: 不涉及。</p>
rule_severity	否	String	<p>参数解释: 缺陷等级。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • 0，致命。 • 1，严重。 • 2，一般。 • 3，提示。 <p>默认取值: 0,1,2,3。默认全部查询</p>
offset	否	Integer	<p>参数解释: 分页索引，偏移量。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 0-999</p> <p>默认取值: 0</p>

参数	是否必选	参数类型	描述
limit	否	Integer	参数解释: 分页查询，每页显示的数量。 约束限制: 不涉及。 取值范围: 0-200 默认取值: 100

请求参数

表 4-176 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释: 用户Token。通过调用IAM服务 获取用户Token 接口获取（响应消息头中X-Subject-Token的值）。 约束限制: 不涉及。 取值范围: 字符串长度不少于1，不超过10万。 默认取值: 不涉及。

响应参数

状态码：200

表 4-177 响应 Body 参数

参数	参数类型	描述
info	Array of RuleListItem objects	参数解释: 规则信息。 取值范围: 不涉及。

参数	参数类型	描述
total	Integer	参数解释: 总数。 取值范围: 不涉及。

表 4-178 RuleListItem

参数	参数类型	描述
rule_set	String	参数解释: 规则的规范分类。 取值范围: 当前的分类有： common,common_standard_recommen d,common_standard_rule,design_sme ls,security,security_standard_recommen d,security_standard_rule。
rule_id	String	参数解释: 规则id。 取值范围: 32位uuid。
rule_language	String	参数解释: 规则所属语言。 取值范围: 当前支持的语言有：ARKTS,C#,C+ +,CSS,GO,HTML,JAVA,JAVASCRIPT,KOT LIN,LUA,PHP,PYTHON,RUST,SCALA,SHE LL,SQL,TYPESCRIPT。
rule_name	String	参数解释: 规则名称。 取值范围: 不涉及。
rule_severity	String	参数解释: 规则问题级别。 取值范围: <ul style="list-style-type: none"> • 0, 致命。 • 1, 严重。 • 2, 一般。 • 3, 提示。

参数	参数类型	描述
rule_tages	String	参数解释: 规则标签。 取值范围: huawei,iso5055,cert,cwe,top10,top25,misra,GJB,autosar,PCI DSS等。
right_example	String	参数解释: 正确示例。 取值范围: 代码块或者文字示例等。
error_example	String	参数解释: 错误示例。 取值范围: 代码块或者文字示例等。
revise_opinion	String	参数解释: 修改建议。 取值范围: 代码块或者文字示例等。
rule_desc	String	参数解释: 规则描述。 取值范围: 不涉及。

状态码：400

表 4-179 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-180 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误描述

请求示例

```
GET https://{endpoint}/v2/rules
```

响应示例

状态码：200

Request succeeded!

```
{
  "info": [ {
    "rule_id": "2561",
    "rule_language": "Java",
    "rule_name": "\"=\"+\"不能用\"+=\"代替",
    "rule_severity": "1",
    "rule_tags": "bug",
    "rule_desc": "<p>使用恒等运算符(<code>===</code>)比较不同的类型，其结果总是 <code>>false</code> 。</p>",
    "right_example": "<pre>\nvar a = 8;\nvar b = \"8\";\n\nif (a == b) {\n // ...\n}\n</pre>\n<p>or</p>\n<pre>\nvar a = 8;\nvar b = \"8\";\n\nif (a === Number(b)) {\n // ...\n}\n</pre>",
    "error_example": "<pre>\nvar a = 8;\nvar b = \"8\";\n\nif (a === b) { //不规范；比较结果总是false\n // ...\n}\n</pre>",
    "revise_opinion": "无",
    "rule_set": "common"
  }, {
    "rule_id": "8048",
    "rule_language": "Java",
    "rule_name": "\"Arrays.stream\" should be used for primitive arrays",
    "rule_severity": "1",
    "rule_tags": "performance",
    "rule_desc": "<p>使用恒等运算符(<code>===</code>)比较不同的类型，其结果总是 <code>>false</code> 。</p>",
    "right_example": "<pre>\nvar a = 8;\nvar b = \"8\";\n\nif (a == b) {\n // ...\n}\n</pre>\n<p>or</p>\n<pre>\nvar a = 8;\nvar b = \"8\";\n\nif (a === Number(b)) {\n // ...\n}\n</pre>",
    "error_example": "<pre>\nvar a = 8;\nvar b = \"8\";\n\nif (a === b) { //不规范；比较结果总是false\n // ...\n}\n</pre>",
    "revise_opinion": "无",
    "rule_set": "common"
  } ],
  "total": 2
}
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.3.2 创建自定义规则集

功能介绍

新建自定义规则集，支持在创建的时候启用规则

调用方法

请参见[如何调用API](#)。

URI

POST /v2/ruleset

请求参数

表 4-181 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释： 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制： 不涉及。</p> <p>取值范围： 字符串长度不少于1，不超过10万。</p> <p>默认取值： 不涉及。</p>

表 4-182 请求 Body 参数

参数	是否必选	参数类型	描述
project_id	是	String	<p>参数解释: 项目的32位uuid, 通过查询项目列表接口查询项目列表获取。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位项目id。</p> <p>默认取值: 不涉及。</p>
template_name	是	String	<p>参数解释: 新规则集名称。</p> <p>约束限制: 不能含有任意中英文, 数字, 点, 连字符-及下划线_以外的特殊字符。</p> <p>取值范围: 任意中英文, 数字, 点, 连字符-及下划线_。</p> <p>默认取值: 不涉及。</p>
language	是	String	<p>参数解释: 规则集语言。</p> <p>约束限制: 只能选择当前系统上支持的语言。</p> <p>取值范围: 只能传一种语言。当前支持的语言有: ARKTS,C#,C++,CSS,GO,HTML,JAVA,JAVASCRIPT,KOTLIN,LUA,PHP,PYTHON,RUST,SCALA,SHELL,SQL,TYPESCRIPT。</p> <p>默认取值: 不涉及。</p>

参数	是否必选	参数类型	描述
is_default	是	String	<p>参数解释: 是否基于已存在的规则集创建新规则集。</p> <p>约束限制: 只能填0或1。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • 0, 新建空白规则集。 • 1, 基于已存在的规则集创建新规则集。 <p>默认取值: 不涉及。</p>
rule_ids	是	String	<p>参数解释: 新启用规则ids,启用多条规则时, id用英文逗号相连。可调用获取规则列表接口获取规则id。</p> <p>约束限制: 规则id为32位uuid, 多个id之间用英文逗号分隔。例如: "00034fc32ee045388365d42ca4e07fe1,002c2eda6d6c11edab16fa163e0fa374"。</p> <p>取值范围: 字符串长度不超过4000。</p> <p>默认取值: 不涉及。</p>
uncheck_ids	否	String	<p>参数解释: 新关闭规则id, id用英文逗号相连。可调用获取规则列表接口获取规则id。</p> <p>约束限制: 规则id为32位uuid, 多个id之间用英文逗号分隔。例如: "00034fc32ee045388365d42ca4e07fe1,002c2eda6d6c11edab16fa163e0fa374"。</p> <p>取值范围: 字符串长度不超过4000。</p> <p>默认取值: 不涉及。</p>

参数	是否必选	参数类型	描述
template_id	否	String	<p>参数解释: 规则集ID。当is_default参数为1时, 该参数需要传参。可调用查询规则集列表获取已存在的规则集id。</p> <p>约束限制: 32位uuid。例如: "da303d80d2594faca60dbcc2e29c18aa"。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
custom_attributes	否	Array of CustomAttributes objects	自定义规则参数项, 支持修改规则阈值。

表 4-183 CustomAttributes

参数	是否必选	参数类型	描述
attribute	否	String	<p>参数解释: 配置项属性, severity: 为规则告警等级。</p> <p>约束限制: 不涉及。</p> <p>取值范围: severity。</p> <p>默认取值: 不涉及。</p>
rules	否	Array of CustomAttributesRule objects	规则详细。

表 4-184 CustomAttributesRule

参数	是否必选	参数类型	描述
rule_id	否	String	<p>参数解释: 规则ID。可调用获取规则列表接口获取。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位项目id。</p> <p>默认取值: 不涉及。</p>
value	否	String	<p>参数解释: attribute的告警等级，0致命，1严重，2一般，3提示。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • 0，致命。 • 1，严重。 • 2，一般。 • 3，提示。 <p>默认取值: 不涉及。</p>
rule_config_list	否	Array of RuleConfig objects	规则阈值详细。

表 4-185 RuleConfig

参数	是否必选	参数类型	描述
id	否	Integer	<p>参数解释: 规则配置ID。可调用获取规则列表接口获取规则配置ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位uuid。</p> <p>默认取值: 不涉及。</p>

参数	是否必选	参数类型	描述
rule_id	否	String	<p>参数解释: 工具规则ID, 即可配置阈值规则的各种阈值对应的ID, 当前无便利手段获取, 请联系客服咨询。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位uuid。</p> <p>默认取值: 不涉及。</p>
default_value	否	String	<p>参数解释: 配置参数默认阈值。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
option_value	否	String	<p>参数解释: 配置参数新阈值。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
option_key	否	String	<p>参数解释: 当前规则配置项key。可调用获取规则列表接口获取规则配置ID。</p> <p>约束限制: 需要和规则id相匹配。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

参数	是否必选	参数类型	描述
option_name	否	String	<p>参数解释: 当前规则配置项key。可调用获取规则列表接口获取规则配置ID。</p> <p>约束限制: 需要和规则id相匹配。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>
template_id	否	String	<p>参数解释: 规则集id。可调用查询规则集列表获取规则集id。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位uuid。</p> <p>默认取值: 不涉及。</p>
description	否	String	<p>参数解释: 配置参数的描述。可不填。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：201

表 4-186 响应 Body 参数

参数	参数类型	描述
template_id	String	<p>参数解释: 规则集id。</p> <p>取值范围: 32位uuid。</p>

状态码：400

表 4-187 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-188 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

创建自定义规则集：规则集对应的项目ID、模板ID根据实际显示，模板名称为 test_cpp，语言类型为cpp

```
POST https://{endpoint}/v2/ruleset
{
  "project_id": "24b97xxxxxxxxfb912625b14cxxxxx",
  "template_id": "da303dxxxxxxxxaca60dbcc2e2xxxxx",
  "template_name": "test_cpp",
  "language": "cpp",
  "is_default": "1",
  "rule_ids": "8139,8138",
  "uncheck_ids": "1101,1102"
}
```

响应示例

状态码：201

Request succeeded!

```
{
  "template_id": "c53417a4804f45eba9c11991131c9e79"
}
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
}
```

```
"error_msg": "网络繁忙, 请稍后再试"  
}
```

状态码: 401

Unauthorized

```
{  
  "error_code": "CC.00000003",  
  "error_msg": "认证信息过期"  
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建自定义规则集：规则集对应的项目ID、模板ID根据实际显示，模板名称为test_cpp，语言类型为cpp

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;  
import com.huaweicloud.sdk.codecheck.v2.*;  
import com.huaweicloud.sdk.codecheck.v2.model.*;  
  
public class CreateRulesetSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new BasicCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        CodeCheckClient client = CodeCheckClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))  
            .build();  
        CreateRulesetRequest request = new CreateRulesetRequest();  
        Ruleset body = new Ruleset();  
        body.withTemplatId("da303dxxxxxxxxaca60dbcc2e2xxxxx");  
        body.withUncheckIds("1101,1102");  
        body.withRuleIds("8139,8138");  
        body.withIsDefault("1");  
        body.withLanguage("cpp");  
        body.withTemplateName("test_cpp");  
        body.withProjectId("24b97xxxxxxxxfb912625b14xxxxx");  
        request.withBody(body);  
        try {  
            CreateRulesetResponse response = client.createRuleset(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

创建自定义规则集：规则集对应的项目ID、模板ID根据实际显示，模板名称为test_cpp，语言类型为cpp

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcodecheck.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]

    credentials = BasicCredentials(ak, sk)

    client = CodeCheckClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateRulesetRequest()
        request.body = Ruleset(
            template_id="da303dxxxxxxxxxaca60dbcc2e2xxxxxx",
            uncheck_ids="1101,1102",
            rule_ids="8139,8138",
            is_default="1",
            language="cpp",
            template_name="test_cpp",
            project_id="24b97exxxxxxxxfb912625b14cxxxxxx"
        )
        response = client.create_ruleset(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

创建自定义规则集：规则集对应的项目ID、模板ID根据实际显示，模板名称为test_cpp，语言类型为cpp

```
package main
```

```
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := codecheck.NewCodeCheckClient(  
        codecheck.CodeCheckClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.CreateRulesetRequest{}  
    templateIdRuleset:= "da303dxxxxxxxxaca60dbcc2e2xxxxxx"  
    uncheckIdsRuleset:= "1101,1102"  
    request.Body = &model.Ruleset{  
        TemplateId: &templateIdRuleset,  
        UncheckIds: &uncheckIdsRuleset,  
        RuleIds: "8139,8138",  
        IsDefault: "1",  
        Language: "cpp",  
        TemplateName: "test_cpp",  
        ProjectId: "24b97exxxxxxxxxfb912625b14cxxxxxx",  
    }  
    response, err := client.CreateRuleset(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
201	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.3.3 查询规则集列表

功能介绍

根据项目ID、语言等条件查询规则集列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/rulesets

表 4-189 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	参数解释: 项目的32位uuid, 通过 查询项目列表 接口查询项目列表获取。 约束限制: 不涉及。 取值范围: 32位项目id。 默认取值: 不涉及。

表 4-190 Query 参数

参数	是否必选	参数类型	描述
category	否	String	<p>参数解释: 规则集类别。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • 0, 只查询系统规则集。 • 1, 只查询当前用户自定义规则集。 • 2, 只查询其他用户自定义规则集。 <p>可以组合, 用英文逗号分隔。</p> <p>默认取值: 0,1,2。默认查询所有类别的规则集</p>
offset	否	Integer	<p>参数解释: 分页索引, 偏移量。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 0-999</p>
limit	否	Integer	<p>参数解释: 分页查询, 每页显示的数量。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 0-100</p>
need_selected_status	否	String	<p>参数解释: 是否查询规则集使用状态。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • true, 查询规则集使用状态。 • false, 不查询规则集使用状态, 规则集使用状态全部返回为未使用状态。

请求参数

表 4-191 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

表 4-192 响应 Body 参数

参数	参数类型	描述
info	Array of RulesetItem objects	<p>参数解释: 规则集列表信息。</p> <p>取值范围: 不涉及。</p>
total	Integer	<p>参数解释: 规则集总数。</p> <p>取值范围: 大于等于0。</p>

表 4-193 RulesetItem

参数	参数类型	描述
template_id	String	参数解释: 规则集id。 取值范围: 32位uuid。
language	String	参数解释: 规则集语言。 取值范围: 当前支持的语言有：ARKTS,C#,C++,CSS,GO,HTML,JAVA,JAVASCRIPT,KOTLIN,LUA,PHP,PYTHON,RUST,SCALA,SHELL,SQL,TYPESCRIPT。
template_name	String	参数解释: 规则集语言。 取值范围: 任意中英文，数字，点，连字符-及下划线_组成的字符串。
creator_id	String	参数解释: 创建者的user_id，值来源于iam的用户id。 取值范围: 32位uuid。
creator_name	String	参数解释: 创建人名称，值来源于IAM用户名称。 取值范围: IAM用户名称。
template_create_time	String	参数解释: 创建时间，时区统一为0时区。 取值范围: 0时区时间。
is_used	String	参数解释: 使用状态。 取值范围: <ul style="list-style-type: none"> 0，空闲中。 1，使用中。

参数	参数类型	描述
rule_ids	String	参数解释: 规则集包含的规则id，每条规则都有1个唯一的规则id。 取值范围: 32位uuid列表，多个uuid以英文逗号分隔。
is_default	String	参数解释: 规则集是否是该语言默认规则集。 取值范围: <ul style="list-style-type: none"> 0，不是。 1，是。
is_devcloud_project_default	String	参数解释: 规则集是否是项目下该语言默认规则集。 取值范围: <ul style="list-style-type: none"> 0，不是。 1，是。
is_default_template	String	参数解释: 规则集是否是系统规则集。 取值范围: <ul style="list-style-type: none"> 0，不是。 1，是。

状态码：400

表 4-194 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-195 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误描述

请求示例

GET https://{endpoint}/v2/{project_id}/rulesets

响应示例

状态码：200

Request succeeded!

```
{
  "info": [ {
    "template_id": "9698e0cb9d2441c3bd8cec5f8641696e",
    "language": "Java",
    "template_name": "test1111",
    "creator_id": "ae161856f2604229ae12056478a7919f",
    "creator_name": "杨老板",
    "template_create_time": "2021-10-21T02:14:55Z",
    "is_used": "0",
    "rule_ids": "",
    "is_default": "0",
    "is_default_template": "1",
    "is_devcloud_project_default": "0"
  }, {
    "template_id": "9698e0cb9d2441c3bd8cec5f8641696e",
    "language": "cpp",
    "template_name": "test222",
    "creator_id": "ae161856f2604229ae12056478a7919f",
    "creator_name": "杨老板",
    "template_create_time": "2021-10-21T02:14:55Z",
    "is_used": "1",
    "rule_ids": "11152,11153,11154,11155",
    "is_default": "0",
    "is_default_template": "1",
    "is_devcloud_project_default": "1"
  } ],
  "total": 2
}
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.3.4 查看规则集的规则列表

功能介绍

根据项目ID、规则集ID等条件查询规则列表。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/ruleset/{ruleset_id}/rules

表 4-196 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	<p>参数解释: 项目的32位uuid, 通过查询项目列表接口查询项目列表获取。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位项目id。</p> <p>默认取值: 不涉及。</p>

参数	是否必选	参数类型	描述
ruleset_id	是	String	<p>参数解释: 规则集ID, 通过查询规则集列表接口可获取到规则集ID。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位规则集ID。</p> <p>默认取值: 不涉及。</p>

表 4-197 Query 参数

参数	是否必选	参数类型	描述
types	否	String	<p>参数解释: 规则状态。</p> <p>约束限制: 不涉及。</p> <p>取值范围:</p> <ul style="list-style-type: none"> • 1, 查询全部规则。 • 2, 查询已启用规则。 • 3, 查询未启用。 <p>默认取值: 1</p>
languages	否	String	<p>参数解释: 规则语言。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 不涉及。</p> <p>默认取值: 不涉及。</p>

参数	是否必选	参数类型	描述
tags	否	String	参数解释: 规则所属标签。 约束限制: 不涉及。 取值范围: 不涉及。 默认取值: 不涉及。
offset	否	Integer	参数解释: 分页索引, 偏移量。 约束限制: 不涉及。 取值范围: 0-999
limit	否	Integer	参数解释: 分页查询, 每页显示的数量。 约束限制: 不涉及。 取值范围: 0-200

请求参数

表 4-198 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	参数解释: 用户Token。通过调用IAM服务 获取用户Token 接口获取(响应消息头中X-Subject-Token的值)。 约束限制: 不涉及。 取值范围: 字符串长度不少于1, 不超过10万。 默认取值: 不涉及。

响应参数

状态码：200

表 4-199 响应 Body 参数

参数	参数类型	描述
info	Array of RuleItem objects	参数解释: 规则集的规则列表信息。 取值范围: 不涉及。
total	Integer	参数解释: 总数。 取值范围: 不涉及。

表 4-200 RuleItem

参数	参数类型	描述
rule_id	String	参数解释: 规则id。 取值范围: 32位uuid。
rule_language	String	参数解释: 规则所属语言。 取值范围: 当前支持的语言有：ARKTS,C#,C++ +,CSS,GO,HTML,JAVA,JAVASCRIPT,KOT LIN,LUA,PHP,PYTHON,RUST,SCALA,SHE LL,SQL,TYPESCRIPT。
rule_name	String	参数解释: 规则名称。 取值范围: 不涉及。

参数	参数类型	描述
rule_severity	String	参数解释: 规则问题级别。 取值范围: <ul style="list-style-type: none"> • 0, 致命。 • 1, 严重。 • 2, 一般。 • 3, 提示。
tags	String	参数解释: 规则标签。 取值范围: huawei,iso5055,cert,cwe,top10,top25,misra,GJB,autosar,PCI DSS等。
checked	String	参数解释: 规则状态。 取值范围: <ul style="list-style-type: none"> • 0, 未启用。 • 1, 已启用。
rule_config_list	Array of RuleConfigParam objects	规则配置参数阈值相关信息。

表 4-201 RuleConfigParam

参数	参数类型	描述
id	Integer	参数解释: 规则配置ID。可调用 获取规则列表接口 获取规则配置ID。 取值范围: 32位uuid。
rule_id	String	参数解释: 工具规则ID，即可配置阈值规则的各种阈值对应的ID，当前无便利手段获取，请联系客服咨询。 取值范围: 32位uuid。

参数	参数类型	描述
default_value	String	参数解释: 配置参数默认阈值。 取值范围: 不涉及。
option_value	String	参数解释: 配置参数新阈值。 取值范围: 不涉及。
option_key	String	参数解释: 当前规则配置项key。可调用 获取规则列表接口 获取规则配置ID。 取值范围: 不涉及。
option_name	String	参数解释: 当前规则配置项key。可调用 获取规则列表接口 获取规则配置ID。 取值范围: 不涉及。
template_id	String	参数解释: 规则集id。可调用 查询规则集列表 获取规则集id。 取值范围: 32位uuid。
description	String	参数解释: 配置参数的描述。可不填。 取值范围: 不涉及。

状态码：400

表 4-202 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-203 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

GET https://{endpoint}/v2/{project_id}/ruleset/{ruleset_id}/rules

响应示例

状态码：200

Request succeeded!

```
{
  "info": [ {
    "rule_language": "cpp",
    "rule_id": "1614",
    "checked": "1",
    "rule_name": "A buffer must have size of 2 integers if used as parameter of pipe().",
    "rule_severity": "1",
    "tags": "cwe",
    "rule_config_list": [ ]
  }, {
    "rule_language": "cpp",
    "rule_id": "1611",
    "checked": "1",
    "rule_name": "A buffer must have size of 2 integers if used as parameter of pipe().",
    "rule_severity": "1",
    "tags": "cwe",
    "rule_config_list": [ {
      "id": 250,
      "rule_id": 11707,
      "template_id": "906e7eac47dd4bde9c984f5e6f2a54e8",
      "option_key": "threshold",
      "option_name": "新增阈值",
      "option_value": "10",
      "default_value": "20",
      "description": "代码复杂度超过阈值，则会被识别为超大圈复杂度，与圈复杂度统计区别在于统计SWITCH的时候，只统计SWITCH的个数，不统计case,default，必填"
    }, {
      "id": 251,
      "rule_id": 11707,
      "template_id": "906e7eac47dd4bde9c984f5e6f2a54e8",
      "option_key": "threshold_modify",
      "option_name": "修改阈值",
      "option_value": "30",
      "default_value": "20",
      "description": "代码复杂度超过阈值，则会被识别为超大圈复杂度。与圈复杂度统计区别在于统计SWITCH的时候，只统计SWITCH的个数，不统计case,default，修改阈值要比新增阈值大，非必填"
    }
  ]
  }, {
    "total": 2
  }
}
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
```

```
{
  "error_msg": "网络繁忙，请稍后再试"
}
```

状态码: 401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.3.5 删除自定义规则集

功能介绍

删除自定义规则集，正在使用中的或默认规则集不能删除

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/ruleset/{ruleset_id}

表 4-204 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	<p>参数解释: 项目的32位uuid，项目唯一标识，通过查询项目列表接口查询项目列表获取。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 32位项目ID。</p> <p>默认取值: 不涉及。</p>
ruleset_id	是	String	<p>参数解释: 规则集ID，通过查询规则集列表接口可获取到规则集ID</p> <p>约束限制: 不涉及</p> <p>取值范围: 32位规则集ID</p>

请求参数

表 4-205 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	<p>参数解释: 用户Token。通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。</p> <p>约束限制: 不涉及。</p> <p>取值范围: 字符串长度不少于1，不超过10万。</p> <p>默认取值: 不涉及。</p>

响应参数

状态码：200

Request succeeded!

状态码：400

表 4-206 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-207 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

```
DELETE https://{endpoint}/v2/{project_id}/ruleset/{ruleset_id}
```

响应示例

状态码：200

Request succeeded!

```
null
```

状态码：400

Bad Request

```
{
  "error_code": "CC.00000000",
  "error_msg": "网络繁忙, 请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code": "CC.00000003",
  "error_msg": "认证信息过期"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;
import com.huaweicloud.sdk.codecheck.v2.*;
import com.huaweicloud.sdk.codecheck.v2.model.*;

public class DeleteRulesetSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        CodeCheckClient client = CodeCheckClient.newBuilder()
            .withCredential(auth)
            .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteRulesetRequest request = new DeleteRulesetRequest();
        request.withProjectId("{project_id}");
        request.withRulesetId("{ruleset_id}");
        try {
            DeleteRulesetResponse response = client.deleteRuleset(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcodecheck.v2 import *

if __name__ == "__main__":
```

```
# The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = os.environ["CLOUD_SDK_AK"]
sk = os.environ["CLOUD_SDK_SK"]

credentials = BasicCredentials(ak, sk)

client = CodeCheckClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteRulesetRequest()
    request.project_id = "{project_id}"
    request.ruleset_id = "{ruleset_id}"
    response = client.delete_ruleset(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := codecheck.NewCodeCheckClient(
        codecheck.CodeCheckClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteRulesetRequest{}
    request.ProjectId = "{project_id}"
    request.RulesetId = "{ruleset_id}"
    response, err := client.DeleteRuleset(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

4.3.6 设置每个项目对应语言的默认规则集配置

功能介绍

设置每个项目对应语言的默认规则集配置。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/ruleset/{ruleset_id}/{language}/default

表 4-208 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	<p>参数解释: 项目的32位uuid，通过查询项目列表接口查询项目列表获取</p> <p>约束限制: 不涉及</p> <p>取值范围: 32位项目id</p>

参数	是否必选	参数类型	描述
ruleset_id	是	String	参数解释: 规则集ID, 通过 查询规则集列表 接口可获取到规则集ID 约束限制: 不涉及 取值范围: 32位规则集ID
language	是	String	参数解释: 规则语言 约束限制: 需要和规则集ID对应的规则集语言属性相同 取值范围: 不涉及

请求参数

表 4-209 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码：200

Request succeeded!

状态码：400

表 4-210 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

状态码：401

表 4-211 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误描述

请求示例

```
POST https://{endpoint}/v2/{project_id}/{language}/ruleset/{ruleset_id}/default
```

响应示例

状态码：200

Request succeeded!

```
null
```

状态码：400

Bad Request

```
{
  "error_code" : "CC.00000000",
  "error_msg" : "网络繁忙，请稍后再试"
}
```

状态码：401

Unauthorized

```
{
  "error_code" : "CC.00000003",
  "error_msg" : "认证信息过期"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.codecheck.v2.region.CodeCheckRegion;
import com.huaweicloud.sdk.codecheck.v2.*;
import com.huaweicloud.sdk.codecheck.v2.model.*;

public class SetDefaultTemplateSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

CodeCheckClient client = CodeCheckClient.newBuilder()
    .withCredential(auth)
    .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
    .build();
SetDefaultTemplateRequest request = new SetDefaultTemplateRequest();
request.withProjectId("{project_id}");
request.withRulesetId("{ruleset_id}");
request.withLanguage("{language}");
try {
    SetDefaultTemplateResponse response = client.setDefaultTemplate(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcodecheck.v2.region.codecheck_region import CodeCheckRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkcodecheck.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk)

    client = CodeCheckClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(CodeCheckRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = SetDefaultTemplateRequest()
        request.project_id = "{project_id}"
        request.ruleset_id = "{ruleset_id}"
        request.language = "{language}"
        response = client.set_default_template(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    codecheck "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/codecheck/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := codecheck.NewCodeCheckClient(
        codecheck.CodeCheckClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SetDefaultTemplateRequest{}
    request.ProjectId = "{project_id}"
    request.RulesetId = "{ruleset_id}"
    request.Language = "{language}"
    response, err := client.SetDefaultTemplate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	Request succeeded!
400	Bad Request
401	Unauthorized

错误码

请参见[错误码](#)。

5 场景化示例

5.1 示例 1：查询任务列表

场景描述

本章节指导用户根据DEVCLLOUD_PROJECT_UUID查询该项目下的任务列表。

约束限制

无。

涉及接口

涉及的接口如下：

查询任务列表：根据DEVCLLOUD_PROJECT_UUID查询任务列表。

操作步骤

查询任务列表。

- 接口相关信息

URI格式：GET /v2/{project_id}/tasks

- 请求示例

```
GET https://{endpoint}/v2/2b31ed520xxxxxxeb6e57xxxxxx/tasks
```

信息请从[终端节点](#)获取。

- 响应示例

```
{
  "result": [
    {
      "projectId": "b9db371361724226937f3280df1d4fc4",
      "devCloudProjectId": "a2b9a42ee6494e0794f5655f8535920e",
      "devCloudProjectName": "TestDemo",
      "codeHubName": "portal-ts",
      "codeHubWebUrl": "https://xxxxx/portal-ts",
      "codeLineTotal": "63371",
      "projectName": "portal-ts-abctest111222",
      "creatorId": "ae161856f2604229ae12056478a7919f",
    }
  ]
}
```

```

"creatorName": "name",
"domainId": "78d0e09c1ad0425b9b4f3a8fdd066164",
"domainName": "name",
"gitUrl": "git@xxxx/portal-ts.git",
"branch": "master",
"language": [
  "html"
],
"filePath": [
  ""
],
"lastCheckTime": "2019-09-24 15:00:41",
"codeQuality": "141.7",
"codeLine": "55777",
"projectStatus": "2",
"riskCoefficient": "7905",
"starLevel": "0",
"defectInfo": {
  "totalCount": "0",
  "criticalCount": "0",
  "seriousCount": "0",
  "normalCount": "0",
  "promptCount": "0"
},
"defectStatusInfo": {
  "totalCount": "2952",
  "newCount": "2952",
  "ignoredCount": "0",
  "fixedCount": "0"
},
"progress": {
},
"createdAt": "2019-08-08 17:56:40",
"duplications": "24.0%",
"complexityNum": "0",
"complexityAvg": "0",
"duplicatedLines": "15206",
"commentLines": "0",
"commentRatio": "0",
"duplicatedBlocks": "0",
"lastExecTime": "2 分 4 秒",
"checkType": "source",
"dependenceType": "",
"logDatasStr": {
}
},
{
  "projectId": "b9db371361724226937f3280df1d4fc4",
  "devCloudProjectId": "a2b9a42ee6494e0794f5655f8535920e",
  "devCloudProjectName": "TestDemo",
  "codeHubName": "portal-ts",
  "codeHubWebUrl": "https://xxxx/portal-ts",
  "codeLineTotal": "63371",
  "projectName": "portal-ts-abctest111222",
  "creatorId": "ae161856f2604229ae12056478a7919f",
  "creatorName": "devcloud_devcloud_y00336947_01",
  "domainId": "78d0e09c1ad0425b9b4f3a8fdd066164",
  "domainName": "name",
  "gitUrl": "git@xxxx/portal-ts.git",
  "branch": "master",
  "language": [
    "html"
  ],
  "filePath": [
    ""
  ],
  "lastCheckTime": "2019-09-24 15:00:41",

```

```
"codeQuality": "141.7",
"codeLine": "55777",
"projectStatus": "2",
"riskCoefficient": "7905",
"starLevel": "0",
"defectInfo": {
  "totalCount": "0",
  "criticalCount": "0",
  "seriousCount": "0",
  "normalCount": "0",
  "promptCount": "0"
},
"defectStatusInfo": {
  "totalCount": "2952",
  "newCount": "2952",
  "ignoredCount": "0",
  "fixedCount": "0"
},
"progress": {
},
"createdAt": "2019-08-08 17:56:40",
"duplications": "24.0%",
"complexityNum": "0",
"complexityAvg": "0",
"duplicatedLines": "15206",
"commentLines": "0",
"commentRatio": "0",
"duplicatedBlocks": "0",
"lastExecTime": "2 分 4 秒",
"checkType": "source",
"dependenceType": "",
"logDatasStr": {
}
},
"status": "success"
}
```

5.2 示例 2：查询缺陷概要

场景描述

本章节指导用户根据检查任务ID查询缺陷结果的概要，包括问题概述、问题状态、圈复杂度、代码重复率等。

约束限制

任务存在。

涉及接口

涉及的接口如下：

查询缺陷概要：查询缺陷概要，包括问题概述、问题状态、圈复杂度、代码重复率等。

操作步骤

- 接口相关信息
URI格式：GET /v2/tasks/{task_id}/defects-summary

- 请求示例
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxexbedb6e57xxxxxxx/defects-summary
信息请从[终端节点](#)获取。

- 响应示例

```
{
  "result":{
    "info":{
      "projectId":"b9db371361724226937f3280df1d4fc4",
      "devCloudProjectId":"a2b9a42ee6494e0794f5655f8535920e",
      "devCloudProjectName":"TestDemo",
      "codeHubName":"portal-ts",
      "codeHubWebUrl":"https://xxxxx/portal-ts",
      "codeLineTotal":"63371",
      "projectName":"portal-ts-abctest111222",
      "creatorId":"ae161856f2604229ae12056478a7919f",
      "creatorName":"name",
      "domainId":"78d0e09c1ad0425b9b4f3a8fdd066164",
      "domainName":"name",
      "gitUrl":"git@xxxxx/portal-ts.git",
      "branch":"master",
      "language":["
        \"html\"
      ],
      "filePath":["
        \"\"
      ],
      "lastCheckTime\":\"2019-09-24 15:00:41",
      "codeQuality":"141.7",
      "codeLine":"55777",
      "projectStatus":"2",
      "riskCoefficient":"790",
      "starLevel":"0",
      "defectInfo":{
        "totalCount":"0",
        "criticalCount":"0",
        "seriousCount":"0",
        "normalCount":"0",
        "promptCount":"0"
      },
      "defectStatusInfo":{
        "totalCount":"2952",
        "newCount":"2952",
        "ignoredCount":"0",
        "fixedCount":"0"
      },
      "progress":{
    },
    "createdAt":"2019-08-08 17:56:40",
    "duplications":"24.0%",
    "complexityNum":"0",
    "complexityAvg":"0",
    "duplicatedLines":"15206",
    "commentLines":"0",
    "commentRatio":"0",
    "duplicatedBlocks":"0",
    "lastExecTime":"2 分 4 秒",
    "checkType":"source",
    "dependenceType":"",
    "logDatasStr":{
      }
    }
  },
  "status":"success"
}
```

6 附录

6.1 状态码

状态码如表6-1所示。

表 6-1 状态码

状态码	编码	错误码说明
100	Continue	继续请求。 这个临时响应用来通知客户端，它的部分请求已经被服务器接收，且仍未被拒绝。
101	Switching Protocols	切换协议。只能切换到更高级的协议。 例如，切换到HTTP的新版本协议。
201	Created	创建类的请求完全成功。
202	Accepted	已经接受请求，但未处理完成。
203	Non-Authoritative Information	非授权信息，请求成功。
204	NoContent	请求完全成功，同时HTTP响应不包含响应体。在响应OPTIONS方法的HTTP请求时返回此状态码。
205	Reset Content	重置内容，服务器处理成功。
206	Partial Content	服务器成功处理了部分GET请求。
300	Multiple Choices	多种选择。请求的资源可包括多个位置，相应可返回一个资源特征与地址的列表用于用户终端（例如：浏览器）选择。
301	Moved Permanently	永久移动，请求的资源已被永久的移动到新的URI，返回信息会包括新的URI。
302	Found	资源被临时移动。

状态码	编码	错误码说明
303	See Other	查看其它地址。使用GET和POST请求查看。
304	Not Modified	所请求的资源未修改，服务器返回此状态码时，不会返回任何资源。
305	Use Proxy	所请求的资源必须通过代理访问。
306	Unused	已经被废弃的HTTP状态码。
400	BadRequest	非法请求。建议直接修改该请求，不要重试该请求。
401	Unauthorized	在客户端提供认证信息后，返回该状态码，表明服务端指出客户端所提供的认证信息不正确或非法。
402	Payment Required	保留请求。
403	Forbidden	请求被拒绝访问。 返回该状态码，表明请求能够到达服务端，且服务端能够理解用户请求，但是拒绝做更多的事情，因为该请求被设置为拒绝访问，建议直接修改该请求，不要重试该请求。
404	NotFound	所请求的资源不存在。建议直接修改该请求，不要重试该请求。
405	MethodNotAllowed	请求中带有该资源不支持的方法。建议直接修改该请求，不要重试该请求。
406	Not Acceptable	服务器无法根据客户端请求的内容特性完成请求。
407	Proxy Authentication Required	请求要求代理的身份认证，与401类似，但请求者应当使用代理进行授权。
408	Request Time-out	服务器等候请求时发生超时。客户端可以随时再次提交该请求而无需进行任何更改。
409	Conflict	服务器在完成请求时发生冲突。 返回该状态码，表明客户端尝试创建的资源已经存在，或者由于冲突请求的更新操作不能被完成。
410	Gone	客户端请求的资源已经不存在。返回该状态码，表明请求的资源已被永久删除。
411	Length Required	服务器无法处理客户端发送的不带Content-Length的请求信息。
412	Precondition Failed	未满足前提条件，服务器未满足请求者在请求中设置的其中一个前提条件。
413	Request Entity Too Large	由于请求的实体过大，服务器无法处理，因此拒绝请求。为防止客户端的连续请求，服务器可能会关闭连接。如果只是服务器暂时无法处理，则会包含一个Retry-After的响应信息。

状态码	编码	错误码说明
414	Request-URI Too Large	请求的URI过长（URI通常为网址），服务器无法处理。
415	Unsupported Media Type	服务器无法处理请求附带的媒体格式。
416	Requested range not satisfiable	客户端请求的范围无效。
417	Expectation Failed	服务器无法满足Expect的请求头信息。
422	Unprocessable Entity	请求格式正确，但是由于含有语义错误，无法响应。
429	TooManyRequests	表明请求超出了客户端访问频率的限制或者服务端接收到多于它能处理的请求。建议客户端读取相应的Retry-After首部，然后等待该首部指出的时间后再重试。
500	InternalServerError	表明服务端能被请求访问到，但是不能理解用户的请求。
501	Not Implemented	服务器不支持请求的功能，无法完成请求。
502	Bad Gateway	充当网关或代理的服务器，从远端服务器接收到了一个无效的请求。
503	ServiceUnavailable	被请求的服务无效。建议直接修改该请求，不要重试该请求。
504	ServerTimeout	请求在给定的时间内无法完成。客户端仅在为请求指定超时（Timeout）参数时会得到该响应。
505	HTTP Version not supported	服务器不支持请求的HTTP协议的版本，无法完成处理。

6.2 错误码

调用接口出错后，将不会返回结果数据。调用方可根据每个接口对应的错误码来定位错误原因。

当调用出错时，如HTTP请求返回一个400或505的HTTP状态码。返回的消息体中是具体的错误代码及错误信息。

在调用方找不到错误原因时，可以联系华为云客服，并提供错误码，以便我们尽快帮您解决问题。

错误响应 Body 体格式说明

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

```
{
  "error_msg": "The format of message is error",
```

```
"error_code": "AS.0001"
}
```

其中，“error_code”表示错误码，“error_msg”表示错误描述信息。

错误码说明

当您调用API时，如果遇到“APIGW”开头的错误码，请参见[API网关错误码](#)进行处理。

表 6-2 错误码

状态码	错误码	错误信息	描述	处理措施
200	CC.00050000	parameters not valid	参数非法或无效	检查参数格式
200	CC.00050001	query task list error	查询任务列表异常	请联系客服
200	CC.00050002	query task detail error	查询任务详情异常	请联系客服
200	CC.00050003	query templates error	查看任务配置模板异常	请联系客服
200	CC.00050004	query task history error	查询任务历史异常	请联系客服
200	CC.00050005	create task error	创建任务异常	请联系客服
200	CC.00050006	service inner error	服务内部错误	请联系客服
200	CC.00050007	IAM authentication failed	IAM校验失败	检查认证token是否有效
200	CC.00050010	current user does not have permission	权限不足	检查权限
200	CC.00050011	not in any project	当前用户不是租户任意项目的项目成员	加入项目
200	CC.00050012	retry later	网络繁忙	待会重试或者联系客服
200	CC.00050013	project not exist	项目不存在或项目已被删除	修改项目相关参数
200	CC.00050014	call getTaskBasicInfo failed	调用getTaskBasicInfo接口异常	待会重试或者联系客服

状态码	错误码	错误信息	描述	处理措施
200	CC.00050015	task is running	任务正在执行，稍后重试	稍后重试
200	CC.00050016	create task failed	新建任务出错！创建未成功！	检查相关参数或者联系客服
200	CC.00050017	call updateIssueStatus failed	请在最新扫描结果上进行忽略告警等处理	请在最新扫描结果上进行忽略告警等处理
200	CC.00050019	operations that are not supported by domain of the earlier version	老版本租户目前不支持的操作	老版本租户目前不支持的操作
200	CC.00050020	The current task has not been checked or has been checked successfully. Please check or try again after the check is successful	当前任务尚未检查或检查成功过，请检查或检查成功后重试	请检查或检查成功后重试

6.3 获取项目 ID

6.3.1 获取 IAM 项目 ID

调用 API 获取 IAM 项目 ID

项目ID还可通过调用[查询指定条件下的项目信息](#)API获取。

获取IAM项目ID的接口为“GET https://{Endpoint}/v3/projects/”，其中“{Endpoint}”为IAM的终端节点，可以从[终端节点](#)获取。接口的认证鉴权请参见[认证鉴权](#)。

响应示例如下，其中“projects”下的“id”即为IAM项目ID。

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",

      "name": "ap-southeast-3",

      "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"
}
}
```

从控制台获取 IAM 项目 ID

在调用接口的时候，部分URL中需要填入项目编号，所以需要获取到项目编号。项目编号获取步骤如下：

1. 登录管理控制台。
2. 鼠标悬停在右上角的用户名，在下拉列表中单击“我的凭证”。
进入“我的凭证 > API凭证”页面，在项目列表中查看项目ID。

6.3.2 获取 CodeArts 项目 ID

从 URL 获取 CodeArts 项目 ID

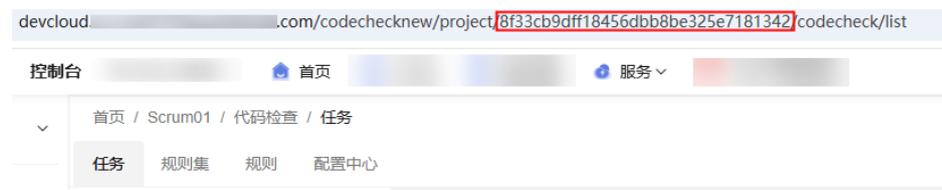
步骤1 使用华为云账号[登录华为云控制台页面](#)。

步骤2 单击页面左上角 ，在服务列表中选择“开发与运维 > 代码检查 CodeArts Check”。

步骤3 单击页面右上角“前往代码检查”，访问代码检查服务页面。

步骤4 进入指定项目的代码检查服务页面。如图6-1所示，红框部分即为CodeArts项目ID。

图 6-1 CodeArts 项目 ID



---结束

6.4 获取账号 ID

在调用接口的时候，部分URL中需要填入账号ID，所以需要先在管理控制台上获取到账号ID。账号ID获取步骤如下：

1. 登录管理控制台。
2. 鼠标悬停在右上角的用户名，在下拉列表中单击“我的凭证”。
进入“我的凭证 > API凭证”页面，即可查看账号ID。

6.5 Cmetrics 工具规格说明

工具介绍

Cmetrics是华为公司自研代码度量工具，通过扫描源码对各项指标进行度量。度量指标包括代码量、函数圈复杂度、文件代码行、函数代码行、文件重复、代码重复、危险函数、冗余代码等多项指标。当前工具版本为2.1.5。

参数说明

表 6-3 参数说明

参数	说明
code_size	代码规模。 统计工程中删除空行和注释行后的代码行数量。
raw_lines	原始代码行数。 统计工程中所有代码文件的物理行总数，包含空行和注释行。
methods_total	函数总数。 统计工程中所有代码文件的方法总个数。其中Java抽象方法的方法声明后不带函数体，不会被识别成函数。
cyclomatic_complexity_total	圈复杂度总数。 统计所有函数圈复杂度数的总和。 method默认圈复杂度为1，每出现特定关键字一次，则圈复杂度加1，出现三目运算符圈复杂度加1。 圈复杂度与cca圈复杂度的区别在于switch语句统计方式：圈复杂度计算时switch里面有几个case，圈复杂度就加几，而cca圈复杂度计算时，无论switch里面有几个case，圈复杂度只加1。
cyclomatic_complexity_per_method	平均圈复杂度。 平均圈复杂度=圈复杂度总数/函数总数
maximum_cyclomatic_complexity	最大圈复杂度。 代码检查任务执行完成后，统计所有函数中圈复杂度最大的值。
huge_cyclomatic_complexity_total	超大圈复杂度函数总数。 统计圈复杂度大于阈值的函数个数。其中阈值由Cmetrics工具版本决定，详情可参考表6-4；部分语言可通过Cmetrics规则修改。

参数	说明
huge_cyclomatic_complexity_ratio	<p>超大圈复杂度比例。</p> <p>超大圈复杂度比例=超大圈复杂度数/函数总数*100（单位：百分比%）</p>
cca_cyclomatic_complexity_total	<p>cca圈复杂度总数。</p> <p>统计所有函数cca圈复杂度数的总和。</p> <p>method默认圈复杂度为1，每出现特定关键字一次，则圈复杂度加1，出现三目运算符圈复杂度加1。</p> <p>圈复杂度与cca圈复杂度的区别在于switch语句统计方式：圈复杂度计算时switch里面有几个case，圈复杂度就加几，而cca圈复杂度计算时，无论switch里面有几个case，圈复杂度只加1。</p>
cca_cyclomatic_complexity_per_method	<p>cca平均圈复杂度。</p> <p>cca平均圈复杂度=cca圈复杂度总数/函数总数</p>
maximum_cca_cyclomatic_complexity	<p>cca最大圈复杂度。</p> <p>代码检查任务执行完成后，统计所有函数中cca圈复杂度最大的值。</p>
huge_cca_cyclomatic_complexity_total	<p>超大cca圈复杂度函数总数。</p> <p>统计cca圈复杂度大于阈值的函数个数。其中阈值由Cmetrics工具版本决定，部分语言可通过Cmetrics规则修改。</p>
cyclomatic_complexity_adequacy	<p>圈复杂度满足度。</p> <p>圈复杂度满足度=((函数总数-超大cca圈复杂度函数总数) / 函数总数)*100（单位：百分比%）</p> <p>cyclomatic complexity adequacy的缩写即为cca。</p>
maximum_depth	<p>最大深度。</p> <p>代码检查任务执行完成后，统计所有函数中深度最大的值。</p>
huge_depth_total	<p>超大深度数。</p> <p>统计最大深度大于阈值的函数个数。其中阈值由Cmetrics工具版本决定，详情可参考表6-4；部分语言可通过Cmetrics规则修改。</p>
huge_depth_ratio	<p>超大深度占比。</p> <p>超大深度占比=超大深度数/函数总数（单位：百分比%）</p>
method_lines	<p>函数总行数。</p> <p>统计所有函数的非空非注释行代码行总和。</p> <p>注意：在去掉空行注释行以后，统计函数内部代码块的行数，函数开头定义行不统计在内。</p>

参数	说明
lines_per_method	函数平均代码行。 函数平均代码行=函数总行数/函数总数
huge_method_total	超大函数数。 函数代码行大于阈值的函数个数。其中阈值由Cmetrics工具版本决定，详情可参考表6-4；部分语言可通过Cmetrics规则修改。
huge_method_ratio	超大函数占比。 超大函数占比=超大函数数/函数总数*100（单位：百分比%）
files_total	文件总数。 扫描工程目录中符合源码文件后缀名的文件总个数，其中空文件不会统计在内。不同语言对应的文件后缀可参考表6-5。
folders_total	目录总数。 工程里目录的总个数。其中空目录不会被计算在内。
lines_per_file	文件平均代码行。 文件平均代码行=代码行总数/文件总数
huge_headerfile_total	超大头文件数。 C语言源码文件中后缀名为“.h”，“.hh”，“.hpp”，“.hxx”，“.h++”，“.inc”，“.inl”的被认为是头文件，其中有效代码行大于阈值的认为是超大头文件。其中阈值由Cmetrics工具版本决定，详情可参考表6-4。
huge_headerfile_ratio	超大头文件占比。 超大头文件占比=超大头文件数/文件总数*100（单位：百分比%）
huge_non_headerfile_total	超大源文件数。 源码文件中后缀名不为头文件，并且有效代码行大于阈值的认为是超大源文件。其中阈值由Cmetrics工具版本决定，详情可参考表6-4；部分语言可通过Cmetrics规则修改。
huge_non_headerfile_ratio	超大源文件占比。 超大源文件占比=超大源文件数/文件总数*100（单位：百分比%）
huge_folder_total	超大目录数。 统计某目录内的文件和子目录总数（不计算该目录节点下子目录下更深层次的子目录数和文件数），当总数大于阈值时，则判断为超大目录。空子目录（子目录里面没有源码文件的）不计算在内。其中阈值由Cmetrics工具版本决定，详情可参考表6-4；部分语言可通过Cmetrics规则修改。
huge_folder_ratio	超大目录占比。 超大目录占比=超大目录数/目录总数*100（单位：百分比%）

参数	说明
file_duplication_total	重复文件数。 统计文件内容完全相同的文件个数。仅统计后缀名符合各语言源码文件后缀名的文件，不同语言对应的文件后缀可参考表6-5。
file_duplication_ratio	文件重复率。 文件重复率=重复文件数/文件总数*100（单位：百分比%）
non_hfile_duplication_total	重复源文件数。 统计C语言重复文件中后缀名为源文件的文件个数。其他语言没有头文件，所有文件都是源文件。
non_hfile_duplication_ratio	源文件重复率。 源文件重复率=重复源文件数/文件总数*100（单位：百分比%）
code_duplication_total	代码重复数。 非空非注释行连续10行相同的代码片段和import行（第三方机构标准），会被识别是重复代码。 说明 代码重复数是过每个文件的重复代码行相加得出，而不是以重复代码明细结果里的代码片段相加。如：三个文件都有相同的10行重复代码，则代码重复数为30；如：A文件1~10行与B文件重复，A文件2~11行与C文件重复，则取代码片段并集的行数，A文件重复代码行为11行。
code_duplication_ratio	代码重复率。 代码重复率=代码重复数/代码行总数*100（单位：百分比%）
non_hfile_code_duplication_total	源文件代码重复数。 源文件的代码重复行数总数。即只统计后缀名符合源代码文件（C/C++语言不包含头文件）要求的文件的重复代码行。
non_hfile_code_duplication_ratio	源文件代码重复率。 源文件代码重复率=源文件代码重复数/代码行总数（单位：百分比%）
unsafe_functions_total	危险函数总数。 统计代码中出现调用不安全函数，如memcpy、memmove等函数的调用次数。 说明 目前clike检查不安全函数，其他语言不检查该指标。
unsafe_functions_kloc	危险函数密度（每千行代码中包含的危险函数个数）。 危险函数密度=危险函数总数/代码行总数*1000

参数	说明
redundant_code_total	冗余代码数。 统计注释中冗余代码块的个数。注释中的代码被视作冗余代码；注释中包含编程语言关键字的或符合语法规则的连续单词组会被判断为冗余代码。 其中块注释有多行冗余代码，只算一次冗余代码数。
redundant_code_kloc	冗余代码块密度（每千行代码中包含的冗余代码数）。 冗余代码块密度=冗余代码数/代码行总数*1000 说明 如果代码行总数少于1000，不计算密度，取值为0。
warning_suppression_total	抑制告警数。 统计被扫描文件的告警抑制语句出现的数量。注释中的字符串符合特定正则表达式，则会被认为是告警抑制语句。不同语言对应的正则表达式请参考表6-7。 该参数需启用Cmetrics规则后生效。
warning_suppression_kloc	抑制告警密度。 抑制告警密度=抑制告警数/代码行总数*1000（每千行代码中包含的冗余代码数） 该参数需启用Cmetrics规则后生效。

表 6-4 Cmetrics 工具默认阈值

阈值名称	默认值
超大圈复杂度函数-圈复杂度阈值	20
超大深度数函数-深度阈值	4
超大函数-行数阈值	50
超大头文件-行数阈值	500
超大源文件-行数阈值	2000
超大目录数-节点阈值	50

表 6-5 Cmetrics 工具编程语言与文件扩展名关联关系

语言	后缀
JAVA	.java
C/C++	.c/.cc/.cpp/.cxx/.cp/.c+ +/.inc/.inl/.mm/.h/.hh/.hpp/.hxx/.h++
TYPESCRIPT	.ts/.tsx

语言	后缀
RUST	.rs
ARKTS	.ets
C#	.cs
GO	.go
JAVASCRIPT	.js/.jsx
LUA	.lua
PYTHON	.py

表 6-6 Cmetrics 工具支持检查 C/C++ 的危险函数列表

_fstrncat	_vsntprintf	strcatbuff	vos_nsprintf
_fstrncpy	_vsnwprintf	strcatbuffa	vos_nvsprintf
_ftscat	_vstprintf	strcatbuffw	vos_snprintf
_ftscopy	bcopy	strcatchainw	vos_sprintf
_getts	copymemory	strcatn	vos_sscanf
_gettws	fscanf	strcatna	vos_strcat
_getws	fwscanf	strcatnw	vos_strcpy
_mbccat	gets	strcatw	vos_strncat
_mbccpy	lstrcat	strcpy	vos_strncpy
_mbscat	lstrcata	strcpya	vos_vsprintf
_mbscopy	lstrcatn	strcpyn	vos_vsscanf
_mbsnbcata	lstrcatnA	strcpyna	vscanf
_mbsnbcopy	lstrcatnW	strcpynw	vsprintf
_mbsncat	lstrcatw	strcpyw	vsprintf
_mbsncopy	lstrcpy	strncat	vsscanf
_snprintf	lstrcpya	strncata	vswprintf
_sntprintf	lstrcpyn	strncatw	vswscanf
_sntscanf	lstrcpyna	strncpy	vwscanf
_snwprintf	lstrcpynw	strncpya	wcscat
_stprintf	lstrcpyw	strncpyw	wcscopy

_stscanf	lstrncat	swprintf	wcsncat
_tccat	memcpy	swscanf	wcsncpy
_tccpy	memmove	vfscanf	wmemcpy
_tcscat	memset	vfwscanf	wmemmove
_tcscopy	nsprintf	vos_chkmemcpy	wnsprintf
_tcsncat	rtlcopymemory	vos_chkmemmove	wnsprintfA
_tcsncpy	scanf	vos_chkmemset	wnsprintfW
_tscanf	snprintf	vos_chkstrcat	wscanf
_vos_bzero	snscanf	vos_chkstrcpy	wsprintf
_vos_memcpy	sntprintf	vos_chkstrncat	wsprintfA
_vos_memmove	snwscanf	vos_chkstrncpy	wsprintfw
_vos_memset	sprintf	vos_mem_copy	wvnsprintf
_vos_strcat	sprintfA	vos_mem_set	wvnsprintfA
_vos_strcpy	sprintfw	vos_mem_zero	wvnsprintfw
_vos_strncat	sscanf	vos_memcpy	wvsprintf
_vos_strncpy	strcat	vos_memmove	wvsprintfA
_vsnprintf	strcata	vos_memset	wvsprintfw

表 6-7 Cmetrics 工具告警抑制规则

语言类型	注释匹配正则表达式
C/C++、JAVA、JAVASCRIPT、TYPESCRIPT	<ul style="list-style-type: none"> • r'(/\s*/)\s*(lint\b\s+\S+ coverity\s*\[.+\?]) eslint-disable)' • r' ^\s*#\s*pragma\s+\w+\s+diagnostic\s+(ignored warning)' • r' ^\s*#\s*pragma\s+warning\s*\(\s*disable' • r' ^\s*@(\w+\.)?SuppressWarnings)'
PYTHON	r'#\s*pylint\s*:'

7 修订记录

文档修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订记录	发布日期
第二次正式发布。本次变更说明如下： 代码检查改成新的服务名称（CodeArts Check）。	2022-04-23
第一次正式发布。	2022-09-30