数据安全中心

接口参考

文档版本 16

发布日期 2025-07-09

华为云计算技术有限公司

商标声明


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云计算技术有限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，华为云计算技术有限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。

华为云计算技术有限公司

地址： 贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址： https://www.huaweicloud.com/

目 录

1 使用前必读

概述

数据安全中心服务（Data Security Center，DSC）是新一代的云原生数据安全管理平台，提供数据分级分类、数据脱敏、数据水印等基础数据安全能力，通过资产地图整体呈现云上数据安全态势，并实现一站式数据安全运营能力。

您可以使用本文档提供的API对数据对进行相关操作，如数据库水印、图片水印、数据脱敏等。

调用说明

数据安全中心服务提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用。

调用方法请参见**如何调用API**。

终端节点

终端节点（Endpoint）即调用API的**请求地址**，不同服务不同区域的终端节点不同，您可以从**地区和终端节点**中查询服务的终端节点。

约束与限制

- DSC支持的文档和图片水印兼容的范围如**表1-1**。

  表 1-1 文档/图片水印支持的类型

文档/图片水印载体	Office(Windows、Mac)	WPS(Windows、Mac、Linux、手机端)	Adobe Reader	浏览器(chrome、Edge)	福昕PDF
PDF	-	√	√	√	√
WORD	√	√	-	-	-
EXCEL	√	√	-	-	-

文档/图片水印载体	Office(Windows、Mac)	WPS(Windows、Mac、Linux、手机端)	Adobe Reader	浏览器(chrome、Edge)	福昕PDF
PPT	√	√	-	-	-

说明

“√”表示支持，“-”表示平台不支持。

- DSC的API接口暂不支持直接对OBS桶数据进行添加/提取水印的操作，如果您需要对OBS桶数据进行水印相关的操作，请先将OBS桶数据读取到本地，再调用水印的API接口进行操作，添加水印后的文档将放在API的响应体中进行返回。
- 更详细的限制请参见具体API的说明。

基本概念

- 账号

  用户注册时的账号，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其进行日常管理工作。

- 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）。

  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。

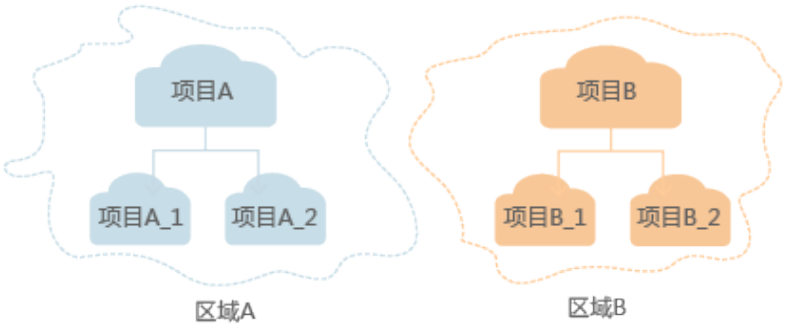
- 可用区（AZ，Availability Zone）

  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。

- 项目

  区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中创建资源，然后以子项目为单位进行授权，使得用户仅能访问特定子项目中资源，使得资源的权限控制更加精确。

图 1-1 项目隔离模型



2 如何调用 API

2.1 构造请求

本节介绍如何构造REST API的请求，并以调用IAM服务的**获取用户Token**说明如何调用API，该API获取用户的Token，Token可以用于调用其他API时鉴权。

请求 URI

请求URI由如下部分组成。

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

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。

- **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**获取。

  例如IAM服务在“中国-香港”区域的Endpoint为“iam.ap-southeast-1.myhuaweicloud.com”。

- **resource-path**：

  资源路径，也即API访问路径。从具体API的URI模块获取，例如“获取用户Token”API的resource-path为“/v3/auth/tokens”。

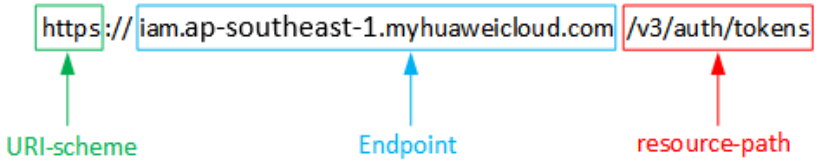
- **query-string**：

  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。

例如您需要获取IAM在“中国-香港”区域的Token，则需使用“中国-香港”区域的Endpoint（iam.ap-southeast-1.myhuaweicloud.com），并在**获取用户Token**的URI部分找到resource-path（/v3/auth/tokens），拼接起来如下所示。

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

图 2-1 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，同一个服务的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**获取用户Token**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-1.myhuaweicloud.com/v3/auth/tokens
```

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。
- **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。

  说明

  API同时支持使用AK/SK认证，AK/SK认证是使用SDK对请求进行签名，签名过程会自动往请求中添加Authorization（签名认证信息）和X-Sdk-Date（请求发送的时间）请求头。

  AK/SK认证的详细说明请参见**AK/SK认证**。

对于**获取用户Token**接口，由于不需要认证，所以只添加“Content-Type”即可，添加消息头后的请求如下所示。

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

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。若请求消息体中参数支持中文，则中文字符必须为UTF-8编码。

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**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，其中***username***为用户名，***domainname***为用户所属的账号名称，*********为用户登录密码，***xxxxxxxxxxxxxxxxx***为project的名称，您可以从**地区和终端节点**获取，对应地区和终端节点页面的“区域”字段的值。

说明

scope参数定义了Token的作用域，下面示例中获取的Token仅能访问project下的资源。您还可以设置Token作用域为某个账号下所有资源或账号的某个project下的资源，详细定义请参见**获取用户Token**。

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": "xxxxxxxxxxxxxxxxx"
            }
        }
    }
}
```

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。对于获取用户Token接口，返回的响应消息头中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

2.2 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认证调用请求。
- AK/SK认证：通过AK（Access Key ID）/SK（Secret Access Key）加密调用请求。推荐使用AK/SK认证，其安全性比Token认证要高。

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。

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**获取用户Token**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": "xxxxxxxx"
            }
        }
    }
}
```

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“ABCDEFG....”，则调用接口时将“X-Auth-Token: ABCDEFG....”加到请求消息头即可，如下所示。

```
GET https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFG....
```

AK/SK 认证

说明

AK/SK签名认证方式仅支持消息体大小12MB以内，12MB以上的请求请使用Token认证。

AK/SK认证就是使用AK/SK对请求进行签名，在请求时将签名信息添加到消息头，从而通过身份认证。

- 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。

使用AK/SK认证时，您可以基于签名算法使用AK/SK对请求进行签名，也可以使用专门的签名SDK对请求进行签名。详细的签名方法和SDK使用方法请参见**API签名指南**。

须知

签名SDK只提供签名功能，与服务提供的SDK不同，使用时请注意。

2.3 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**获取用户Token**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**获取用户Token**接口，返回如**图2-2**所示的消息头，其中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

图 2-2 获取用户 Token 响应消息头

connection → keep-alive

content-type → application/json

date → Tue, 12 Feb 2019 06:52:13 GMT

server → Web Server

strict-transport-security → max-age=31536000; includeSubdomains;

transfer-encoding → chunked

via → proxy A

x-content-type-options → nosniff

x-download-options → noopen

x-frame-options → SAMEORIGIN

x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5

x-subject-token
→ MIIYXQYJKoZIhvcNAQcCoIIYTjCCGEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOnsiZXhwaXJlc19hdCI6IjIwMTktMDItMTNUMD
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkqjACgklqO1wi4JIGzrpd18LGXK5txldfq4lqHCYb8P4NaY0NYejcAgzJVeFIYtLWT1GSO0zxKZmIQHQj82HBqHdgIZO9fuEbL5dMhdavj+33wEI
xHRCE9l87o+k9-
j+CMZSEB7bUGd5Uj6eRASXI1jipPEGA270g1FruooL6jqglFkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==

x-xss-protection → 1; mode=block;

响应消息体（可选）

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**获取用户Token**接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

```
{
    "token": {
        "expires_at": "2019-02-13T06:52:13.855000Z",
```



```
        "methods": [
            "password"
        ],
        "catalog": [
            {
                "endpoints": [
                    {
                        "region_id": "xxxxxxxx",
.....
```

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": {
        "message": "The request you have made requires authentication.",
        "title": "Unauthorized"
    }
}
```

其中，error_code表示错误码，error_msg表示错误描述信息。

3 API 说明

3.1 资源管理

3.1.1 实例下单

功能介绍

根据计费方式、计费周期等信息进行实例下单

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/period/order

表 3-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-2 请求 Body 参数

参数	是否必选	参数类型	描述
charging_mode	是	Integer	计费模式，0：包周期计费， 1：按需计费，2：一次性计费

参数	是否必选	参数类型	描述
cloud_service_type	是	String	云服务类型
composite_product_id	否	String	组合套餐ID
discount_id	否	String	折扣ID
is_auto_renew	否	Integer	是否自动续费
period_num	是	Integer	订购周期数目
period_type	是	Integer	订购周期类型，2：月，3：年
product_infos	是	Array of **ProductInfoBean** objects	产品信息列表
promotion_activity_id	否	String	促销ID
promotion_info	否	String	促销信息
region_id	是	String	当前项目所在region的id，如：xx-xx-1。
zone	是	String	所属国家区域

表 3-3 ProductInfoBean

参数	是否必选	参数类型	描述
all_resource_names	否	Array of strings	资源名称列表
cloud_service_type	是	String	云服务类型
display_id	否	String	展示ID
product_id	是	String	产品ID
product_spec_desc	否	String	产品规格描述
resource_name	否	String	资源名称
resource_size	是	Integer	产品支持的数据库数量，或者支持obs的扫描量

参数	是否必选	参数类型	描述
resource_size_measure_id	是	Integer	资源容量度量标识，枚举值举例如下：15：mbps（购买带宽时使用），17：gb（购买云硬盘时使用），14：个/次
resource_spec_code	是	String	产品编码
resource_type	是	String	资源类型
usage_factor	否	String	已使用系数
usage_measure_id	否	Integer	已使用容量度量标识
usage_value	否	Double	已使用值

响应参数

状态码：200

表 3-4 响应 Body 参数

参数	参数类型	描述
order_id	String	订单ID

状态码：400

表 3-5 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

在中国区域xxxxxxxxxxxx项目购买2个月的hws.service.type.sdg云服务实例，该产品支持XX个数据库或者obs的扫描量。

```
POST /v1/{project_id}/period/order

{
  "charging_mode" : 0,
  "cloud_service_type" : "hws.service.type.sdg",
  "is_auto_renew" : 0,
  "period_num" : 1,
  "period_type" : 2,
```

```
  "region_id" : "xxxxxxxxxxxx",
  "zone" : "CH",
  "product_infos" : [ {
    "cloud_service_type" : "hws.service.type.sdg",
    "product_id" : "xxxxxxxxxxxx",
    "resource_size" : 30,
    "resource_size_measure_id" : 30,
    "resource_spec_code" : "base_professional",
    "resource_type" : "hws.resource.type.dsc.base"
  } ]
}
```

响应示例

状态码：200

OK

```
{
  "order_id" : "xxxxxxxxxxxx"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

在中国区域xxxxxxxxxxxx项目购买2个月的hws.service.type.sdg云服务实例，该产品支持XX个数据库或者obs的扫描量。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;

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

public class CreateProductOrderSolution {

    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");
        String projectId = "{project_id}";
```

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateProductOrderRequest request = new CreateProductOrderRequest();
        PeriodOrderRequest body = new PeriodOrderRequest();
        List<ProductInfoBean> listbodyProductInfos = new ArrayList<>();
        listbodyProductInfos.add(
            new ProductInfoBean()
                .withCloudServiceType("hws.service.type.sdg")
                .withProductId("xxxxxxxxxxx")
                .withResourceSize(30)
                .withResourceSizeMeasureId(30)
                .withResourceSpecCode("base_professional")
                .withResourceType("hws.resource.type.dsc.base")
        );
        body.withZone("CH");
        body.withRegionId("xxxxxxxxxxx");
        body.withProductInfos(listbodyProductInfos);
        body.withPeriodType(2);
        body.withPeriodNum(1);
        body.withIsAutoRenew(0);
        body.withCloudServiceType("hws.service.type.sdg");
        body.withChargingMode(0);
        request.withBody(body);
        try {
            CreateProductOrderResponse response = client.createProductOrder(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

在中国区域xxxxxxxxxxxx项目购买2个月的hws.service.type.sdg云服务实例，该产品支持XX个数据库或者obs的扫描量。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"
```

```
credentials = BasicCredentials(ak, sk, projectId)

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

try:
    request = CreateProductOrderRequest()
    listProductInfosbody = [
        ProductInfoBean(
            cloud_service_type="hws.service.type.sdg",
            product_id="xxxxxxxxxxxx",
            resource_size=30,
            resource_size_measure_id=30,
            resource_spec_code="base_professional",
            resource_type="hws.resource.type.dsc.base"
        )
    ]
    request.body = PeriodOrderRequest(
        zone="CH",
        region_id="xxxxxxxxxxxx",
        product_infos=listProductInfosbody,
        period_type=2,
        period_num=1,
        is_auto_renew=0,
        cloud_service_type="hws.service.type.sdg",
        charging_mode=0
    )
    response = client.create_product_order(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

在中国区域xxxxxxxxxxxx项目购买2个月的hws.service.type.sdg云服务实例，该产品支持XX个数据库或者obs的扫描量。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

```
    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateProductOrderRequest{}
    var listProductInfosbody = []model.ProductInfoBean{
        {
            CloudServiceType: "hws.service.type.sdg",
            ProductId: "xxxxxxxxxxxx",
            ResourceSize: int32(30),
            ResourceSizeMeasureId: int32(30),
            ResourceSpecCode: "base_professional",
            ResourceType: "hws.resource.type.dsc.base",
        },
    }
    isAutoRenewPeriodOrderRequest:= int32(0)
    request.Body = &model.PeriodOrderRequest{
        Zone: "CH",
        RegionId: "xxxxxxxxxxxx",
        ProductInfos: listProductInfosbody,
        PeriodType: int32(2),
        PeriodNum: int32(1),
        IsAutoRenew: &isAutoRenewPeriodOrderRequest,
        CloudServiceType: "hws.service.type.sdg",
        ChargingMode: int32(0),
    }
    response, err := client.CreateProductOrder(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	OK
400	无效请求

错误码

请参见**错误码**。

3.1.2 查询资源开通信息

功能介绍

查询资源开通信息，根据项目ID查询订单详情

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/period/product/specification

表 3-6 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

无

响应参数

状态码：200

表 3-7 响应 Body 参数

参数	参数类型	描述
order_infos	Array of **ProductOrderInfo** objects	订单列表

表 3-8 ProductOrderInfo

参数	参数类型	描述
tenant_id	String	租户ID
period_type	String	订购周期类型
period_num	Integer	订购周期数量
resource_id	String	资源ID
product_info	**ProductInfo** object	产品信息

表 3-9 ProductInfo

参数	参数类型	描述
all_resourcenames	Array of strings	资源名称列表
cloud_service_type	String	云服务类型
display_id	String	展示ID
product_id	String	产品ID
product_spec_desc	String	产品规格描述
resource_name	String	资源名称
resource_size	Integer	产品支持的数据库数量，或者支持obs的扫描量
resource_size_measure_id	Integer	资源容量度量标识，枚举值举例如下：15：mbps（购买带宽时使用），17：gb（购买云硬盘时使用），14：个/次
resource_spec_code	String	产品编码
resource_type	String	资源类型
usage_factor	String	已使用系数
usage_measure_id	Integer	已使用容量度量标识
usage_value	Double	已使用值

状态码：400

表 3-10 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询资源开通信息

```
GET /v1/{project_id}/period/product/specification
```

响应示例

状态码：200

请求成功

```
{
  "order_infos" : [ {
    "tenant_id" : "xxxxxxxxxxxx",
    "period_type" : 2,
    "period_num" : 1,
    "resource_id" : "xxxxxxxxxxxx",
    "product_info" : {
      "cloud_service_type" : "hws.service.type.sdg",
      "product_id" : "xxxxxxxxxxxx",
      "resource_size" : 30,
      "resource_size_measure_id" : 30,
      "resource_spec_code" : "base_professional",
      "resource_type" : "hws.resource.type.dsc.base"
    }
  } ]
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowSpecificationSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowSpecificationRequest request = new ShowSpecificationRequest();
        try {
            ShowSpecificationResponse response = client.showSpecification(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowSpecificationRequest()
        response = client.show_specification(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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowSpecificationRequest{}
    response, err := client.ShowSpecification(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.2 数据水印

3.2.1 嵌入数据水印

功能介绍

对json体中数据动态添加水印

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/sdg/database/watermark/embed

表 3-11 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-12 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）

表 3-13 请求 Body 参数

参数	是否必选	参数类型	描述
watermark_content	是	String	添加水印的内容
watermark_key	是	String	水印密钥
columns	是	Array of **Columns** objects	字段类型列表，最大长度100。使用时，至少包含两个字段，一个“primary_key”为true表示主键，一个为false用来嵌入水印
data	是	Array of Map<String,Object> objects	数据字段的内容，支持1500-50000行数据

表 3-14 Columns

参数	是否必选	参数类型	描述
name	是	String	数据的字段名称，最大支持长度256
type	是	String	数据的字段类型

参数	是否必选	参数类型	描述
primary_key	是	Boolean	标记该字段是否为主键。true为主键，表示用来定位水印位置；false为非主键，将在该列嵌入/提取水印内容。字段类型列表中可同时包含多个为true或为false的字段

响应参数

状态码：200

表 3-15 响应 Body 参数

参数	参数类型	描述
marked_data	Array of Map<String,Object> objects	嵌入水印后的数据

状态码：400

表 3-16 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

嵌入内容为test12345678test的水印，水印密钥是keyword，字段类型列表中数据的字段名称为item1，该字段为主键。

```
POST https://{endpoit}/v1/{project_id}/sdg/database/watermark/embed

{
  "watermark_content" : "test12345678test",
  "watermark_key" : "keyword",
  "columns" : [ {
    "name" : "item1",
    "type" : "INTEGER",
    "primary_key" : true
  }, {
    "name" : "item2",
    "type" : "INTEGER",
    "primary_key" : false
  } ],
  "data" : [ {
    "item1" : {
```

```
    "col" : 0
   },
   "item2" : {
    "col" : 3
   }
  }, {
   "item1" : {
    "col" : 1
   },
   "item2" : {
    "col" : 4
   }
  } ]
}
```

响应示例

状态码：200

请求成功

```
{
  "marked_data" : [ {
    "item2" : {
      "col2" : 3
    },
    "item1" : {
      "col1" : "test"
    }
  }, {
    "item2" : {
      "col2" : 5
    },
    "item1" : {
      "col1" : "test"
    }
  } ]
}
```

状态码：400

参数错误

```
{
  "error_code" : "DSC.00000004",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

嵌入内容为test12345678test的水印，水印密钥是keyword，字段类型列表中数据的字段名称为item1，该字段为主键。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
```

```
import com.huaweicloud.sdk.dsc.v1.model.*;

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class CreateDatabaseWaterMarkSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateDatabaseWaterMarkRequest request = new CreateDatabaseWaterMarkRequest();
        EmbeddedDatabaseWatermark body = new EmbeddedDatabaseWatermark();
        Map<String, Object> listDataData = new HashMap<>();
        listDataData.put("item1", "{\"col\":1}");
        listDataData.put("item2", "{\"col\":4}");
        Map<String, Object> listDataData1 = new HashMap<>();
        listDataData1.put("item1", "{\"col\":0}");
        listDataData1.put("item2", "{\"col\":3}");
        List<Map<String, Object>> listbodyData = new ArrayList<>();
        listbodyData.add(listDataData);
        listbodyData.add(listDataData1);
        List<Columns> listbodyColumns = new ArrayList<>();
        listbodyColumns.add(
            new Columns()
                .withName("item1")
                .withType(Columns.TypeEnum.fromValue("INTEGER"))
                .withPrimaryKey(true)
        );
        listbodyColumns.add(
            new Columns()
                .withName("item2")
                .withType(Columns.TypeEnum.fromValue("INTEGER"))
                .withPrimaryKey(false)
        );
        body.withData(listbodyData);
        body.withColumns(listbodyColumns);
        body.withWatermarkKey("keyword");
        body.withWatermarkContent("test12345678test");
        request.withBody(body);
        try {
            CreateDatabaseWaterMarkResponse response = client.createDatabaseWaterMark(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

嵌入内容为test12345678test的水印，水印密钥是keyword，字段类型列表中数据的字段名称为item1，该字段为主键。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateDatabaseWaterMarkRequest()
        listDataData = {
            "item1": "{\"col\":1}",
            "item2": "{\"col\":4}"
        }
        listDataData1 = {
            "item1": "{\"col\":0}",
            "item2": "{\"col\":3}"
        }
        listDatabody = [
            listDataData,
            listDataData1
        ]
        listColumnsbody = [
            Columns(
                name="item1",
                type="INTEGER",
                primary_key=True
            ),
            Columns(
                name="item2",
                type="INTEGER",
                primary_key=False
            )
        ]
        request.body = EmbeddedDatabaseWatermark(
            data=listDatabody,
            columns=listColumnsbody,
            watermark_key="keyword",
            watermark_content="test12345678test"
        )
        response = client.create_database_water_mark(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

嵌入内容为test12345678test的水印，水印密钥是keyword，字段类型列表中数据的字段名称为item1，该字段为主键。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDatabaseWaterMarkRequest{}
    var listDataData = map[string]interface{}{
        "item1": "{\"col\":1}",
        "item2": "{\"col\":4}",
    }
    var listDataData3 = map[string]interface{}{
        "item1": "{\"col\":0}",
        "item2": "{\"col\":3}",
    }
    var listDatabody = []map[string]interface{}{
        listDataData,
        listDataData3,
    }
    var listColumnsbody = []model.Columns{
        {
            Name: "item1",
            Type: model.GetColumnsTypeEnum().INTEGER,
            PrimaryKey: true,
        },
        {
            Name: "item2",
            Type: model.GetColumnsTypeEnum().INTEGER,
            PrimaryKey: false,
        },
    }
    request.Body = &model.EmbeddedDatabaseWatermark{
```

```
        Data: listDatabody,
        Columns: listColumnsbody,
        WatermarkKey: "keyword",
        WatermarkContent: "test12345678test",
    }
    response, err := client.CreateDatabaseWaterMark(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	请求成功
400	参数错误

错误码

请参见**错误码**。

3.2.2 提取数据水印

功能介绍

提取请求数据中水印内容

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/sdg/database/watermark/extract

表 3-17 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-18 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）

表 3-19 请求 Body 参数

参数	是否必选	参数类型	描述
watermark_key	是	String	水印密钥
columns	是	Array of **Columns** objects	字段类型列表，最大长度100。使用时，要包含嵌入时所有“primary_key”为true的字段，和至少一个为false的字段用来提取水印
data	是	Array of Map<String,Object> objects	水印数据，数据条数不超过30,000条

表 3-20 Columns

参数	是否必选	参数类型	描述
name	是	String	数据的字段名称，最大支持长度256
type	是	String	数据的字段类型
primary_key	是	Boolean	标记该字段是否为主键。true为主键，表示用来定位水印位置；false为非主键，将在该列嵌入/提取水印内容。字段类型列表中可同时包含多个为true或为false的字段

响应参数

状态码：200

表 3-21 响应 Body 参数

参数	参数类型	描述
watermarks	Array of strings	提取水印内容列表。上传数据中不同列可能包含不同水印，返回时将所有提取到的水印返回，列表中水印个数不超过100

状态码：400

表 3-22 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

提取数据水印，水印密钥是key，字段类型列表中数据的字段名称为col，该字段为非主键。

```
POST https://{endpoint}/v1/{project_id}/sdg/database/watermark/extract

{
  "watermark_key" : "key",
  "columns" : [ {
    "name" : "col",
    "type" : "INTEGER",
    "primary_key" : false
  } ],
  "data" : [ {
    "col" : {
      "a" : 0.1
    }
  } ]
}
```

响应示例

状态码：200

请求成功

```
{
  "watermarks" : [ "watermark" ]
}
```

状态码：400

参数错误

```
{
  "error_code" : "DSC.00000004",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

提取数据水印，水印密钥是key，字段类型列表中数据的字段名称为col，该字段为非主键。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class ShowDatabaseWaterMarkSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowDatabaseWaterMarkRequest request = new ShowDatabaseWaterMarkRequest();
        ExtractedDatabaseWatermark body = new ExtractedDatabaseWatermark();
        Map<String, Object> listDataData = new HashMap<>();
        listDataData.put("col", "{\"a\":0.1}");
        List<Map<String, Object>> listbodyData = new ArrayList<>();
        listbodyData.add(listDataData);
        List<Columns> listbodyColumns = new ArrayList<>();
        listbodyColumns.add(
            new Columns()
                .withName("col")
                .withType(Columns.TypeEnum.fromValue("INTEGER"))
                .withPrimaryKey(false)
        );
        body.withData(listbodyData);
        body.withColumns(listbodyColumns);
        body.withWatermarkKey("key");
        request.withBody(body);
        try {
            ShowDatabaseWaterMarkResponse response = client.showDatabaseWaterMark(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

提取数据水印，水印密钥是key，字段类型列表中数据的字段名称为col，该字段为非主键。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowDatabaseWaterMarkRequest()
        listDataData = {
            "col": "{\"a\":0.1}"
        }
        listDatabody = [
            listDataData
        ]
        listColumnsbody = [
            Columns(
                name="col",
                type="INTEGER",
                primary_key=False
            )
        ]
        request.body = ExtractedDatabaseWatermark(
            data=listDatabody,
            columns=listColumnsbody,
            watermark_key="key"
        )
        response = client.show_database_water_mark(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

提取数据水印，水印密钥是key，字段类型列表中数据的字段名称为col，该字段为非主键。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDatabaseWaterMarkRequest{}
    var listDataData = map[string]interface{}{
        "col": "{\"a\":0.1}",
    }
    var listDatabody = []map[string]interface{}{
        listDataData,
    }
    var listColumnsbody = []model.Columns{
        {
            Name: "col",
            Type: model.GetColumnsTypeEnum().INTEGER,
            PrimaryKey: false,
        },
    }
    request.Body = &model.ExtractedDatabaseWatermark{
        Data: listDatabody,
        Columns: listColumnsbody,
        WatermarkKey: "key",
    }
    response, err := client.ShowDatabaseWaterMark(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	请求成功
400	参数错误

错误码

请参见**错误码**。

3.3 数据动态脱敏

3.3.1 对数据进行脱敏

功能介绍

对数据进行脱敏

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/data/mask

表 3-23 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-24 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）

表 3-25 请求 Body 参数

参数	是否必选	参数类型	描述
mask_strategies	是	Array of **MaskStrategies** objects	脱敏策略列表，每一个策略对应一个字段，脱敏策略数最多100个。
data	是	Array of Map<String,Object> objects	数据列表。

表 3-26 MaskStrategies

参数	是否必选	参数类型	描述
name	是	String	需要脱敏的字段名称，最大支持长度256。
algorithm	是	String	脱敏算法名称，详情见附录"动态脱敏策略配置"。
parameters	否	Map<String,Object>	脱敏算法参数，详情见附录"动态脱敏策略配置"。

响应参数

状态码：200

表 3-27 响应 Body 参数

参数	参数类型	描述
masked_data	Array of Map<String,Object> objects	脱敏后的数据的数据列表，结构与请求中结构相同

状态码：400

表 3-28 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

脱敏策略列表中对字段col使用KEYWORD脱敏算法，将指定关键字keyword替换为target。

```
POST https://{endpoint}/v1/{project_id}/data/mask

{
  "mask_strategies" : [ {
    "name" : "col",
    "algorithm" : "KEYWORD",
    "parameters" : {
      "key" : {
        "keyword" : "keyword"
      },
      "target" : {
        "target" : "target"
      }
    }
  } ],
  "data" : [ {
    "col" : {
      "keyword" : "keyword"
    }
  } ]
}
```

响应示例

状态码：200

脱敏成功

```
{
  "masked_data" : [ {
    "col" : {
      "a" : "target"
    }
  } ]
}
```

状态码：400

无效请求

```
{
  "error_code" : "DSC.00000004",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

脱敏策略列表中对字段col使用KEYWORD脱敏算法，将指定关键字keyword替换为target。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class BatchAddDataMaskSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchAddDataMaskRequest request = new BatchAddDataMaskRequest();
        DynamicDataMask body = new DynamicDataMask();
        Map<String, Object> listDataData = new HashMap<>();
        listDataData.put("col", "{\"keyword\":\"keyword\"}");
        List<Map<String, Object>> listbodyData = new ArrayList<>();
        listbodyData.add(listDataData);
        Map<String, Object> listMaskStrategiesParameters = new HashMap<>();
        listMaskStrategiesParameters.put("key", "{\"keyword\":\"keyword\"}");
        listMaskStrategiesParameters.put("target", "{\"target\":\"target\"}");
        List<MaskStrategies> listbodyMaskStrategies = new ArrayList<>();
        listbodyMaskStrategies.add(
            new MaskStrategies()
                .withName("col")
                .withAlgorithm(MaskStrategies.AlgorithmEnum.fromValue("KEYWORD"))
                .withParameters(listMaskStrategiesParameters)
        );
        body.withData(listbodyData);
        body.withMaskStrategies(listbodyMaskStrategies);
        request.withBody(body);
        try {
            BatchAddDataMaskResponse response = client.batchAddDataMask(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

脱敏策略列表中对字段col使用KEYWORD脱敏算法，将指定关键字keyword替换为target。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchAddDataMaskRequest()
        listDataData = {
            "col": "{\"keyword\":\"keyword\"}"
        }
        listDatabody = [
            listDataData
        ]
        listParametersMaskStrategies = {
            "key": "{\"keyword\":\"keyword\"}",
            "target": "{\"target\":\"target\"}"
        }
        listMaskStrategiesbody = [
            MaskStrategies(
                name="col",
                algorithm="KEYWORD",
                parameters=listParametersMaskStrategies
            )
        ]
        request.body = DynamicDataMask(
            data=listDatabody,
            mask_strategies=listMaskStrategiesbody
        )
        response = client.batch_add_data_mask(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

脱敏策略列表中对字段col使用KEYWORD脱敏算法，将指定关键字keyword替换为target。

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchAddDataMaskRequest{}
    var listDataData = map[string]interface{}{
        "col": "{\"keyword\":\"keyword\"}",
    }
    var listDatabody = []map[string]interface{}{
        listDataData,
    }
    var listParametersMaskStrategies = map[string]interface{}{
        "key": "{\"keyword\":\"keyword\"}",
        "target": "{\"target\":\"target\"}",
    }
    var listMaskStrategiesbody = []model.MaskStrategies{
        {
            Name: "col",
            Algorithm: model.GetMaskStrategiesAlgorithmEnum().KEYWORD,
            Parameters: listParametersMaskStrategies,
        },
    }
    request.Body = &model.DynamicDataMask{
        Data: listDatabody,
        MaskStrategies: listMaskStrategiesbody,
    }
    response, err := client.BatchAddDataMask(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	脱敏成功
400	无效请求

错误码

请参见**错误码**。

3.4 告警通知

3.4.1 查询告警通知主题

功能介绍

查询告警通知主题，返回默认主题、已确认主题数量及列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/sdg/smn/topics

表 3-29 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 3-30 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小

请求参数

无

响应参数

状态码：200

表 3-31 响应 Body 参数

参数	参数类型	描述
id	String	DSC告警主题ID（非消息通知服务主题ID）
default_topic_urn	String	默认消息通知主题的唯一资源标识符
topic_count	Integer	已确认的消息通知主题数量
topics	Array of **TopicBean** objects	已确认的消息通知主题列表

表 3-32 TopicBean

参数	参数类型	描述
name	String	消息通知主题名称
topic_urn	String	消息通知主题的唯一资源标识符

状态码：400

表 3-33 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询告警通知主题

```
GET /v1/{project_id}/sdg/smn/topics
```

响应示例

状态码：200

请求成功

```
{
  "id" : "xxxxxx",
  "default_topic_urn" : "xxxxxx",
  "topic_count" : 1,
```

```
 "topics" : [ {
   "name" : "xxxxxx",
   "topic_urn" : "xxxxxx"
 } ]
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowTopicsSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowTopicsRequest request = new ShowTopicsRequest();
        try {
            ShowTopicsResponse response = client.showTopics(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowTopicsRequest()
        response = client.show_topics(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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

```
    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowTopicsRequest{}
    response, err := client.ShowTopics(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.4.2 修改告警通知主题

功能介绍

修改告警通知的关联项目ID、通知主题、通知状态(0为关闭通知，1为开启通知)等通用配置

调用方法

请参见**如何调用API**。

URI

PUT /v1/{project_id}/sdg/smn/topic

表 3-34 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-35 请求 Body 参数

参数	是否必选	参数类型	描述
id	是	String	DSC告警主题ID（非消息通知服务主题ID）
project_id	否	String	项目ID
status	否	Integer	告警通知状态
topic_urn	是	String	消息通知主题的唯一资源标识符

响应参数

状态码：200

表 3-36 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息
status	String	返回状态，如'200','400'

状态码：400

表 3-37 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

修改xxxxxxxxxxxxxxxxxxxx为通知主题，项目ID为xxxxxxxxxxxxxxxxxxx，告警通知状态为开启。

```
PUT /v1/{project_id}/sdg/smn/topic

{
  "id" : "xxxxxxxxxxxxxxxxxx",
  "project_id" : "xxxxxxxxxxxxxxxxxxx",
  "status" : 1,
  "topic_urn" : "xxxxxxxxxxxxxxxxxxx"
}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改xxxxxxxxxxxxxxxxxxxx为通知主题，项目ID为xxxxxxxxxxxxxxxxxxx，告警通知状态为开启。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class UpdateDefaultTopicSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateDefaultTopicRequest request = new UpdateDefaultTopicRequest();
        DefaultTopicRequest body = new DefaultTopicRequest();
        body.withTopicUrn("xxxxxxxxxxxxxxxxxxx");
        body.withStatus(1);
```

```
        body.withProjectId("xxxxxxxxxxxxxxxxxxx");
        body.withId("xxxxxxxxxxxxxxxxxxx");
        request.withBody(body);
        try {
            UpdateDefaultTopicResponse response = client.updateDefaultTopic(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xxxxxxxxxxxxxxxxxxx为通知主题，项目ID为xxxxxxxxxxxxxxxxxxx，告警通知状态为开启。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateDefaultTopicRequest()
        request.body = DefaultTopicRequest(
            topic_urn="xxxxxxxxxxxxxxxxxxx",
            status=1,
            project_id="xxxxxxxxxxxxxxxxxxx",
            id="xxxxxxxxxxxxxxxxxxx"
        )
        response = client.update_default_topic(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xxxxxxxxxxxxxxxxxxxx为通知主题，项目ID为xxxxxxxxxxxxxxxxxxx，告警通知状态为开启。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateDefaultTopicRequest{}
    statusDefaultTopicRequest:= int32(1)
    projectIdDefaultTopicRequest:= "xxxxxxxxxxxxxxxxxx"
    request.Body = &model.DefaultTopicRequest{
        TopicUrn: "xxxxxxxxxxxxxxxxxx",
        Status: &statusDefaultTopicRequest,
        ProjectId: &projectIdDefaultTopicRequest,
        Id: "xxxxxxxxxxxxxxxxxx",
    }
    response, err := client.UpdateDefaultTopic(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	请求成功

状态码	描述
400	无效请求

错误码

请参见**错误码**。

3.5 图片水印

3.5.1 图片嵌入暗水印

功能介绍

对图片嵌入文字暗水印或者图片暗水印，用户以formData的格式传入待加水印图片和水印相关信息，DSC服务对图片加完水印后返回给用户已嵌入水印的图片二进制流，目前支持的图片格式为：*.jpg, *.bmp, *.png, *.jpeg, *.tiff, *.tif, *.tga, *.gif。 不支持过小图片

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/image/watermark/embed

表 3-38 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-39 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）

表 3-40 FormData 参数

参数	是否必选	参数类型	描述
file	是	File	要添加水印的图片文件，添加的图片短边尺寸需要超过512像素。
blind_watermark	否	String	待嵌入的文字暗水印内容，长度不超过32个字符。当前仅支持数字及英文大小写。与图片暗水印image_watermark二选一填充。
image_watermark	否	File	待嵌入的图片暗水印文件，与文字暗水印 blind_watermark 二选一填充。

响应参数

状态码：200

表 3-41 响应 Body 参数

参数	参数类型	描述
-	File	

状态码：400

表 3-42 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

给test.PNG图片添加testWaterMark的文字暗水印。 > 通过form表单提交，file为具体文件。

```
POST /v1/{project_id}/image/watermark/embed HTTP/1.1

{
  "blind_watermark" : "testWaterMark",
  "file" : "test.PNG"
}
```

响应示例

状态码：200

请求成功

```
"{\"Watermarked image\"}"
```

状态码：400

无效请求

```
{
  "error_code" : "DSC.00000007",
  "error_msg" : "File format error"
}
```

SDK 代码示例

SDK代码示例如下。

Java

给test.PNG图片添加testWaterMark的文字暗水印。 > 通过form表单提交，file为具体文件。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class CreateImageWatermarkSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateImageWatermarkRequest request = new CreateImageWatermarkRequest();
        CreateImageWatermarkRequestBody bodybody = new CreateImageWatermarkRequestBody();
        bodybody.withFile("test.PNG")
            .withBlindWatermark("testWaterMark");
        body.withBody(bodybody);
        request.withBody(listbodyBody);
        try {
```

```
            CreateImageWatermarkResponse response = client.createImageWatermark(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

给test.PNG图片添加testWaterMark的文字暗水印。 > 通过form表单提交，file为具体文件。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateImageWatermarkRequest()
        bodybody = CreateImageWatermarkRequestBody(
            file="test.PNG",
            blind_watermark="testWaterMark"
        )
        request.body = listBodybody
        response = client.create_image_watermark(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

给test.PNG图片添加testWaterMark的文字暗水印。 > 通过form表单提交，file为具体文件。

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateImageWatermarkRequest{}
    blindWatermarkBody:= "testWaterMark"
    bodybody := &model.CreateImageWatermarkRequestBody{
        File: "test.PNG",
        BlindWatermark: &blindWatermarkBody,
    }
    request.Body = listBodybody
    response, err := client.CreateImageWatermark(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.5.2 提取图片中的文字暗水印

功能介绍

对已嵌入文字暗水印的图片进行水印提取，用户以formData的格式传入待提取水印的图片，DSC服务以JSON的格式返回从图片里提取的出的文字暗水印。目前支持的图片格式为：*.jpg, *.bmp, *.png, *.jpeg, *.tiff, *.tif, *.tga, *.gif。

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/image/watermark/extract

表 3-43 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-44 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）

表 3-45 FormData 参数

参数	是否必选	参数类型	描述
file	是	File	待提取暗水印的图片文件。
mark_len	否	String	指定待提取水印的长度，mark_len长度大于0，小于32。设置后可以提升水印提取性能

响应参数

状态码：200

表 3-46 响应 Body 参数

参数	参数类型	描述
watermark	String	暗水印内容，长度不超过32个字节

状态码：400

表 3-47 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

提取test.PNG图片中的文字暗水印。 > 通过form表单提交，file为具体文件。

```
POST /v1/{project_id}/image/watermark/extract HTTP/1.1

{
  "file" : "test.PNG"
}
```

响应示例

状态码：200

请求成功

```
{
  "watermark" : "mark!"
}
```

状态码：400

无效请求

```
{
  "error_code" : "DSC.00000007 ",
  "error_msg" : "File format error"
}
```

SDK 代码示例

SDK代码示例如下。

Java

提取test.PNG图片中的文字暗水印。 > 通过form表单提交，file为具体文件。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowImageWatermarkSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowImageWatermarkRequest request = new ShowImageWatermarkRequest();
        ShowImageWatermarkRequestBody bodybody = new ShowImageWatermarkRequestBody();
        bodybody.withFile("test.PNG");
        body.withBody(bodybody);
        request.withBody(listbodyBody);
        try {
            ShowImageWatermarkResponse response = client.showImageWatermark(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

提取test.PNG图片中的文字暗水印。 > 通过form表单提交，file为具体文件。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowImageWatermarkRequest()
        bodybody = ShowImageWatermarkRequestBody(
            file="test.PNG"
        )
        request.body = listBodybody
        response = client.show_image_watermark(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

提取test.PNG图片中的文字暗水印。 > 通过form表单提交，file为具体文件。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowImageWatermarkRequest{}
    bodybody := &model.ShowImageWatermarkRequestBody{
        File: "test.PNG",
    }
    request.Body = listBodybody
    response, err := client.ShowImageWatermark(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.5.3 提取图片中的图片暗水印

功能介绍

对已嵌入图片暗水印的图片进行水印提取，用户以formData的格式传入待提取水印的图片，DSC服务以图片二进制流的格式返回从图片里提取的出的图片暗水印。目前支持的图片格式为：*.jpg, *.bmp, *.png, *.jpeg, *.tiff, *.tif, *.tga, *.gif。

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/image/watermark/extract-image

表 3-48 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-49 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）

表 3-50 FormData 参数

参数	是否必选	参数类型	描述
file	是	File	待提取暗水印的图片文件。

响应参数

状态码：200

表 3-51 响应 Body 参数

参数	参数类型	描述
-	File	

状态码：400

表 3-52 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

提取test.PNG图片中的图片暗水印。 > 通过form表单提交，file为具体文件。

```
POST /v1/{project_id}/image/watermark/extract-image HTTP/1.1

{
  "file" : "test.PNG"
}
```

响应示例

状态码：200

请求成功

```
"{\"image\"}"
```

状态码：400

无效请求

```
{
  "error_code" : "DSC.00000007 ",
  "error_msg" : "File format error"
}
```

SDK 代码示例

SDK代码示例如下。

Java

提取test.PNG图片中的图片暗水印。 > 通过form表单提交，file为具体文件。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowImageWatermarkWithImageSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowImageWatermarkWithImageRequest request = new ShowImageWatermarkWithImageRequest();
        ShowImageWatermarkWithImageRequestBody bodybody = new
ShowImageWatermarkWithImageRequestBody();
        bodybody.withFile("test.PNG");
        body.withBody(bodybody);
        request.withBody(listbodyBody);
        try {
            ShowImageWatermarkWithImageResponse response =
client.showImageWatermarkWithImage(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

提取test.PNG图片中的图片暗水印。 > 通过form表单提交，file为具体文件。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowImageWatermarkWithImageRequest()
        bodybody = ShowImageWatermarkWithImageRequestBody(
            file="test.PNG"
        )
        request.body = listBodybody
        response = client.show_image_watermark_with_image(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

提取test.PNG图片中的图片暗水印。 > 通过form表单提交，file为具体文件。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
```

```
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowImageWatermarkWithImageRequest{}
    bodybody := &model.ShowImageWatermarkWithImageRequestBody{
        File: "test.PNG",
    }
    request.Body = listBodybody
    response, err := client.ShowImageWatermarkWithImage(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.5.4 图片嵌入暗水印（文件地址版本）

功能介绍

对指定存储地址信息（目前支持OBS）的图片嵌入文字暗水印或者图片暗水印，已嵌入的水印的图片将存放在用户指定的位置（目前支持OBS），支持的图片格式为：*.jpg, *.bmp, *.png, *.jpeg, *.tiff, *.tif, *.tga, *.gif。 不支持过小图片

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/image-address/watermark/embed

表 3-53 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-54 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）

表 3-55 请求 Body 参数

参数	是否必选	参数类型	描述
region_id	是	String	当前项目所在region的id，如：xx-xx-1。

参数	是否必选	参数类型	描述
src_file	是	String	待加暗水印的图片地址，当前只支持OBS文件，格式为 **obs://bucket/object** ，其中bucket为和当前项目处于同一区域的OBS桶名称，object为对象全路径名。例如：**obs://hwbucket/hwinfo/hw.png**，其中obs://表示OBS存储，hwbucket为桶名，hwinfo/hw.png为对象全路径名。
blind_watermark	否	String	待嵌入的文字暗水印内容，长度不超过32个字符。当前仅支持数字及英文大小写。与图片暗水印image_watermark二选一。
image_watermark	否	String	待嵌入的图片暗水印地址，格式要求同src_file字段，与文字暗水印 blind_watermark 二选一，都填写时，生效image_watermark。
dst_file	否	String	添加水印后的图片存放的地址，格式要求同src_file字段，不设置时，默认取src_file的值，即添加水印后覆盖原文件。

响应参数

状态码：200

表 3-56 响应 Body 参数

参数	参数类型	描述
region_id	String	当前项目所在region的id，如：xx-xx-1。
watermarked_file	String	添加水印后的OBS图片地址，当前只支持OBS文件，格式为 **obs://bucket/object** ，其中bucket为和当前项目处于同一区域的OBS桶名称，object为对象全路径名。例如：**obs://hwbucket/hwinfo/hw.png**，其中obs://表示OBS存储，hwbucket为桶名，hwinfo/hw.png为对象全路径名。

状态码：400

表 3-57 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

在xx-xx-1区域，给obs://bucket/test.png路径下的图片嵌入testWaterMark的暗水印。

```
POST /v1/{project_id}/image-address/watermark/embed HTTP/1.1

{
  "region_id" : "xx-xx-1",
  "src_file" : "obs://bucket/test.png",
  "blind_watermark" : "testWaterMark"
}
```

响应示例

状态码：200

请求成功

```
{
  "region_id" : "xx-xx-1",
  "watermarked_file" : "obs://bucket/test.png"
}
```

状态码：400

无效请求

```
{
  "error_code" : "DSC.00000007",
  "error_msg" : "File format error"
}
```

SDK 代码示例

SDK代码示例如下。

Java

在xx-xx-1区域，给obs://bucket/test.png路径下的图片嵌入testWaterMark的暗水印。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class CreateImageWatermarkByAddressSolution {
```

```
    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateImageWatermarkByAddressRequest request = new CreateImageWatermarkByAddressRequest();
        CreateImageWatermarkByAddressRequestBody body = new CreateImageWatermarkByAddressRequestBody();
        body.withBlindWatermark("testWaterMark");
        body.withSrcFile("obs://bucket/test.png");
        body.withRegionId("xx-xx-1");
        request.withBody(body);
        try {
            CreateImageWatermarkByAddressResponse response = client.createImageWatermarkByAddress(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

在xx-xx-1区域，给obs://bucket/test.png路径下的图片嵌入testWaterMark的暗水印。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
    client = DscClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DscRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateImageWatermarkByAddressRequest()
        request.body = CreateImageWatermarkByAddressRequestBody(
            blind_watermark="testWaterMark",
            src_file="obs://bucket/test.png",
            region_id="xx-xx-1"
        )
        response = client.create_image_watermark_by_address(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

在xx-xx-1区域，给obs://bucket/test.png路径下的图片嵌入testWaterMark的暗水印。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateImageWatermarkByAddressRequest{}
    blindWatermarkCreateImageWatermarkByAddressRequestBody:= "testWaterMark"
    request.Body = &model.CreateImageWatermarkByAddressRequestBody{
        BlindWatermark: &blindWatermarkCreateImageWatermarkByAddressRequestBody,
        SrcFile: "obs://bucket/test.png",
        RegionId: "xx-xx-1",
    }
    response, err := client.CreateImageWatermarkByAddress(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.5.5 提取图片中的文字暗水印（文件地址版本）

功能介绍

对指定存储地址信息（目前支持OBS）的已嵌入文字暗水印的图片提取文字暗水印，支持的图片格式为：*.jpg, *.bmp, *.png, *.jpeg, *.tiff, *.tif, *.tga, *.gif。

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/image-address/watermark/extract

表 3-58 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-59 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）

表 3-60 请求 Body 参数

参数	是否必选	参数类型	描述
region_id	是	String	项目所在region的id，如：xx-xx-1。
src_file	是	String	待提取文字暗水印的图片地址，当前只支持OBS，格式为 **obs://bucket/object** ，其中bucket为和当前项目处于同一区域的OBS桶名称，object为对象全路径名。例如：**obs://hwbucket/hwinfo/hw.png**，其中obs://表示OBS存储，hwbucket为桶名，hwinfo/hw.png为对象全路径名。
mark_len	否	Integer	指定待提取水印的长度，最小0，最大32.。设置后可以提升水印提取性能。

响应参数

状态码：200

表 3-61 响应 Body 参数

参数	参数类型	描述
watermark	String	暗水印内容，长度不超过32个字节

状态码：400

表 3-62 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

提取obs://bucket/info/wm.png路径下的图片中的文字暗水印。

```
POST /v1/{project_id}/image-address/watermark/extract HTTP/1.1

{
  "region_id" : "xx-xx-1",
  "src_file" : "obs://bucket/info/wm.png"
}
```

响应示例

状态码：200

请求成功

```
{
  "watermark" : "mark!"
}
```

状态码：400

无效请求

```
{
  "error_code" : "DSC.00000007 ",
  "error_msg" : "File format error"
}
```

SDK 代码示例

SDK代码示例如下。

Java

提取obs://bucket/info/wm.png路径下的图片中的文字暗水印。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowImageWatermarkByAddressSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowImageWatermarkByAddressRequest request = new ShowImageWatermarkByAddressRequest();
        ShowImageWatermarkByAddressRequestBody body = new
ShowImageWatermarkByAddressRequestBody();
        body.withSrcFile("obs://bucket/info/wm.png");
        body.withRegionId("xx-xx-1");
        request.withBody(body);
        try {
            ShowImageWatermarkByAddressResponse response =
client.showImageWatermarkByAddress(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

提取obs://bucket/info/wm.png路径下的图片中的文字暗水印。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DscClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DscRegion.value_of("<YOUR REGION>")) \
```

```
        .build()

    try:
        request = ShowImageWatermarkByAddressRequest()
        request.body = ShowImageWatermarkByAddressRequestBody(
            src_file="obs://bucket/info/wm.png",
            region_id="xx-xx-1"
        )
        response = client.show_image_watermark_by_address(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

提取obs://bucket/info/wm.png路径下的图片中的文字暗水印。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowImageWatermarkByAddressRequest{}
    request.Body = &model.ShowImageWatermarkByAddressRequestBody{
        SrcFile: "obs://bucket/info/wm.png",
        RegionId: "xx-xx-1",
    }
    response, err := client.ShowImageWatermarkByAddress(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.5.6 提取图片中的图片暗水印（文件地址版本）

功能介绍

对指定存储地址信息（目前支持OBS）的已嵌入图片暗水印的图片提取图片暗水印，提取出的水印图片将存放在用户指定的位置（目前支持OBS），支持的图片格式为：*.jpg, *.bmp, *.png, *.jpeg, *.tiff, *.tif, *.tga, *.gif。

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/image-address/watermark/extract-image

表 3-63 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-64 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）

表 3-65 请求 Body 参数

参数	是否必选	参数类型	描述
region_id	是	String	项目所在region的id，如：xx-xx-1。
src_file	是	String	待提取图片暗水印的图片地址，当前只支持OBS对象，格式为 **obs://bucket/object** ，其中bucket为和当前项目处于同一区域的OBS桶名称，object为对象全路径名。例如：**obs://hwbucket/hwinfo/hw.png**，其中obs://表示OBS存储，hwbucket为桶名，hwinfo/hw.png为对象全路径名。
image_watermark	是	String	提取出来的水印图片存放地址，格式要求同src_file。

响应参数

状态码：200

表 3-66 响应 Body 参数

参数	参数类型	描述
region_id	String	当前项目所在region的id，如：xx-xx-1。
image_watermark	String	提取出的水印图片存放地址，当前只支持OBS对象，格式为 **obs://bucket/object** ，其中bucket为和当前项目处于同一区域的OBS桶名称，object为对象全路径名。例如：**obs://hwbucket/hwinfo/hw.png**，其中obs://表示OBS存储，hwbucket为桶名，hwinfo/hw.png为对象全路径名。

状态码：400

表 3-67 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	错误信息

请求示例

提取obs://bucket/info/wm.png路径下的图片中的图片暗水印，将提取的水印图片存放在obs://bucket/watermarkfile/mark.png路径下。

```
POST /v1/{project_id}/obs-image/image-watermark/extract HTTP/1.1

{
  "region_id" : "xx-xx-1",
  "src_file" : "obs://bucket/info/wm.png",
  "image_watermark" : "obs://bucket/watermarkfile/mark.png"
}
```

响应示例

状态码：200

请求成功

```
{
  "region_id" : "xx-xx-1",
  "image_watermark" : "obs://bucket/watermarkfile/mark.png"
}
```

状态码：400

无效请求

```
{
  "error_code" : "DSC.00000007 ",
  "error_msg" : "File format error"
}
```

SDK 代码示例

SDK代码示例如下。

Java

提取obs://bucket/info/wm.png路径下的图片中的图片暗水印，将提取的水印图片存放在obs://bucket/watermarkfile/mark.png路径下。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowImageWatermarkWithImageByAddressSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowImageWatermarkWithImageByAddressRequest request = new
ShowImageWatermarkWithImageByAddressRequest();
        ShowImageWatermarkWithImageByAddressRequestBody body = new
ShowImageWatermarkWithImageByAddressRequestBody();
        body.withImageWatermark("obs://bucket/watermarkfile/mark.png");
        body.withSrcFile("obs://bucket/info/wm.png");
        body.withRegionId("xx-xx-1");
        request.withBody(body);
        try {
            ShowImageWatermarkWithImageByAddressResponse response =
client.showImageWatermarkWithImageByAddress(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

提取obs://bucket/info/wm.png路径下的图片中的图片暗水印，将提取的水印图片存放在obs://bucket/watermarkfile/mark.png路径下。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
    client = DscClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DscRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowImageWatermarkWithImageByAddressRequest()
        request.body = ShowImageWatermarkWithImageByAddressRequestBody(
            image_watermark="obs://bucket/watermarkfile/mark.png",
            src_file="obs://bucket/info/wm.png",
            region_id="xx-xx-1"
        )
        response = client.show_image_watermark_with_image_by_address(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

提取obs://bucket/info/wm.png路径下的图片中的图片暗水印，将提取的水印图片存放在obs://bucket/watermarkfile/mark.png路径下。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowImageWatermarkWithImageByAddressRequest{}
    request.Body = &model.ShowImageWatermarkWithImageByAddressRequestBody{
        ImageWatermark: "obs://bucket/watermarkfile/mark.png",
        SrcFile: "obs://bucket/info/wm.png",
        RegionId: "xx-xx-1",
    }
    response, err := client.ShowImageWatermarkWithImageByAddress(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.6 资产管理

3.6.1 编辑资产名称

功能介绍

编辑数据资产名称

调用方法

请参见**如何调用API**。

URI

PUT /v1/{project_id}/sdg/asset/{asset_id}/name

表 3-68 路径参数

参数	是否必选	参数类型	描述
asset_id	是	String	资产ID
project_id	是	String	项目ID

请求参数

表 3-69 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	资产名称

响应参数

状态码：200

表 3-70 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息
status	String	返回状态，如'200','400'

状态码：400

表 3-71 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

编辑数据资产的名称为xxxxxxx。

```
PUT /v1/{project_id}/sdg/asset/{asset_id}/name

{
  "name" : "xxxxxxx"
}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

编辑数据资产的名称为xxxxxxx。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class UpdateAssetNameSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateAssetNameRequest request = new UpdateAssetNameRequest();
        request.withAssetId("{asset_id}");
        AssetNameRequest body = new AssetNameRequest();
        body.withName("xxxxxxx");
        request.withBody(body);
        try {
            UpdateAssetNameResponse response = client.updateAssetName(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

编辑数据资产的名称为xxxxxxx。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdateAssetNameRequest()
        request.asset_id = "{asset_id}"
        request.body = AssetNameRequest(
            name="xxxxxxx"
        )
        response = client.update_asset_name(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

编辑数据资产的名称为xxxxxxx。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
```

```
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateAssetNameRequest{}
    request.AssetId = "{asset_id}"
    nameAssetNameRequest:= "xxxxxxx"
    request.Body = &model.AssetNameRequest{
        Name: &nameAssetNameRequest,
    }
    response, err := client.UpdateAssetName(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.6.2 查看资产列表

功能介绍

查询数据资产扫描授权列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/sdg/asset/obs/buckets

表 3-72 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 3-73 Query 参数

参数	是否必选	参数类型	描述
added	否	Boolean	已授权
offset	否	Integer	页码
limit	否	Integer	分页大小

请求参数

无

响应参数

状态码：200

表 3-74 响应 Body 参数

参数	参数类型	描述
buckets	Array of **Bucket** objects	OBS桶列表
total	Integer	OBS桶总数

表 3-75 Bucket

参数	参数类型	描述
asset_name	String	资产名称
bucket_location	String	桶位置
bucket_name	String	桶名称
bucket_policy	String	桶策略
create_time	Long	创建时间
deleted	Boolean	是否被删除
id	String	桶ID

参数	参数类型	描述
is_deleted	Boolean	是否被删除

状态码：400

表 3-76 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询数据资产扫描授权列表

```
GET /v1/{project_id}/sdg/asset/obs/buckets
```

响应示例

状态码：200

请求成功

```
{
  "buckets" : [ {
    "bucket_name" : "xxxx",
    "bucket_location" : "xxxx",
    "create_time" : 1667379757698,
    "is_deleted" : false
  } ],
  "total" : 1
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ListBucketsSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ListBucketsRequest request = new ListBucketsRequest();
        try {
            ListBucketsResponse response = client.listBuckets(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DscClient.new_builder() \
        .with_credentials(credentials) \
```

```
        .with_region(DscRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListBucketsRequest()
        response = client.list_buckets(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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListBucketsRequest{}
    response, err := client.ListBuckets(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.6.3 添加资产授权

功能介绍

添加数据资产扫描授权

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/sdg/asset/obs/buckets

表 3-77 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 3-78 Query 参数

参数	是否必选	参数类型	描述
type	否	String	资产类型

请求参数

表 3-79 请求 Body 参数

参数	是否必选	参数类型	描述
buckets	否	Array of **BucketBean** objects	OBS桶列表

表 3-80 BucketBean

参数	是否必选	参数类型	描述
asset_name	否	String	资产名称
location	否	String	桶位置
bucket_name	否	String	桶名称
bucket_policy	否	String	桶策略

响应参数

状态码：200

表 3-81 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息
status	String	返回状态，如'200','400'

状态码：400

表 3-82 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

添加桶名称为xxxx的数据资产扫描授权。

```
POST /v1/{project_id}/sdg/asset/obs/buckets

{
  "buckets" : [ {
    "asset_name" : "xxxx",
    "location" : "xxxx",
    "bucket_name" : "xxxx",
    "bucket_policy" : "private"
  } ]
}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

添加桶名称为xxxx的数据资产扫描授权。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;

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

public class AddBucketsSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        AddBucketsRequest request = new AddBucketsRequest();
        BucketsRequest body = new BucketsRequest();
        List<BucketBean> listbodyBuckets = new ArrayList<>();
        listbodyBuckets.add(
            new BucketBean()
                .withAssetName("xxxx")
                .withLocation("xxxx")
                .withBucketName("xxxx")
```

```
                .withBucketPolicy("private")
        );
        body.withBuckets(listbodyBuckets);
        request.withBody(body);
        try {
            AddBucketsResponse response = client.addBuckets(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

添加桶名称为xxxx的数据资产扫描授权。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddBucketsRequest()
        listBucketsbody = [
            BucketBean(
                asset_name="xxxx",
                location="xxxx",
                bucket_name="xxxx",
                bucket_policy="private"
            )
        ]
        request.body = BucketsRequest(
            buckets=listBucketsbody
        )
        response = client.add_buckets(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

添加桶名称为xxxx的数据资产扫描授权。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddBucketsRequest{}
    assetNameBuckets:= "xxxx"
    locationBuckets:= "xxxx"
    bucketNameBuckets:= "xxxx"
    bucketPolicyBuckets:= "private"
    var listBucketsbody = []model.BucketBean{
        {
            AssetName: &assetNameBuckets,
            Location: &locationBuckets,
            BucketName: &bucketNameBuckets,
            BucketPolicy: &bucketPolicyBuckets,
        },
    }
    request.Body = &model.BucketsRequest{
        Buckets: &listBucketsbody,
    }
    response, err := client.AddBuckets(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.6.4 删除资产授权

功能介绍

删除数据资产扫描授权

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/sdg/asset/obs/bucket/{bucket_id}

表 3-83 路径参数

参数	是否必选	参数类型	描述
bucket_id	是	String	桶ID
project_id	是	String	项目ID

请求参数

无

响应参数

状态码：200

表 3-84 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息
status	String	返回状态，如'200','400'

状态码：400

表 3-85 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

删除数据资产扫描授权

```
DELETE /v1/{project_id}/sdg/asset/obs/bucket/{bucket_id}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class DeleteBucketSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteBucketRequest request = new DeleteBucketRequest();
        request.withBucketId("{bucket_id}");
        try {
            DeleteBucketResponse response = client.deleteBucket(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteBucketRequest()
        request.bucket_id = "{bucket_id}"
        response = client.delete_bucket(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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteBucketRequest{}
    request.BucketId = "{bucket_id}"
    response, err := client.DeleteBucket(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.7 敏感数据发现

3.7.1 查询扫描任务列表

功能介绍

查询扫描任务列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/sdg/scan/jobs

表 3-86 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 3-87 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小
content	否	String	内容

请求参数

无

响应参数

状态码：200

表 3-88 响应 Body 参数

参数	参数类型	描述
tasks	Array of **ScanJob** objects	本次返回的扫描任务列表

参数	参数类型	描述
total	Long	任务总数

表 3-89 ScanJob

参数	参数类型	描述
id	String	任务ID
name	String	任务名称
rule_groups	Array of strings	任务使用的规则组
scan_templates	Map<String,String>	任务使用的模板
cycle	String	任务执行方式
status	String	任务当前状态
last_run_time	Long	任务上一次执行时间
create_time	Long	任务创建时间
last_scan_risk	String	任务上一次扫描风险等级结果
use_nlp	Boolean	是否使用了NLP进行扫描
open	Boolean	任务开启状态
topic_urn	String	SMN服务通知主题
start_time	Long	任务启动时间
security_level_name	String	识别结果风险等级名称
security_level_color	Long	识别结果风险等级
asset_infos	Array of **AssetInfo** objects	资产列表

表 3-90 AssetInfo

参数	参数类型	描述
asset_id	String	资产ID
asset_type	String	资产类型

状态码：400

表 3-91 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询扫描任务列表

```
GET /v1/{project_id}/sdg/scan/jobs
```

响应示例

状态码：200

OK

```
{
  "total" : 1,
  "tasks" : [ {
    "id" : "xxxxxxx",
    "name" : "ScanDemo",
    "cycle" : "ONCE",
    "status" : "FINISHED",
    "open" : true,
    "rule_groups" : [ "PCI" ],
    "last_run_time" : 1634612489173,
    "create_time" : 1630982438506,
    "last_scan_risk" : "HIGH",
    "use_nlp" : false,
    "topic_urn" : "",
    "start_time" : 1630983532673
  } ]
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
```

```
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowScanJobsSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowScanJobsRequest request = new ShowScanJobsRequest();
        try {
            ShowScanJobsResponse response = client.showScanJobs(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DscClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(DscRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
    try:
        request = ShowScanJobsRequest()
        response = client.show_scan_jobs(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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowScanJobsRequest{}
    response, err := client.ShowScanJobs(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	OK
400	无效请求

错误码

请参见**错误码**。

3.7.2 查询指定任务扫描结果

功能介绍

查询指定任务扫描结果

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/sdg/scan/job/{job_id}/results

表 3-92 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
job_id	是	String	任务ID

表 3-93 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小
type	否	String	资产类型

请求参数

无

响应参数

状态码：200

表 3-94 响应 Body 参数

参数	参数类型	描述
job_id	String	任务ID
job_name	String	任务名
type	String	查询资产类型
db_scan_result	**DbScanResult** object	数据库扫描结果
obs_scan_result	**ObsScanResult** object	OBS扫描结果
es_scan_result	**EsScanResult** object	ES扫描结果

表 3-95 DbScanResult

参数	参数类型	描述
total	Integer	扫描结果总数
db_scan_results	Array of **DbScanResultInfo** objects	数据库扫描结果列表

表 3-96 DbScanResultInfo

参数	参数类型	描述
task_id	String	任务ID
db_name	String	数据库名称
table_id	String	表ID
table_name	String	表名称
risk_level	Integer	风险等级
sensitive_data_type	Array of strings	匹配到的规则
match_info	Array of **DbMatchInfo** objects	表中各列匹配到的规则

表 3-97 DbMatchInfo

参数	参数类型	描述
column_name	String	列名
rule_name	String	匹配的规则名
rule_id	String	匹配的规则ID
rule_risk_level	Integer	匹配规则风险等级
column_line	Array of longs	风险数据行

表 3-98 ObsScanResult

参数	参数类型	描述
total	Integer	扫描结果总数
db_scan_results	Array of **ObsScanResultInfo** objects	OBS扫描结果列表

表 3-99 ObsScanResultInfo

参数	参数类型	描述
task_id	String	任务ID
bucket_id	String	OBS桶ID
bucket_name	String	OBS桶名称
file_path	String	文件路径
file_name	String	文件名
md5	String	文件md5值
risk_level	Integer	风险等级
sensitive_data_type	Array of strings	风险数据类型

表 3-100 EsScanResult

参数	参数类型	描述
total	Integer	扫描结果总数

参数	参数类型	描述
db_scan_results	Array of **EsScanResultInfo** objects	ES扫描结果列表

表 3-101 EsScanResultInfo

参数	参数类型	描述
task_id	String	任务ID
index_name	String	索引名
type_id	String	类型ID
type_name	String	类型名
risk_level	Integer	风险等级
sensitive_data_type	Array of strings	敏感数据类型
match_info	Array of **EsMatchInfo** objects	规则匹配具体信息

表 3-102 EsMatchInfo

参数	参数类型	描述
field_name	String	数据字段名
rule_name	String	规则名
rule_id	String	规则ID
rule_risk_level	Integer	规则风险等级

状态码：400

表 3-103 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询指定任务扫描结果

```
GET /v1/{project_id}/sdg/scan/job/{job_id}/results
```

响应示例

状态码：200

OK

```
{
  "job_id" : "xxxxxx",
  "job_name" : "xxxxxx",
  "type" : "DATABASE",
  "db_scan_result" : {
    "total" : 1,
    "db_scan_results" : [ {
      "task_id" : "xxxxxx",
      "db_name" : "xxxxxx",
      "table_id" : "xxxxxx",
      "table_name" : "student",
      "risk_level" : 6,
      "sensitive_data_type" : [ "xxxxxx", "xxxxxx" ],
      "match_info" : [ {
        "column_name" : "phone",
        "rule_name" : "xxxxxx",
        "rule_id" : "xxxxxx",
        "rule_risk_level" : 6,
        "column_line" : [ 1, 3 ]
      }, {
        "column_name" : "email",
        "rule_name" : "xxxxxx",
        "rule_id" : "xxxxxx",
        "rule_risk_level" : 1,
        "column_line" : [ 1, 3 ]
      } ]
    } ]
  },
  "obs_scan_result" : null,
  "es_scan_result" : null
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowScanJobResultsSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowScanJobResultsRequest request = new ShowScanJobResultsRequest();
        request.withJobId("{job_id}");
        try {
            ShowScanJobResultsResponse response = client.showScanJobResults(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = DscClient.new_builder() \
        .with_credentials(credentials) \
```

```
        .with_region(DscRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowScanJobResultsRequest()
        request.job_id = "{job_id}"
        response = client.show_scan_job_results(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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowScanJobResultsRequest{}
    request.JobId = "{job_id}"
    response, err := client.ShowScanJobResults(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	OK
400	无效请求

错误码

请参见**错误码**。

3.7.3 查看规则列表

功能介绍

查询扫描规则列表，返回扫描规则总数和扫描规则列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/sdg/server/scan/rules

表 3-104 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 3-105 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小

请求参数

无

响应参数

状态码：200

表 3-106 响应 Body 参数

参数	参数类型	描述
rules	Array of **ResponseRule** objects	规则列表
total	Integer	规则总数

表 3-107 ResponseRule

参数	参数类型	描述
category	String	规则类别，内置规则(BUILT_IN)或自建规则(BUILT_SELF)
delete_allowed	Boolean	是否允许删除
group_names	String	相关的规则组
id	String	规则ID
logic_operator	String	逻辑运算符，"AND","OR","REGEX"
min_match	Integer	最小匹配次数
risk_level	Integer	风险等级
rule_content	String	规则内容
rule_desc	String	规则描述
rule_name	String	规则名称
rule_type	String	规则类型，关键字(KEYWORD)、正则表达式(REGEX)或自然语言(NLP)
selected	Boolean	是否可选

状态码：400

表 3-108 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询扫描规则列表

```
GET /v1/{project_id}/sdg/server/scan/rules
```

响应示例

状态码：200

请求成功

```
{
  "total" : 1,
  "rules" : [ {
    "category" : "BUILT_SELF",
    "delete_allowed" : true,
    "group_names" : "xxxx",
    "id" : "xxxxxxxxxxxx",
    "logic_operator" : "AND",
    "min_match" : 1,
    "risk_level" : 1,
    "rule_content" : "xxxx",
    "rule_desc" : "xxxx",
    "rule_name" : "xxxx",
    "rule_type" : "KEYWORD",
    "selected" : true
  } ]
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowRulesSolution {

    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");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
```

```
            .withAk(ak)
            .withSk(sk);

    DscClient client = DscClient.newBuilder()
            .withCredential(auth)
            .withRegion(DscRegion.valueOf("<YOUR REGION>"))
            .build();
    ShowRulesRequest request = new ShowRulesRequest();
    try {
        ShowRulesResponse response = client.showRules(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowRulesRequest()
        response = client.show_rules(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"
```

```
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowRulesRequest{}
    response, err := client.ShowRules(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.7.4 创建扫描规则

功能介绍

根据指定的规则名称、规则类型、风险等级、最小匹配次数等参数创建自定义的敏感数据识别规则

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/sdg/server/scan/rules

表 3-109 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-110 请求 Body 参数

参数	是否必选	参数类型	描述
category	是	String	规则类别，内置规则(BUILT_IN)或自建规则(BUILT_SELF)
id	否	String	规则ID
logic_operator	是	String	逻辑运算符，"AND","OR","REGEX"
min_match	是	Integer	最小匹配次数
risk_level	是	Integer	风险等级
rule_content	是	String	规则内容
rule_desc	否	String	规则描述
rule_name	是	String	规则名称
rule_type	是	String	规则类型，关键字(KEYWORD)、正则表达式(REGEX)或自然语言(NLP)

响应参数

状态码：200

表 3-111 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息

参数	参数类型	描述
status	String	返回状态，如'200','400'

状态码：400

表 3-112 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

创建规则名称为xxxx的自建规则，逻辑运算符选择AND，其中最小匹配值为1、风险等级为1、规则内容为xxxx、规则描述为xxxx、规则类型为KEYWORD。

```
POST /v1/{project_id}/sdg/server/scan/rules

{
  "category" : "BUILT_SELF",
  "logic_operator" : "AND",
  "min_match" : 1,
  "risk_level" : 1,
  "rule_content" : "xxxx",
  "rule_desc" : "xxxx",
  "rule_name" : "xxxx",
  "rule_type" : "KEYWORD"
}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建规则名称为xxxx的自建规则，逻辑运算符选择AND，其中最小匹配值为1、风险等级为1、规则内容为xxxx、规则描述为xxxx、规则类型为KEYWORD。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class AddRuleSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        AddRuleRequest request = new AddRuleRequest();
        RuleRequest body = new RuleRequest();
        body.withRuleType(RuleRequest.RuleTypeEnum.fromValue("KEYWORD"));
        body.withRuleName("xxxx");
        body.withRuleDesc("xxxx");
        body.withRuleContent("xxxx");
        body.withRiskLevel(1);
        body.withMinMatch(1);
        body.withLogicOperator("AND");
        body.withCategory(RuleRequest.CategoryEnum.fromValue("BUILT_SELF"));
        request.withBody(body);
        try {
            AddRuleResponse response = client.addRule(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

创建规则名称为xxxx的自建规则，逻辑运算符选择AND，其中最小匹配值为1、风险等级为1、规则内容为xxxx、规则描述为xxxx、规则类型为KEYWORD。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddRuleRequest()
        request.body = RuleRequest(
            rule_type="KEYWORD",
            rule_name="xxxx",
            rule_desc="xxxx",
            rule_content="xxxx",
            risk_level=1,
            min_match=1,
            logic_operator="AND",
            category="BUILT_SELF"
        )
        response = client.add_rule(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

创建规则名称为xxxx的自建规则，逻辑运算符选择AND，其中最小匹配值为1、风险等级为1、规则内容为xxxx、规则描述为xxxx、规则类型为KEYWORD。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddRuleRequest{}
    ruleDescRuleRequest:= "xxxx"
    request.Body = &model.RuleRequest{
        RuleType: model.GetRuleRequestRuleTypeEnum().KEYWORD,
        RuleName: "xxxx",
        RuleDesc: &ruleDescRuleRequest,
        RuleContent: "xxxx",
        RiskLevel: int32(1),
        MinMatch: int32(1),
        LogicOperator: "AND",
        Category: model.GetRuleRequestCategoryEnum().BUILT_SELF,
    }
    response, err := client.AddRule(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.7.5 修改扫描规则

功能介绍

修改自定义的敏感数据识别规则

调用方法

请参见**如何调用API**。

URI

PUT /v1/{project_id}/sdg/server/scan/rules

表 3-113 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-114 请求 Body 参数

参数	是否必选	参数类型	描述
category	是	String	规则类别，内置规则(BUILT_IN)或自建规则(BUILT_SELF)
id	是	String	规则ID
logic_operator	是	String	逻辑运算符，"AND","OR","REGEX"
min_match	是	Integer	最小匹配次数
risk_level	是	Integer	风险等级
rule_content	是	String	规则内容
rule_desc	否	String	规则描述
rule_name	是	String	规则名称
rule_type	是	String	规则类型，关键字(KEYWORD)、正则表达式(REGEX)或自然语言(NLP)

响应参数

状态码：200

表 3-115 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息

参数	参数类型	描述
status	String	返回状态，如'200','400'

状态码：400

表 3-116 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

修改规则名称为xxxx的自建规则，逻辑运算符是OR，其中最小匹配值为1、风险等级为1、规则内容为xxxx、规则描述为xxxx、规则类型为xxxx。

```
PUT /v1/{project_id}/sdg/server/scan/rules

{
  "category" : "BUILT_SELF",
  "id" : "xxxxxxxxxxxxxxxxxxx",
  "logic_operator" : "OR",
  "min_match" : 1,
  "risk_level" : 1,
  "rule_content" : "xxxx",
  "rule_desc" : "xxxx",
  "rule_name" : "xxxx",
  "rule_type" : "xxxx"
}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改规则名称为xxxx的自建规则，逻辑运算符是OR，其中最小匹配值为1、风险等级为1、规则内容为xxxx、规则描述为xxxx、规则类型为xxxx。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ChangeRuleSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ChangeRuleRequest request = new ChangeRuleRequest();
        RuleChangeRequest body = new RuleChangeRequest();
        body.withRuleType(RuleChangeRequest.RuleTypeEnum.fromValue("xxxx"));
        body.withRuleName("xxxx");
        body.withRuleDesc("xxxx");
        body.withRuleContent("xxxx");
        body.withRiskLevel(1);
        body.withMinMatch(1);
        body.withLogicOperator("OR");
        body.withId("xxxxxxxxxxxxxxxxxx");
        body.withCategory(RuleChangeRequest.CategoryEnum.fromValue("BUILT_SELF"));
        request.withBody(body);
        try {
            ChangeRuleResponse response = client.changeRule(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

修改规则名称为xxxx的自建规则，逻辑运算符是OR，其中最小匹配值为1、风险等级为1、规则内容为xxxx、规则描述为xxxx、规则类型为xxxx。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ChangeRuleRequest()
        request.body = RuleChangeRequest(
            rule_type="xxxx",
            rule_name="xxxx",
            rule_desc="xxxx",
            rule_content="xxxx",
            risk_level=1,
            min_match=1,
            logic_operator="OR",
            id="xxxxxxxxxxxxxxxxxxx",
            category="BUILT_SELF"
        )
        response = client.change_rule(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

修改规则名称为xxxx的自建规则，逻辑运算符是OR，其中最小匹配值为1、风险等级为1、规则内容为xxxx、规则描述为xxxx、规则类型为xxxx。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeRuleRequest{}
    ruleDescRuleChangeRequest:= "xxxx"
    request.Body = &model.RuleChangeRequest{
        RuleType: model.GetRuleChangeRequestRuleTypeEnum().XXXX,
        RuleName: "xxxx",
        RuleDesc: &ruleDescRuleChangeRequest,
        RuleContent: "xxxx",
        RiskLevel: int32(1),
        MinMatch: int32(1),
        LogicOperator: "OR",
        Id: "xxxxxxxxxxxxxxxxxx",
        Category: model.GetRuleChangeRequestCategoryEnum().BUILT_SELF,
    }
    response, err := client.ChangeRule(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.7.6 删除扫描规则

功能介绍

删除指定的敏感数据识别规则

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/sdg/server/scan/rules/{rule_id}

表 3-117 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
rule_id	是	String	规则ID

请求参数

无

响应参数

状态码：200

表 3-118 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息
status	String	返回状态，如'200','400'

状态码：400

表 3-119 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

删除指定的扫描规则

```
DELETE /v1/{project_id}/sdg/server/scan/rules/{rule_id}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class DeleteRuleSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteRuleRequest request = new DeleteRuleRequest();
```

```
        request.withRuleId("{rule_id}");
        try {
            DeleteRuleResponse response = client.deleteRule(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteRuleRequest()
        request.rule_id = "{rule_id}"
        response = client.delete_rule(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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteRuleRequest{}
    request.RuleId = "{rule_id}"
    response, err := client.DeleteRule(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.7.7 查询扫描规则组列表

功能介绍

根据指定的项目ID查询扫描规则组列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/sdg/server/scan/groups

表 3-120 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 3-121 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小

请求参数

无

响应参数

状态码：200

表 3-122 响应 Body 参数

参数	参数类型	描述
total	Integer	规则组总数
groups	Array of **ResponseGroup** objects	规则组列表

表 3-123 ResponseGroup

参数	参数类型	描述
category	String	规则类别，内置规则(BUILT_IN)或自建规则(BUILT_SELF)
delete_allowed	Boolean	是否允许删除
group_desc	String	规则组描述
group_name	String	规则组名称
id	String	规则组ID

参数	参数类型	描述
rule_names	String	规则名称
task_names	String	扫描任务名称
is_default	String	是否为默认规则组

状态码：400

表 3-124 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询扫描规则组列表

```
GET /v1/{project_id}/sdg/server/scan/groups
```

响应示例

状态码：200

请求成功

```
{
  "total" : 1,
  "groups" : [ {
    "id" : "xxxxxxxxxxxx",
    "group_name" : "xxxx",
    "group_desc" : "xxxx",
    "category" : "private",
    "rule_names" : "xxxx",
    "is_default" : false
  } ]
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ListRuleGroupsSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ListRuleGroupsRequest request = new ListRuleGroupsRequest();
        try {
            ListRuleGroupsResponse response = client.listRuleGroups(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListRuleGroupsRequest()
        response = client.list_rule_groups(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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListRuleGroupsRequest{}
    response, err := client.ListRuleGroups(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.7.8 创建扫描规则组

功能介绍

根据指定的规则组名称和扫描规则列表创建敏感数据扫描规则组

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/sdg/server/scan/groups

表 3-125 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-126 请求 Body 参数

参数	是否必选	参数类型	描述
category	否	String	规则类别，内置规则(BUILT_IN)或自建规则(BUILT_SELF)
default_status	否	Boolean	是否默认规则组
group_desc	否	String	规则组描述
group_name	否	String	规则组名称
id	否	String	规则组ID

参数	是否必选	参数类型	描述
rule_ids	否	Array of strings	包含的规则ID列表

响应参数

状态码：200

表 3-127 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息
status	String	返回状态，如'200','400'

状态码：400

表 3-128 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

创建规则组名称为xxxx的自建扫描规则组。

```
POST /v1/{project_id}/sdg/server/scan/groups

{
  "category" : "BUILT_SELF",
  "group_desc" : "xxxx",
  "group_name" : "xxxx",
  "rule_ids" : [ "xxxxxxxxxxxxxxxxxx", "xxxxxxxxxxxxxxxxxx" ]
}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建规则组名称为xxxx的自建扫描规则组。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;

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

public class AddRuleGroupSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        AddRuleGroupRequest request = new AddRuleGroupRequest();
        RuleGroupRequest body = new RuleGroupRequest();
        List<String> listbodyRuleIds = new ArrayList<>();
        listbodyRuleIds.add("xxxxxxxxxxxxxxxxxxx");
        listbodyRuleIds.add("xxxxxxxxxxxxxxxxxxx");
        body.withRuleIds(listbodyRuleIds);
        body.withGroupName("xxxx");
        body.withGroupDesc("xxxx");
        body.withCategory(RuleGroupRequest.CategoryEnum.fromValue("BUILT_SELF"));
        request.withBody(body);
        try {
            AddRuleGroupResponse response = client.addRuleGroup(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

创建规则组名称为xxxx的自建扫描规则组。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddRuleGroupRequest()
        listRuleIdsbody = [
            "xxxxxxxxxxxxxxxxxx",
            "xxxxxxxxxxxxxxxxxx"
        ]
        request.body = RuleGroupRequest(
            rule_ids=listRuleIdsbody,
            group_name="xxxx",
            group_desc="xxxx",
            category="BUILT_SELF"
        )
        response = client.add_rule_group(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

创建规则组名称为xxxx的自建扫描规则组。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddRuleGroupRequest{}
    var listRuleIdsbody = []string{
        "xxxxxxxxxxxxxxxxxx",
        "xxxxxxxxxxxxxxxxxx",
    }
    groupNameRuleGroupRequest:= "xxxx"
    groupDescRuleGroupRequest:= "xxxx"
    categoryRuleGroupRequest:= model.GetRuleGroupRequestCategoryEnum().BUILT_SELF
    request.Body = &model.RuleGroupRequest{
        RuleIds: &listRuleIdsbody,
        GroupName: &groupNameRuleGroupRequest,
        GroupDesc: &groupDescRuleGroupRequest,
        Category: &categoryRuleGroupRequest,
    }
    response, err := client.AddRuleGroup(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.7.9 删除扫描规则组

功能介绍

根据扫描规则组ID删除指定的扫描规则组

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/sdg/server/scan/groups/{group_id}

表 3-129 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
group_id	是	String	规则组ID

请求参数

无

响应参数

状态码：200

表 3-130 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息
status	String	返回状态，如'200','400'

状态码：400

表 3-131 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	错误信息

请求示例

删除指定的扫描规则组

```
DELETE /v1/{project_id}/sdg/server/scan/groups/{group_id}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class DeleteRuleGroupSolution {

    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");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
                .withProjectId(projectId)
```

```
            .withAk(ak)
            .withSk(sk);

    DscClient client = DscClient.newBuilder()
            .withCredential(auth)
            .withRegion(DscRegion.valueOf("<YOUR REGION>"))
            .build();
    DeleteRuleGroupRequest request = new DeleteRuleGroupRequest();
    request.withGroupId("{group_id}");
    try {
        DeleteRuleGroupResponse response = client.deleteRuleGroup(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteRuleGroupRequest()
        request.group_id = "{group_id}"
        response = client.delete_rule_group(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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteRuleGroupRequest{}
    request.GroupId = "{group_id}"
    response, err := client.DeleteRuleGroup(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.7.10 创建扫描任务

功能介绍

根据指定的任务名称、扫描方式、扫描周期、扫描规则组、扫描时间创建扫描任务

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/sdg/scan/job

表 3-132 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-133 请求 Body 参数

参数	是否必选	参数类型	描述
asset_ids	是	Array of strings	资产ID列表
cycle	是	String	扫描周期，日(DAY)，周(WEEK)，月(MONTH)，单次扫描(ONCE)
name	是	String	扫描任务名
open	否	Boolean	是否开启任务
rule_group_ids	是	Array of strings	规则组ID列表
start_time	否	Long	扫描任务开始时间
time_zone	否	String	时区
topic_urn	否	String	主题的唯一资源标识符
use_nlp	否	Boolean	是否用nlp

响应参数

状态码：200

表 3-134 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息

参数	参数类型	描述
status	String	返回状态，如'200','400'

状态码：400

表 3-135 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

创建任务名为xxxx的扫描任务，选择xxxxxxxxxxxx规则组，单次扫描，立即执行。

```
POST /v1/{project_id}/sdg/scan/job

{
  "asset_ids" : [ "xxxx", "xxxx" ],
  "cycle" : "ONCE",
  "name" : "xxxx",
  "open" : true,
  "rule_group_ids" : [ "xxxx", "xxxx" ],
  "start_time" : 0,
  "time_zone" : 8,
  "topic_urn" : "xxxxxxxxxxx",
  "use_nlp" : false
}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建任务名为xxxx的扫描任务，选择xxxxxxxxxxxx规则组，单次扫描，立即执行。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;

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

public class AddScanJobSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        AddScanJobRequest request = new AddScanJobRequest();
        ScanJobRequest body = new ScanJobRequest();
        List<String> listbodyRuleGroupIds = new ArrayList<>();
        listbodyRuleGroupIds.add("xxxx");
        listbodyRuleGroupIds.add("xxxx");
        List<String> listbodyAssetIds = new ArrayList<>();
        listbodyAssetIds.add("xxxx");
        listbodyAssetIds.add("xxxx");
        body.withUseNlp(false);
        body.withTopicUrn("xxxxxxxxxxxx");
        body.withTimeZone("8");
        body.withStartTime(0L);
        body.withRuleGroupIds(listbodyRuleGroupIds);
        body.withOpen(true);
        body.withName("xxxx");
        body.withCycle(ScanJobRequest.CycleEnum.fromValue("ONCE"));
        body.withAssetIds(listbodyAssetIds);
        request.withBody(body);
        try {
            AddScanJobResponse response = client.addScanJob(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

创建任务名为xxxx的扫描任务，选择xxxxxxxxxxxx规则组，单次扫描，立即执行。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = AddScanJobRequest()
        listRuleGroupIdsbody = [
            "xxxx",
            "xxxx"
        ]
        listAssetIdsbody = [
            "xxxx",
            "xxxx"
        ]
        request.body = ScanJobRequest(
            use_nlp=False,
            topic_urn="xxxxxxxxxxx",
            time_zone="8",
            start_time=0,
            rule_group_ids=listRuleGroupIdsbody,
            open=True,
            name="xxxx",
            cycle="ONCE",
            asset_ids=listAssetIdsbody
        )
        response = client.add_scan_job(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

创建任务名为xxxx的扫描任务，选择xxxxxxxxxxxx规则组，单次扫描，立即执行。

```
package main
```

```
import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AddScanJobRequest{}
    var listRuleGroupIdsbody = []string{
        "xxxx",
        "xxxx",
    }
    var listAssetIdsbody = []string{
        "xxxx",
        "xxxx",
    }
    useNlpScanJobRequest:= false
    topicUrnScanJobRequest:= "xxxxxxxxxxx"
    timeZoneScanJobRequest:= "8"
    startTimeScanJobRequest:= int64(0)
    openScanJobRequest:= true
    request.Body = &model.ScanJobRequest{
        UseNlp: &useNlpScanJobRequest,
        TopicUrn: &topicUrnScanJobRequest,
        TimeZone: &timeZoneScanJobRequest,
        StartTime: &startTimeScanJobRequest,
        RuleGroupIds: listRuleGroupIdsbody,
        Open: &openScanJobRequest,
        Name: "xxxx",
        Cycle: model.GetScanJobRequestCycleEnum().ONCE,
        AssetIds: listAssetIdsbody,
    }
    response, err := client.AddScanJob(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.7.11 删除扫描任务

功能介绍

删除扫描任务

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/sdg/scan/job/{job_id}

表 3-136 路径参数

参数	是否必选	参数类型	描述
job_id	是	String	任务ID
project_id	是	String	项目ID

请求参数

无

响应参数

状态码：200

表 3-137 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息
status	String	返回状态，如'200','400'

状态码：400

表 3-138 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

删除扫描任务

```
DELETE /v1/{project_id}/sdg/scan/job/{job_id}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class DeleteScanJobSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteScanJobRequest request = new DeleteScanJobRequest();
        request.withJobId("{job_id}");
        try {
            DeleteScanJobResponse response = client.deleteScanJob(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteScanJobRequest()
        request.job_id = "{job_id}"
        response = client.delete_scan_job(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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteScanJobRequest{}
    request.JobId = "{job_id}"
    response, err := client.DeleteScanJob(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.8 文档水印

3.8.1 文档嵌入水印

功能介绍

对WORD(.docx)，PPT(.pptx)，EXCEL(.xlsx)，PDF(.pdf) 类型的文件嵌入文字暗水印、文字明水印或者图片明水印，用户以formData的格式传入待加水印的文件和水印相关信息，DSC服务给文件加完水印后返回给用户已嵌入水印的文件的二进制流。

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/sdg/doc/watermark/embed

表 3-139 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-140 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）

表 3-141 FormData 参数

参数	是否必选	参数类型	描述
doc_type	是	String	要嵌入水印的文档类型
file_password	否	String	输入文件有密码时，读取文件的密码， 最大支持长度256。如果Office文档有读密码或域控的权限密码，请输入读密码，或者有读权限的域控密码。

参数	是否必选	参数类型	描述
marked_file_password	否	String	添加水印后给文件设置密码，最大支持长度256。默认不加文档密码。
readonly_password	否	String	添加水印后给文件设置只读密码， 最大支持长度256。默认不加只读密码。
visible_watermark	否	String	明水印内容，与“blind_watermark”字段至少有一个不为空
font_size	否	String	明水印字体大小，取值为[1,100]，默认值50
rotation	否	String	明水印旋转角度，逆时针方向，取值为[0,90]，默认值45
opacity	否	String	明水印的透明度，取值[0,1]，默认值为0.3；
blind_watermark	否	String	暗水印内容，与“visible_watermark”字段至少有一个不为空
file	是	File	要添加水印的文档
image_mark	否	File	图形水印的字节流。图形文件的格式必须为“png”或“jpg”，否则返回参数错误；图像文件大小不超过1MB；在分段的请求体“Content-Disposition”部分，参数“name”的值必须为“image_mark”。
visible_type	否	String	该字段为空时，默认为**TEXT**类型。 当该字段为**IMAGE**时: ● 请求的表单中必须包含名为“image”的图像文件，图像格式必须为“png”或“jpg”，否则返回参数错误； ● 图像文件大小不超过1MB； ● “visible_watermark”，“font_size”，“rotation”和“opacity”字段无效。

响应参数

状态码：200

表 3-142 响应 Body 参数

参数	参数类型	描述
-	File	

状态码：400

表 3-143 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

说明

通过form表单提交请求，其中file是具体文件。

```
POST /v1/{project_id}/sdg/doc/watermark/embed

{
  "file" : "test.doc",
  "doc_type" : "WORD",
  "opacity" : "0.1",
  "font_size" : "30",
  "rotation" : "45",
  "blind_watermark" : "blind_watermark",
  "visible_watermark" : "visible_watermark"
}
```

响应示例

状态码：200

请求成功

```
"{\"Watermarked document\"}"
```

状态码：400

无效请求

```
{
  "error_code" : "DSC.00000007 ",
  "error_msg" : "File format error"
}
```

SDK 代码示例

SDK代码示例如下。

Java

说明

通过form表单提交请求，其中file是具体文件。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class CreateDocWatermarkSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateDocWatermarkRequest request = new CreateDocWatermarkRequest();
        CreateDocWatermarkRequestBody bodybody = new CreateDocWatermarkRequestBody();
        bodybody.withDocType(CreateDocWatermarkRequestBody.DocTypeEnum.fromValue("WORD"))
            .withVisibleWatermark("visible_watermark")
            .withFontSize("30")
            .withRotation("45")
            .withOpacity("0.1")
            .withBlindWatermark("blind_watermark")
            .withFile("test.doc");
        body.withBody(bodybody);
        request.withBody(listbodyBody);
        try {
            CreateDocWatermarkResponse response = client.createDocWatermark(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

说明

通过form表单提交请求，其中file是具体文件。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateDocWatermarkRequest()
        bodybody = CreateDocWatermarkRequestBody(
            doc_type="WORD",
            visible_watermark="visible_watermark",
            font_size="30",
            rotation="45",
            opacity="0.1",
            blind_watermark="blind_watermark",
            file="test.doc"
        )
        request.body = listBodybody
        response = client.create_doc_watermark(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

说明

通过form表单提交请求，其中file是具体文件。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDocWatermarkRequest{}
    visibleWatermarkBody:= "visible_watermark"
    fontSizeBody:= "30"
    rotationBody:= "45"
    opacityBody:= "0.1"
    blindWatermarkBody:= "blind_watermark"
    bodybody := &model.CreateDocWatermarkRequestBody{
        DocType: model.GetCreateDocWatermarkRequestBodyDocTypeEnum().WORD,
        VisibleWatermark: &visibleWatermarkBody,
        FontSize: &fontSizeBody,
        Rotation: &rotationBody,
        Opacity: &opacityBody,
        BlindWatermark: &blindWatermarkBody,
        File: "test.doc",
    }
    request.Body = listBodybody
    response, err := client.CreateDocWatermark(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.8.2 文档提取暗水印

功能介绍

对已嵌入文字暗水印的WORD(.docx)，PPT(.pptx)，EXCEL(.xlsx)，PDF(.pdf)类型的文档进行文字暗水印提取，用户以formData的格式传入待提取水印的文件，DSC服务以JSON的格式返回从文档里提取的出的文字暗水印内容。

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/sdg/doc/watermark/extract

表 3-144 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-145 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）

表 3-146 FormData 参数

参数	是否必选	参数类型	描述
doc_type	是	String	待提取水印的文档类型
file_password	否	String	解密文件的密码， 最大支持长度256。如果Office文档有读密码或域控的权限密码，请输入读密码，或者有读权限的域控密码。
file	是	File	上传要提取水印的文档

响应参数

状态码：200

表 3-147 响应 Body 参数

参数	参数类型	描述
watermark	String	暗水印内容，长度不超过32个字节

状态码：400

表 3-148 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

提取文档类型为WORD的testMarked.doc文件的暗水印。 > 通过form表单提交请求，其中file为具体文件

```
POST /v1/{project_id}/sdg/doc/watermark/extract

{
  "file" : "testMarked.doc",
  "doc_type" : "WORD"
}
```

响应示例

状态码：200

请求成功

```
{
  "watermark" : "mark!"
}
```

状态码：400

无效请求

```
{
  "error_code" : "DSC.00000007",
  "error_msg" : "File format error"
}
```

SDK 代码示例

SDK代码示例如下。

Java

提取文档类型为WORD的testMarked.doc文件的暗水印。 > 通过form表单提交请求，其中file为具体文件

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowDocWatermarkSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowDocWatermarkRequest request = new ShowDocWatermarkRequest();
        ShowDocWatermarkRequestBody bodybody = new ShowDocWatermarkRequestBody();
        bodybody.withDocType(ShowDocWatermarkRequestBody.DocTypeEnum.fromValue("WORD"))
            .withFile("testMarked.doc");
        body.withBody(bodybody);
        request.withBody(listbodyBody);
        try {
            ShowDocWatermarkResponse response = client.showDocWatermark(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

提取文档类型为WORD的testMarked.doc文件的暗水印。 > 通过form表单提交请求，其中file为具体文件

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowDocWatermarkRequest()
        bodybody = ShowDocWatermarkRequestBody(
            doc_type="WORD",
            file="testMarked.doc"
        )
        request.body = listBodybody
        response = client.show_doc_watermark(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

提取文档类型为WORD的testMarked.doc文件的暗水印。 > 通过form表单提交请求，其中file为具体文件

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

```
    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDocWatermarkRequest{}
    bodybody := &model.ShowDocWatermarkRequestBody{
        DocType: model.GetShowDocWatermarkRequestBodyDocTypeEnum().WORD,
        File: "testMarked.doc",
    }
    request.Body = listBodybody
    response, err := client.ShowDocWatermark(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.8.3 文档嵌入水印（文件地址版本）

功能介绍

对WORD(.docx)，PPT(.pptx)，EXCEL(.xlsx)，PDF(.pdf)*类型的文档嵌入文字暗水印、文字明水印或者图片明水印，用户传入待加水印的文档地址（目前支持OBS)和水印相关信息，DSC服务对文档加完水印后返回给用户已嵌入水印的文档的存放地址。

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/doc-address/watermark/embed

表 3-149 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-150 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）

表 3-151 请求 Body 参数

参数	是否必选	参数类型	描述
region_id	是	String	项目所在region的id，如：xx-xx-1。
src_file	是	String	待添加水印的文档地址，当前只支持OBS对象，格式为 **obs://bucket/object** ，其中bucket为和当前项目处于同一区域的OBS桶名称，object为对象全路径名。例如：**obs://bucket/info/wm.png**，其中obs://表示OBS存储，bucket为桶名，info/wm.png为对象全路径名。
doc_type	是	String	待嵌入水印的文档类型。
dst_file	否	String	添加水印后的文档存放地址，格式和要求同src_file字段，不设置时，默认取src_file的值，即添加水印后覆盖原文件。
blind_watermark	否	String	暗文字水印内容，与“visible_watermark”字段至少有一个不为空
visible_watermark	否	String	明文字水印内容，与暗水印“blind_watermark”字段至少有一个不为空。

参数	是否必选	参数类型	描述
image_mark	否	String	待嵌入的图形明水印文件的地址, 字段格式要求同src_file字段，图形文件的格式必须为“png”或“jpg”，否则返回参数错误；图像文件大小不超过1MB
visible_type	否	String	该字段控制明水印嵌入文字还是图片。默认为**TEXT**类型，需填写visible_watermark字段设置明文字水印； 当该字段为**IMAGE**时，需填写image_watermark字段设置水印图片地址此时，“visible_watermark”，“font_size”，“rotation”和“opacity”字段无效。
file_password	否	String	待加水印文件有密码时，读取文件的密码， 最大支持长度256。如果Office文档有读密码或域控的权限密码，请输入读密码，或者有读权限的域控密码。
marked_file_password	否	String	添加水印后给文件设置密码，最大支持长度256。默认不加文档密码。
readonly_password	否	String	添加水印后给文件设置只读密码， 最大支持长度256。默认不加只读密码。
front	否	Integer	明水印字体大小，取值为[1,100]，默认值50
rotation	否	Integer	明水印旋转角度，逆时针方向，取值为[0,90]，默认值45。
opacity	否	Float	明水印的透明度，取值[0,1]，默认值为0.3；

响应参数

状态码：200

表 3-152 响应 Body 参数

参数	参数类型	描述
region_id	String	当前项目所在region的id，如：xx-xx-1。

参数	参数类型	描述
watermarked_file	String	添加水印后的文档地址，当前只支持OBS对象，格式为 **obs://bucket/object** ，其中bucket为和当前项目处于同一区域的OBS桶名称，object为对象全路径名。例如：**obs://bucket/info/hw.doc**，其中obs://表示OBS存储，hwbucket为桶名，hwinfo/hw.doc为对象全路径名。

状态码：400

表 3-153 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

给obs://bucket/info/wm.doc路径下的WORD文档嵌入blind_watermark的明水印和visible_watermark的暗水印。

```
POST /v1/{project_id}/doc-address/watermark/embed

{
  "region_id" : "xx-xx-1",
  "src_file" : "obs://bucket/info/wm.doc",
  "doc_type" : "WORD",
  "blind_watermark" : "blind_watermark",
  "visible_watermark" : "visible_watermark"
}
```

响应示例

状态码：200

请求成功

```
{
  "region_id" : "xx-xx-1",
  "watermarked_file" : "obs://bucket/info/wm.docx"
}
```

状态码：400

无效请求

```
{
  "error_code" : "DSC.00000007 ",
  "error_msg" : "File format error"
}
```

SDK 代码示例

SDK代码示例如下。

Java

给obs://bucket/info/wm.doc路径下的WORD文档嵌入blind_watermark的明水印和visible_watermark的暗水印。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class CreateDocWatermarkByAddressSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateDocWatermarkByAddressRequest request = new CreateDocWatermarkByAddressRequest();
        CreateDocWatermarkByAddressRequestBody body = new CreateDocWatermarkByAddressRequestBody();
        body.withVisibleWatermark("visible_watermark");
        body.withBlindWatermark("blind_watermark");
        body.withDocType(CreateDocWatermarkByAddressRequestBody.DocTypeEnum.fromValue("WORD"));
        body.withSrcFile("obs://bucket/info/wm.doc");
        body.withRegionId("xx-xx-1");
        request.withBody(body);
        try {
            CreateDocWatermarkByAddressResponse response = client.createDocWatermarkByAddress(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

给obs://bucket/info/wm.doc路径下的WORD文档嵌入blind_watermark的明水印和visible_watermark的暗水印。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateDocWatermarkByAddressRequest()
        request.body = CreateDocWatermarkByAddressRequestBody(
            visible_watermark="visible_watermark",
            blind_watermark="blind_watermark",
            doc_type="WORD",
            src_file="obs://bucket/info/wm.doc",
            region_id="xx-xx-1"
        )
        response = client.create_doc_watermark_by_address(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

给obs://bucket/info/wm.doc路径下的WORD文档嵌入blind_watermark的明水印和visible_watermark的暗水印。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDocWatermarkByAddressRequest{}
    visibleWatermarkCreateDocWatermarkByAddressRequestBody:= "visible_watermark"
    blindWatermarkCreateDocWatermarkByAddressRequestBody:= "blind_watermark"
    request.Body = &model.CreateDocWatermarkByAddressRequestBody{
        VisibleWatermark: &visibleWatermarkCreateDocWatermarkByAddressRequestBody,
        BlindWatermark: &blindWatermarkCreateDocWatermarkByAddressRequestBody,
        DocType: model.GetCreateDocWatermarkByAddressRequestBodyDocTypeEnum().WORD,
        SrcFile: "obs://bucket/info/wm.doc",
        RegionId: "xx-xx-1",
    }
    response, err := client.CreateDocWatermarkByAddress(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.8.4 文档提取暗水印（文档地址版本）

功能介绍

支持对已嵌入文字暗水印的WORD(.docx)，PPT(.pptx)，EXCEL(.xlsx)，PDF(.pdf)类型的文档进行水印提取，用户传入待提取水印的文档地址（目前支持OBS），DSC服务以JSON的格式返回从文档里提取的出的文字暗水印内容。

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/doc-address/watermark/extract

表 3-154 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

请求参数

表 3-155 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）

表 3-156 请求 Body 参数

参数	是否必选	参数类型	描述
region_id	是	String	项目所在region的id，如：xx-xx-1。
doc_type	是	String	待提取水印的文档类型
src_file	是	String	待提取文字暗水印的文档的地址，当前只支持OBS对象，格式为 **obs://bucket/object** ，其中bucket为和当前项目处于同一区域的OBS桶名称，object为对象全路径名。例如：**obs://hwbucket/hwinfo/hw.doc**，其中obs://表示OBS存储，hwbucket为桶名，hwinfo/hw.doc为对象全路径名。
file_password	否	String	解密文件的密码， 最大支持长度256。如果Office文档有读密码或域控的权限密码，请输入读密码，或者有读权限的域控密码。

响应参数

状态码：200

表 3-157 响应 Body 参数

参数	参数类型	描述
watermark	String	暗水印内容，长度不超过32个字节

状态码：400

表 3-158 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

提取obs://bucket/info/wm.docx路径下的WORD文档中的暗水印。

```
POST /v1/{project_id}/doc-address/watermark/extract

{
  "region_id" : "xx-xx-1",
  "src_file" : "obs://bucket/info/wm.docx",
  "doc_type" : "WORD"
}
```

响应示例

状态码：200

请求成功

```
{
  "watermark" : "blind_watermark"
}
```

状态码：400

无效请求

```
{
  "error_code" : "DSC.00000007",
  "error_msg" : "File format error"
}
```

SDK 代码示例

SDK代码示例如下。

Java

提取obs://bucket/info/wm.docx路径下的WORD文档中的暗水印。

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowDocWatermarkByAddressSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowDocWatermarkByAddressRequest request = new ShowDocWatermarkByAddressRequest();
        ShowDocWatermarkByAddressRequestBody body = new ShowDocWatermarkByAddressRequestBody();
        body.withSrcFile("obs://bucket/info/wm.docx");
        body.withDocType(ShowDocWatermarkByAddressRequestBody.DocTypeEnum.fromValue("WORD"));
        body.withRegionId("xx-xx-1");
        request.withBody(body);
        try {
            ShowDocWatermarkByAddressResponse response = client.showDocWatermarkByAddress(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

提取obs://bucket/info/wm.docx路径下的WORD文档中的暗水印。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowDocWatermarkByAddressRequest()
        request.body = ShowDocWatermarkByAddressRequestBody(
            src_file="obs://bucket/info/wm.docx",
            doc_type="WORD",
            region_id="xx-xx-1"
        )
        response = client.show_doc_watermark_by_address(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

提取obs://bucket/info/wm.docx路径下的WORD文档中的暗水印。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
        WithCredential(auth).
        Build())

    request := &model.ShowDocWatermarkByAddressRequest{}
    request.Body = &model.ShowDocWatermarkByAddressRequestBody{
        SrcFile: "obs://bucket/info/wm.docx",
        DocType: model.GetShowDocWatermarkByAddressRequestBodyDocTypeEnum().WORD,
        RegionId: "xx-xx-1",
    }
    response, err := client.ShowDocWatermarkByAddress(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.9 数据静态脱敏

3.9.1 查询脱敏任务执行列表

功能介绍

查询脱敏任务执行列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/sdg/server/mask/dbs/templates/{template_id}/tasks

表 3-159 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
template_id	是	String	模板ID

表 3-160 Query 参数

参数	是否必选	参数类型	描述
workspace_id	否	String	工作区ID
offset	否	Integer	页码
limit	否	Integer	分页大小

请求参数

无

响应参数

状态码：200

表 3-161 响应 Body 参数

参数	参数类型	描述
tasks	Array of **DBMaskTaskInfo** objects	脱敏任务列表
total	Integer	脱敏任务总数

表 3-162 DBMaskTaskInfo

参数	参数类型	描述
db_type	String	DB类型
end_time	Long	任务结束时间
execute_line	Integer	执行行数
id	String	任务ID
progress	Integer	执行进度
run_status	String	任务运行状态

参数	参数类型	描述
start_time	Long	任务开始时间
task_template_id	String	任务模板ID
type	String	任务类型

状态码：400

表 3-163 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询脱敏任务执行列表

```
GET /v1/{project_id}/sdg/server/mask/dbs/templates/{template_id}/tasks
```

响应示例

状态码：200

OK

```
{
  "tasks" : [ {
    "db_type" : "MySQL",
    "end_time" : 1658717568622,
    "execute_line" : 100000,
    "id" : "xxxxxxxxxxx",
    "progress" : 100,
    "run_status" : "FINISHED",
    "start_time" : 1658717544469,
    "task_template_id" : "xxxxxxxxxxx",
    "type" : "MANUAL"
  } ],
  "total" : 1
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ListDbMaskTaskSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ListDbMaskTaskRequest request = new ListDbMaskTaskRequest();
        request.withTemplateId("{template_id}");
        try {
            ListDbMaskTaskResponse response = client.listDbMaskTask(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListDbMaskTaskRequest()
        request.template_id = "{template_id}"
        response = client.list_db_mask_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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDbMaskTaskRequest{}
    request.TemplateId = "{template_id}"
    response, err := client.ListDbMaskTask(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	OK
400	无效请求

错误码

请参见**错误码**。

3.9.2 开启/停止脱敏任务

功能介绍

开启/停止脱敏任务

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/sdg/server/mask/dbs/templates/{template_id}/operation

表 3-164 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
template_id	是	String	模板ID

请求参数

表 3-165 请求 Body 参数

参数	是否必选	参数类型	描述
status	是	Integer	脱敏任务状态

响应参数

状态码：200

表 3-166 响应 Body 参数

参数	参数类型	描述
msg	String	返回消息
status	String	返回状态，如'200','400'

状态码：400

表 3-167 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

开启脱敏任务

```
POST /v1/{project_id}/sdg/server/mask/dbs/templates/{template_id}/operation

{
  "status" : 1
}
```

响应示例

状态码：200

请求成功

```
{
  "msg" : "xxxx",
  "status" : "RESPONSE_SUCCESS"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

SDK 代码示例

SDK代码示例如下。

Java

开启脱敏任务

```
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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ChangeDbTemplateOperationSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ChangeDbTemplateOperationRequest request = new ChangeDbTemplateOperationRequest();
        request.withTemplateId("{template_id}");
        MaskSwitchRequest body = new MaskSwitchRequest();
        body.withStatus(1);
        request.withBody(body);
        try {
            ChangeDbTemplateOperationResponse response = client.changeDbTemplateOperation(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

开启脱敏任务

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
```

```
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ChangeDbTemplateOperationRequest()
        request.template_id = "{template_id}"
        request.body = MaskSwitchRequest(
            status=1
        )
        response = client.change_db_template_operation(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

开启脱敏任务

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
    request := &model.ChangeDbTemplateOperationRequest{}
    request.TemplateId = "{template_id}"
    request.Body = &model.MaskSwitchRequest{
        Status: int32(1),
    }
    response, err := client.ChangeDbTemplateOperation(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	请求成功
400	无效请求

错误码

请参见**错误码**。

3.10 API 调用记录

3.10.1 查询 OpenApi 调用记录

功能介绍

查询OpenApi调用记录

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/openapi/called-records

表 3-168 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID。

表 3-169 Query 参数

参数	是否必选	参数类型	描述
limit	否	Integer	分页大小，默认1000，最大2000。
called_url	否	String	需要查询调用记录的URL，例如：/v1/{project_id}/sdg/database/watermark/embed
start_time	否	Long	开始时间（Unix timestamp），单位：毫秒，例如：0
end_time	否	Long	结束时间（Unix timestamp），单位：毫秒，例如：1638515803572
marker	否	String	指定一个标识符。获取第一页时不用赋值，获取下一页时取上页查询结果的返回值。

请求参数

无

响应参数

状态码：200

表 3-170 响应 Body 参数

参数	参数类型	描述
total	Integer	调用API总次数
succeed	Integer	调用API成功次数
failed	Integer	调用API失败次数
openapi_called_records	Array of **OpenApiCalledRecord** objects	API调用记录列表
next_marker	String	获取下一页所需的标识符。

表 3-171 OpenApiCalledRecord

参数	参数类型	描述
user_name	String	调用API的user_name
user_id	String	调用API的user_id
domain_name	String	调用API的domain_name
domain_id	String	调用API的domain_id
request_url	String	调用API的URL
request_method	String	http请求方法
response_code	String	http状态码
fail_reason	String	调用API失败原因
timestamp	Long	调用API的时间（Unix timestamp），单位：毫秒

状态码：400

表 3-172 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询OpenAPI调用记录

```
GET /v1/{project_id}/openapi/called-records
```

响应示例

状态码：200

OK

```
{
  "total" : 11,
  "succeed" : 11,
  "failed" : 0,
  "openapi_called_records" : [ {
    "user_name" : "xxxxxxxxxxxx",
    "user_id" : "xxxxxxxxxxxxxxxx",
    "domain_name" : "xxxxxxxxxxxxxxxx",
    "domain_id" : "xxxxxxxxxxxxxxxxxx",
    "request_url" : "/v1/xxxxxxxxxxxxxxxxxxxxxxxxxxxx/sdg/doc/watermark/embed",
    "request_method" : "POST",
    "response_code" : "200",
```

```
    "fail_reason" : "",
    "timestamp" : 1638427576524
  } ],
  "next_marker" : "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

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.dsc.v1.region.DscRegion;
import com.huaweicloud.sdk.dsc.v1.*;
import com.huaweicloud.sdk.dsc.v1.model.*;


public class ShowOpenApiCalledRecordsSolution {

    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");
        String projectId = "{project_id}";

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

        DscClient client = DscClient.newBuilder()
                .withCredential(auth)
                .withRegion(DscRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowOpenApiCalledRecordsRequest request = new ShowOpenApiCalledRecordsRequest();
        try {
            ShowOpenApiCalledRecordsResponse response = client.showOpenApiCalledRecords(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 huaweicloudsdkdsc.v1.region.dsc_region import DscRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkdsc.v1 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"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowOpenApiCalledRecordsRequest()
        response = client.show_open_api_called_records(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"
    dsc "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/dsc/v1/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")
    projectId := "{project_id}"

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

```
    client := dsc.NewDscClient(
        dsc.DscClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowOpenApiCalledRecordsRequest{}
    response, err := client.ShowOpenApiCalledRecords(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	OK
400	无效请求

错误码

请参见**错误码**。

3.11 敏感数据识别

3.11.1 查询敏感数据识别任务列表

功能介绍

查询敏感数据识别任务列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/sdg/scan/job

表 3-173 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 3-174 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小
content	否	String	任务名称
is_new	否	Boolean	是否新版分级分类

请求参数

无

响应参数

状态码：200

表 3-175 响应 Body 参数

参数	参数类型	描述
tasks	Array of **ScanJob** objects	本次返回的扫描任务列表
total	Long	任务总数

表 3-176 ScanJob

参数	参数类型	描述
id	String	任务ID
name	String	任务名称
rule_groups	Array of strings	任务使用的规则组
scan_templates	Map<String,String>	任务使用的模板
cycle	String	任务执行方式
status	String	任务当前状态

参数	参数类型	描述
last_run_time	Long	任务上一次执行时间
create_time	Long	任务创建时间
last_scan_risk	String	任务上一次扫描风险等级结果
use_nlp	Boolean	是否使用了NLP进行扫描
open	Boolean	任务开启状态
topic_urn	String	SMN服务通知主题
start_time	Long	任务启动时间
security_level_name	String	识别结果风险等级名称
security_level_color	Long	识别结果风险等级
asset_infos	Array of **AssetInfo** objects	资产列表

表 3-177 AssetInfo

参数	参数类型	描述
asset_id	String	资产ID
asset_type	String	资产类型

状态码：400

表 3-178 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询敏感数据识别任务列表

```
GET /v1/{project_id}/sdg/scan/job

{
  "limit" : 10,
  "offset" : 0,
  "content" : "xxxxxxxxxxx",
```

```
  "is_new" : true
}
```

响应示例

状态码：200

请求成功

```
{
  "total" : 1,
  "tasks" : [ {
    "id" : "xxxxxxxxxxxx",
    "name" : "xxxx",
    "cycle" : "xxxx",
    "status" : "FINISHED",
    "open" : true,
    "rule_groups" : [ "xxx" ],
    "scan_templates" : {
      "xxx" : "xxx"
    },
    "last_run_time" : 1712904721162,
    "create_time" : 1712022723454,
    "last_scan_risk" : "HIGH",
    "use_nlp" : false,
    "topic_urn" : "",
    "start_time" : 0,
    "security_level_name" : "L4",
    "security_level_color" : 9,
    "asset_infos" : [ {
      "asset_id" : "xxx",
      "asset_type" : "obs"
    } ]
  } ]
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

状态码

状态码	描述
200	请求成功
400	无效请求

错误码

请参见**错误码**。

3.11.2 查询敏感数据识别资产子任务列表

功能介绍

查询敏感数据识别资产子任务列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/sdg/scan/job/{job_id}/task

表 3-179 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
job_id	是	String	敏感数据识别任务ID

表 3-180 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小
workspace_id	否	String	工作空间ID

请求参数

无

响应参数

状态码：200

表 3-181 响应 Body 参数

参数	参数类型	描述
total	Integer	子任务总数
tasks	Array of **ScanTask** objects	子任务列表

表 3-182 ScanTask

参数	参数类型	描述
id	String	任务ID
category	String	资产类型
status	String	任务状态
progress	Integer	任务进度
asset_name	String	资产名称
asset_id	String	资产ID
start_time	Long	任务开始时间
end_time	Long	任务结束时间
scanned_object_num	Integer	已识别对象数
to_be_scanned_object_num	Integer	待识别对象数
scan_speed	Integer	扫描速率
skip_object_num	Integer	跳过对象数
last_scan_risk	String	上次识别结果
security_level_name	String	识别结果
security_level_color	Integer	识别结果等级

状态码：400

表 3-183 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询敏感数据识别资产子任务列表

```
GET /v1/{project_id}/sdg/scan/job/{job_id}/task
```

响应示例

状态码：200

请求成功

```
{
  "tasks" : [ {
    "id" : "xxxxxxxxxxxx",
    "category" : "OBS",
    "status" : "FINISHED",
    "progress" : 0,
    "asset_name" : "xxxx",
    "asset_id" : "xxxxxxxxxxxx",
    "start_time" : 1711960461182,
    "end_time" : 1711961941073,
    "scanned_object_num" : 4,
    "to_be_scanned_object_num" : 0,
    "scan_speed" : 1,
    "skip_object_num" : 15,
    "last_scan_risk" : "HIGH",
    "security_level_name" : "L4",
    "security_level_color" : 9
  } ],
  "total" : 1
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

状态码

状态码	描述
200	请求成功
400	无效请求

错误码

请参见**错误码**。

3.11.3 获取敏感数据识别分级列表

功能介绍

获取敏感数据识别分级列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/scan-security-levels

表 3-184 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 3-185 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小
name	否	String	级别名称
category	否	String	级别来源
is_deleted	否	Boolean	true：查询所有分级；false：查询没被禁用的分级

请求参数

无

响应参数

状态码：200

表 3-186 响应 Body 参数

参数	参数类型	描述
total	Integer	规则总数
security_levels_list	Array of **SecurityLevel** objects	规则列表

表 3-187 SecurityLevel

参数	参数类型	描述
level_id	String	任务ID
project_id	String	项目ID

参数	参数类型	描述
security_level_name	String	识别结果
color_number	Integer	识别结果等级
security_level_desc	String	识别结果等级描述
used_count	Integer	关联的识别模板个数
category	String	规则创建类型
create_time	Long	规则创建时间
update_time	Long	规则更新时间
sort_weight	Integer	规则排序位置
is_deleted	Boolean	规则是否禁用

状态码：400

表 3-188 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

获取敏感数据识别分级列表

```
GET /v1/{project_id}/scan-security-levels
```

响应示例

状态码：200

请求成功

```
{
  "security_levels_list" : [ {
    "level_id" : "xxxxxxxxxxxx",
    "project_id" : "xxxxxxxxxxxx",
    "security_level_name" : "L4",
    "color_number" : 4,
    "security_level_desc" : "xxxxx",
    "used_count" : 4,
    "category" : "BUILT_SELF",
    "create_time" : 1710560820120,
    "update_time" : 1710573345350,
    "sort_weight" : 21,
```

```
    "is_deleted" : false
  } ],
  "total" : 1
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

状态码

状态码	描述
200	请求成功
400	无效请求

错误码

请参见**错误码**。

3.11.4 获取敏感数据的安全等级列表

功能介绍

获取敏感数据的安全等级列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/scan-jobs/{job_id}/security-levels

表 3-189 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
job_id	是	String	敏感数据识别任务ID

表 3-190 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小
keyword	否	String	敏感信息对象名称关键字
asset_type	否	String	资源类型
asset_id	否	String	资产ID
security_level_ids	否	Array of strings	风险等级ID数组
marker	否	Array of strings	上一批/下一批信息

请求参数

无

响应参数

状态码：200

表 3-191 响应 Body 参数

参数	参数类型	描述
total	Integer	统计风险信息总数
security_level_list	Array of **ClassificationSecurityLevelInfo** objects	统计风险信息列表

表 3-192 ClassificationSecurityLevelInfo

参数	参数类型	描述
security_level_id	String	风险等级ID
security_level_name	String	风险等级名称
security_level_color	Integer	风险等级颜色
count	Integer	风险对象个数

参数	参数类型	描述
percent	String	风险等级在统计信息中的百分比

状态码：400

表 3-193 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

获取敏感数据的安全等级列表

```
GET /v1/{project_id}/scan-jobs/{job_id}/security-levels
```

响应示例

状态码：200

请求成功

```
{
  "security_level_list" : [ {
    "security_level_id" : "xxxxxxxxxxxx",
    "security_level_name" : "L4",
    "security_level_color" : 6,
    "count" : 2,
    "percent" : "0.2000"
  } ],
  "total" : 1
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

状态码

状态码	描述
200	请求成功
400	无效请求

错误码

请参见**错误码**。

3.11.5 获取敏感数据识别结果 Top10 命中规则

功能介绍

获取敏感数据识别结果Top10命中规则

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/scan-jobs/{job_id}/hit-rules

表 3-194 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
job_id	是	String	敏感数据识别任务ID

表 3-195 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小
keyword	否	String	敏感信息对象名称关键字
asset_type	否	String	资源类型
asset_id	否	String	资产ID
security_level_ids	否	Array of strings	风险等级ID数组
marker	否	Array of strings	上一批/下一批信息

请求参数

无

响应参数

状态码：200

表 3-196 响应 Body 参数

参数	参数类型	描述
total	Integer	统计风险信息总数
hit_rule_list	Array of **ClassificationHitInfo** objects	统计风险信息列表

表 3-197 ClassificationHitInfo

参数	参数类型	描述
rule_id	String	规则ID
rule_name	String	规则名称
count	Integer	关联模板个数
top_objects	Array of strings	命中对象列表

状态码：400

表 3-198 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

获取敏感数据识别结果Top10命中规则

```
GET /v1/{project_id}/scan-jobs/{job_id}/hit-rules
```

响应示例

状态码：200

请求成功

```
{
  "hit_rule_list" : [ {
    "rule_id" : "xxxxxxxxxxx",
    "rule_name" : "xxxxxxxxxxx",
```

```
    "count" : 5,
    "top_objects" : [ "xxxx", "xxxxxxx" ]
  } ],
  "total" : 1
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

状态码

状态码	描述
200	请求成功
400	无效请求

错误码

请参见**错误码**。

3.11.6 获取敏感数据分类结果列表

功能介绍

获取敏感数据分类结果列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/scan-jobs/{job_id}/classification-results

表 3-199 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
job_id	是	String	敏感数据识别任务ID

表 3-200 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小
keyword	否	String	敏感信息对象名称关键字
asset_type	否	String	资源类型
asset_id	否	String	资产ID
security_level_ids	否	Array of strings	风险等级ID数组
marker	否	Array of strings	上一批/下一批信息

请求参数

无

响应参数

状态码：200

表 3-201 响应 Body 参数

参数	参数类型	描述
total	Integer	识别结果总数
classification_list	Array of **ClassificationResultsEntity** objects	识别结果对象列表
marker	Array of objects	分批结果

表 3-202 ClassificationResultsEntity

参数	参数类型	描述
id	String	结果标记
project_id	String	项目ID
job_id	String	子任务ID
task_id	String	任务ID
ins_id	String	实例ID

参数	参数类型	描述
asset_id	String	资产ID
asset_name	String	资产名称
asset_type	String	资产类型
object_name	String	对象名称
object_full_path	String	对象全路径
security_level_name	String	识别结果
security_level_id	String	识别结果等级ID
security_level_color	Integer	识别结果等级
create_time	Long	创建时间
update_time	Long	更新时间
scan_time	Long	识别个数
match_info	Array of **ClassificationMatchInfo** objects	匹配规则列表

表 3-203 ClassificationMatchInfo

参数	参数类型	描述
template_id	String	识别模板ID
template_name	String	识别模板名称
rule_id	String	规则ID
rule_name	String	规则名称
security_level_name	String	识别结果等级名称
security_level_id	String	识别结果等级ID
security_level_color	Integer	识别结果等级
classification_name	String	分类分级名称
classification_id	String	分类分级ID
matched_detail	String	匹配明细

参数	参数类型	描述
matched_examples	Array of MatchedExamples objects	匹配规则列表

表 3-204 MatchedExamples

参数	参数类型	描述
line_number	Long	行数
matched_content	String	规则内容

状态码：400

表 3-205 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

获取敏感数据识别结果列表

```
GET /v1/{project_id}/scan-jobs/{job_id}/classification-results
```

响应示例

状态码：200

请求成功

```
{
  "classification_list" : [ {
    "id" : "xxxxxxxxxxx",
    "project_id" : "xxxxxxxxxxx",
    "job_id" : "xxxxxxxxxxx",
    "task_id" : "xxxxxxxxxxx",
    "ins_id" : "xxxxxxxxxxx",
    "asset_id" : "xxxxxxxxxxx",
    "asset_name" : "xxxxxxxxxxx",
    "asset_type" : "OBS",
    "object_name" : "xxxxxxxxxxx",
    "object_full_path" : "xxxxxxxxxxx",
    "security_level_name" : "L4",
    "security_level_id" : "xxxxxxxxxxx",
    "security_level_color" : 4,
    "create_time" : 1710560820120,
    "update_time" : 1710573345350,
    "scan_time" : 12293,
```

```
  "match_info" : [ {
    "template_id" : "xxxxxxxxxxx",
    "template_name" : "xxxxxxxxxxx",
    "rule_id" : "xxxxxxxxxxx",
    "rule_name" : "xxxxx",
    "security_level_name" : "L4",
    "security_level_id" : "xxxxxxxxxxx",
    "security_level_color" : 5,
    "classification_name" : "xxxxxxxxxxx",
    "classification_id" : "xxxxxxxxxxx",
    "matched_detail" : "xxx"
   } ]
  } ],
  "total" : 1,
  "marker" : [ { }, { } ]
}
```


状态码：400

无效请求


```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```


状态码

状态码	描述
200	请求成功
400	无效请求

错误码

请参见**错误码**。

3.11.7 查询敏感数据识别分级分类模板列表

功能介绍

查询敏感数据识别分级分类模板列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/scan-templates

表 3-206 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID

表 3-207 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小

请求参数

无

响应参数

状态码：200

表 3-208 响应 Body 参数

参数	参数类型	描述
total	Integer	识别模板总数
scan_templates_list	Array of **ScanTemplateEntity** objects	识别模板列表

表 3-209 ScanTemplateEntity

参数	参数类型	描述
id	String	识别模板ID
project_id	String	项目ID
name	String	模板名称
description	String	模板描述
category	String	模板分类
is_default	Boolean	是否默认模板
create_time	Long	创建时间
update_time	Long	更新时间
is_deleted	Boolean	是否删除

状态码：400

表 3-210 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询敏感数据识别分级分类模板列表

```
GET /v1/{project_id}/scan-templates
```

响应示例

状态码：200

请求成功

```
{
  "scan_templates_list" : [ {
    "id" : "xxxxxxxxxxxx",
    "project_id" : "xxxxxxxxxxxx",
    "name" : "xxxxxxxxxxxx",
    "description" : "xxxxxxxxxxxx",
    "category" : "BUILT_IN",
    "is_default" : true,
    "create_time" : 1710560820120,
    "update_time" : 1710573345350,
    "is_deleted" : false
  } ],
  "total" : 1
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

状态码

状态码	描述
200	请求成功
400	无效请求

错误码

请参见**错误码**。

3.11.8 查询敏感数据识别分级分类树结构信息

功能介绍

查询敏感数据识别分级分类树结构信息

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/scan-templates/{template_id}/classifications

表 3-211 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
template_id	是	String	模板ID

请求参数

无

响应参数

状态码：200

表 3-212 响应 Body 参数

参数	参数类型	描述
template_id	String	识别模板ID
template_category	String	识别模板分类
template_name	String	识别模板名称
classification_trees	Array of **TemplateClassificationTree** objects	分级分类属性结构

表 3-213 TemplateClassificationTree

参数	参数类型	描述
id	String	分级分类ID

参数	参数类型	描述
project_id	String	项目ID
name	String	分级分类名称
root_id	String	分级分类根ID
parent_id	String	分级分类上级ID
depth	Integer	深度
children_nums	Integer	子分级分类总数
create_time	Long	创建时间
update_time	Long	更新时间
rule_nums	Long	规则个数
children	Array of TemplateClassificationTree objects	子分类分级列表

状态码：400

表 3-214 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询敏感数据识别分级分类树结构信息

```
GET /v1/{project_id}/scan-templates/{template_id}/classifications
```

响应示例

状态码：200

请求成功

```
{
  "template_id" : "xxxxxxxxxxxx",
  "template_category" : "BUILT_IN",
  "template_name" : "xxxxxxxxxxxx",
  "classification_trees" : [ {
    "id" : "xxxxxxxxxxxx",
    "project_id" : "xxxxxxxxxxxx",
    "name" : "xxxxxxxxxxxx",
    "root_id" : "xxxxxxxxxxxx",
    "parent_id" : "",
```

```
    "depth" : 1,
    "children_nums" : 1,
    "create_time" : 1686888427971,
    "update_time" : 1686888427971,
    "rule_nums" : 3,
    "children" : [ {
      "id" : "xxxxxxxxxxx",
      "project_id" : "xxxxxxxxxxx",
      "name" : "xxxxxxxxxxx",
      "root_id" : "xxxxxxxxxxx",
      "parent_id" : "xxxxxxxxxxx",
      "depth" : 1,
      "children_nums" : 0,
      "create_time" : 1686888427971,
      "update_time" : 1686888427971,
      "rule_nums" : 5,
      "children" : null
    } ]
  } ]
}
```


状态码：400

无效请求


```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```


状态码

状态码	描述
200	请求成功
400	无效请求

错误码

请参见**错误码**。

3.11.9 查询模板规则关联关系

功能介绍

查询模板规则关联关系

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/scan-templates/{template_id}/scan-rules

表 3-215 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
template_id	是	String	模板ID

表 3-216 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	页码
limit	否	Integer	分页大小
keyword	否	String	敏感信息对象名称关键字
classification_ids	否	Array of strings	分类ID
security_level_ids	否	Array of strings	风险等级ID数组
is_used	否	Boolean	是否启用
rule_name	否	String	规则名称关键字

请求参数

无

响应参数

状态码：200

表 3-217 响应 Body 参数

参数	参数类型	描述
total	Long	规则总数
template_rules_list	Array of **TemplateRule** objects	规则列表

表 3-218 TemplateRule

参数	参数类型	描述
rule_id	String	规则ID

参数	参数类型	描述
project_id	String	项目ID
rule_name	String	规则名称
template_id	String	模板ID
classification_id	String	分级分类ID
security_level_id	String	风险等级ID
security_level_name	String	风险等级名称
security_level_color	Integer	风险等级
is_used	Boolean	是否启用
rule_description	String	规则名称

状态码：400

表 3-219 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询模板规则关联关系

```
GET /v1/{project_id}/scan-templates/{template_id}/scan-rules
```

响应示例

状态码：200

请求成功

```
{
  "template_rules_list" : [ {
    "project_id" : "xxxxxxxxxxxx",
    "template_id" : "xxxxxxxxxxxx",
    "classification_id" : "xxxxxxxxxxxx",
    "security_level_id" : "xxxxxxxxxxxx",
    "security_level_name" : "L1",
    "security_level_color" : 1,
    "rule_id" : "xxxxxxxxxxxx",
    "rule_name" : "xxxxxxxxxxxx",
    "is_used" : true,
    "rule_description" : "xxxxxxxxxxxx"
```

```
  } ],
  "total" : 1
}
```

状态码：400

无效请求

```
{
  "error_code" : "dsc.40000011",
  "error_msg" : "Invalid parameter"
}
```

状态码

状态码	描述
200	请求成功
400	无效请求

错误码

请参见**错误码**。

A 附录

A.1 状态码

状态码	编码	状态说明
200	OK	请求成功。
400	Bad Request	请求失败。 建议直接修改该请求，不要重试该请求。
401	Unauthorized	请求认证失败。 表明服务端指出客户端所提供的认证信息不正确或非法。
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。
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。

A.2 获取项目 ID

调用 API 获取项目 ID

项目ID可以通过调用**查询指定条件下的项目信息**API获取。

获取项目ID的接口为“GET https://{Endpoint}/v3/projects”，其中{Endpoint}为IAM的终端节点，可以从**地区和终端节点**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”即为项目ID。

```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "xxxxxxxx",
            "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. 单击用户名，在下拉列表中单击“我的凭证”。
3. 在“API凭证”页面的项目列表中查看项目ID。

图 A-1 查看项目 ID

A.3 动态脱敏策略配置

A.3.1 SHA256/512

该算法将指定字段的内容用其SHA256/SHA512的摘要值代替。

请求参数

表 A-1 请求参数

参数	是否必选	参数类型	描述
algorithm	是	String	算法类型，SHA256/SHA512
parameters	是	Object	参数，该算法无参数可忽略

请求样例

```
{
"mask_strategies": [
{
"name": "col1",
"algorithm": "SHA256",
"parameters": {}
}
],
"data": [
{
"col1": "test"
]
}
```

A.3.2 AES

将目标字符串使用AES算法进行加密。

AES算法参数配置如下：初始向量IV为16字节随机数，加密模式为GCM模式，填充方式为PKCS7-Padding（即CMS-Padding）。

加密脱敏的密文中，前16字节存放IV值，后续为对应的密文内容。加密密文为二进制，脱敏引擎输入的密文经过Base64编码，以字符串的形式存储在数据库中。

请求参数

表 A-2 请求参数

参数	是否必选	参数类型	描述
algorithm	是	String	算法类型，AES
parameters	是	**parameters** Object	脱敏算法参数

表 A-3 parameters 参数

参数	是否必选	参数类型	描述
key	是	String	AES算法密钥

参数	是否必选	参数类型	描述
len	是	String	分组长度，目前取值仅支持128，192，256

请求样例

```
{
"mask_strategies": [
{
"name": "col1",
"algorithm": "AES",
"parameters": {
"key": "df643533b90b6926c9bff63cc16173db",
"len": "128"
}
}
],
"data": [
{
"col1": "test"
}
]
}
```

A.3.3 PRESNM

对目标字符串保留前n位和后m位，对中间部分内容掩盖。

请求参数

表 A-4 请求参数

参数	是否必选	参数类型	描述
algorithm	是	String	算法类型，PRESNM
parameters	是	**parameters** Object	脱敏算法参数

表 A-5 parameters

参数	是否必选	参数类型	描述
type	是	String	掩盖字符串的方式，取值如下： • CHAR：字符掩盖 • RAND：随机掩盖

参数	是否必选	参数类型	描述
method	是	String	掩盖参数，若type字段为“RAND”时，method取值只能为： ● CHAR：替换为字母 ● DIGITAL：替换为数字 ● BOTH：数字与字母混合替换
n	是	Int	保留前n位字符
m	是	Int	保留后m位字符

请求样例

```
{
  "mask_strategies": [
    {
      "name": "col1",
      "algorithm": "PRESNM",
      "parameters": {
        "type": "CHAR",
        "n": 1,
        "m": 1,
        "method": "*"
      }
    }
  ],
  "data": [
    {
      "col1": "test"
    }
  ]
}
或:
{
  "mask_strategies": [
    {
      "name": "col1",
      "algorithm": "PRESNM",
      "parameters": {
        "type": "RAND",
        "n": 1,
        "m": 1,
        "method": "BOTH"
      }
    }
  ],
  "data": [
    {
      "col1": "test"
    }
  ]
}
```

A.3.4 MASKNM

对目标字符串掩盖前n位和后m位，对中间部分内容保留。

请求参数

表 A-6 请求参数

参数	是否必选	参数类型	描述
algorithm	是	String	算法类型，MASKNM
parameters	是	**parameters** Object	脱敏算法参数

表 A-7 parameters

参数	是否必选	参数类型	描述
type	是	String	掩盖字符串的方式，取值为： ● CHAR：字符掩盖 ● RAND：随机掩盖
method	是	String	掩盖参数，若type字段为“RAND”时，method取值只能为： ● CHAR：替换为字母 ● DIGITAL：替换为数字 ● BOTH：数字与字母混合替换
n	是	Int	掩盖前n位字符
m	是	Int	掩盖后m位字符

请求样例

```
{
   "mask_strategies": [
      {
         "name": "col1",
         "algorithm": "MASKNM",
         "parameters": {
            "type": "CHAR",
            "n": 1,
            "m": 1,
            "method": "*"
         }
      }
   ],
   "data": [
      {
         "col1": "test"
      }
   ]
}
或:
{
```

```
  "mask_strategies": [
    {
      "name": "col1",
      "algorithm": "MASKNM",
      "parameters": {
        "type": "RAND",
        "n": 1,
        "m": 1,
        "method": "BOTH"
      }
    }
  ],
  "data": [
    {
      "col1": "test"
    }
  ]
}
```

A.3.5 PRESXY

保留目标字符串x至y的内容，其他部分内容掩盖。

请求参数

表 A-8 请求参数

参数	是否必选	参数类型	描述
algorithm	是	String	算法类型，PRESXY
parameters	是	**parameters** Object	脱敏算法参数

表 A-9 parameters

参数	是否必选	参数类型	描述
type	是	String	掩盖字符串的方式，取值为： • CHAR：字符掩盖 • RAND：随机掩盖
method	是	String	掩盖参数，若type字段为“RAND”时，method取值只能为： • CHAR：替换为字母 • DIGITAL：替换为数字 • BOTH：数字与字母混合替换
x	是	Int	保留字符的开始位置
y	是	Int	保留字符的结束位置

请求样例

```
{
   "mask_strategies": [
      {
         "name": "col1",
         "algorithm": "PRESXY",
         "parameters": {
            "type": "CHAR",
            "x": 1,
            "y": 1,
            "method": "*"
         }
      }
   ],
   "data": [
      {
         "col1": "test"
      }
   ]
}
或:
{
   "mask_strategies": [
      {
         "name": "col1",
         "algorithm": "PRESXY",
         "parameters": {
            "type": "RAND",
            "x": 1,
            "y": 1,
            "method": "CHAR"
         }
      }
   ],
   "data": [
      {
         "col1": "test"
      }
   ]
}
```

A.3.6 MASKXY

掩盖字符串x至y的内容，其他部分内容保留。

请求参数

表 A-10 请求参数

参数	是否必选	参数类型	描述
algorithm	是	String	算法类型，MASKXY
parameters	是	**parameters** Object	脱敏算法参数

表 A-11 parameters

参数	是否必选	参数类型	描述
type	是	String	掩盖字符串的方式，取值为： ● CHAR：字符掩盖 ● RAND：随机掩盖
method	是	String	掩盖参数，若type字段为“RAND”时，method取值只能为： ● CHAR：替换为字母 ● DIGITAL：替换为数字 ● BOTH：数字与字母混合替换
x	是	Int	掩盖字符的开始位置
y	是	Int	掩盖字符的结束位置

请求样例

```
{
   "mask_strategies": [
      {
         "name": "col1",
         "algorithm": "MASKXY",
         "parameters": {
            "type": "CHAR",
            "x": 1,
            "y": 1,
            "method": "*"
         }
      }
   ],
   "data": [
      {
         "col1": "test"
      }
   ]
}
或:
{
   "mask_strategies": [
      {
         "name": "col1",
         "algorithm": "MASKXY",
         "parameters": {
            "type": "RAND",
            "x": 1,
            "y": 1,
            "method": "CHAR"
         }
      }
   ],
   "data": [
      {
         "col1": "test"
      }
```

```
  ]
}
```

A.3.7 SYMBOL

对特殊符号前或后的内容掩盖，另一部分内容保留。

请求参数

表 A-12 请求参数

参数	是否必选	参数类型	描述
algorithm	是	String	算法类型，SYMBOL
parameters	是	**parameters** Object	脱敏算法参数

表 A-13 parameters

参数	是否必选	参数类型	描述
type	是	String	掩盖字符串的方式，取值为： ● CHAR：字符掩盖 ● RAND：随机掩盖
method	是	String	掩盖参数，若type字段为“RAND”时，method取值只能为： ● CHAR：替换为字母 ● DIGITAL：替换为数字 ● BOTH：数字与字母混合替换
direction	是	Int	掩盖的方向 0：特殊字符前掩盖 1：特殊字符后掩盖
symbol	是	String	指定的特殊字符

请求样例

```
{
  "mask_strategies": [
    {
      "name": "col1",
      "algorithm": "SYMBOL",
      "parameters": {
        "type": "CHAR",
        "direction": 1,
```

```
            "symbol": "@",
            "method": "x"
         }
      }
   ],
   "data": [
      {
         "col1": "test"
      }
   ]
}
或:
{
   "mask_strategies": [
      {
         "name": "col1",
         "algorithm": "SYMBOL", //脱敏算法参数
         "parameters": {
            "type": "RAND",
            "direction": 0,
            "symbol": "@",
            "method": "CHAR"
         }
      }
   ],
   "data": [
      {
         "col1": "test"
      }
   ]
}
```

A.3.8 KEYWORD

对目标字符串中的指定关键词进行替换。

请求参数

表 A-14 请求参数

参数	是否必选	参数类型	描述
algorithm	是	String	算法类型，KEYWORD
parameters	是	**parameters** Object	脱敏算法参数

表 A-15 parameters

参数	是否必选	参数类型	描述
key	是	String	指定的关键词
target	是	String	替换后的目标值

请求样例

```
{
   "mask_strategies": [
```

```
    {
      "name": "col1",
      "algorithm": "KEYWORD",
      "parameters": {
        "key": "关键字",
        "target": "目标字符串"
      }
    }
  ],
  "data": [
    {
      "col1": "关键字"
    }
  ]
}
```

A.3.9 NUMERIC

对数值型字段向下泛化。

请求参数

表 A-16 请求参数

参数	是否必选	参数类型	描述
algorithm	是	String	算法类型，NUMERIC
parameters	是	**parameters** Object	脱敏算法参数

表 A-17 parameters

参数	是否必选	参数类型	描述
value	是	Double	指定数据类需要泛化的区间范围值，只能为整数或者小数，且必须大于0

请求样例

```
{
  "mask_strategies": [
    {
      "name": "col1",
      "algorithm": " NUMERIC",
      "parameters": {
        "value": 0.05
      }
    }
  ],
  "data": [
    {
      "col1": "test"
    }
  ]
}
```

A.4 错误码

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。

状态码	错误码	错误信息	描述	处理措施
400	DSC.00000001	The project ID is not in the trust list	Project ID不在白名单列表中!	请重试或联系技术支持。
400	DSC.00000002	Internal service error	内部错误!	请重试或联系技术支持。
400	DSC.00000003	Failed to read the upload file	读取文件错误!	请检查文件类型和大小。
400	DSC.00000004	Invalid parameter	参数错误!	请传递正确的参数。
400	DSC.00000005	Invalid project id	Project ID长度错误!	请重试或联系技术支持。
400	DSC.00000006	File size exceeds maximum limit	上传文件超过最大限制!	请检查文件的大小。
400	DSC.00000007	File format error	文件格式错误!	请检查文件的格式。
400	DSC.00000008	Upload file is empty	上传文件为空!	请检查文件内容是否为空。
400	DSC.00000009	There are many tasks at present. Please try again later	当前任务较多，请稍后再试!	请重试或联系技术支持。
400	DSC.00000010	Image watermark file is empty	文档图形水印文件为空!	请添加图形文件。
400	DSC.00000011	Image watermark file exceeds maximum limit	文档图形水印文件超过最大限制!	请限制图形文件大小。

状态码	错误码	错误信息	描述	处理措施
400	DSC.00000012	Invalid project image watermark file type	文档图形水印类型错误!	请上传.png或.jpg格式的文件。
400	DSC.00000013	Invalid Auth	认证失败!	请使用有效token。
400	mask.20000001	Unknown error	未知错误!	请重试或联系技术支持。
400	mask.20000002	Internal service error	内部错误!	请重试或联系技术支持。
400	mask.20000003	Invalid parameter	参数错误!	请传递正确的参数
400	mask.20000004	Mask data failed	数据脱敏失败!	请传递正确的参数
400	watermark.10000001	Unknown error	未知错误!	请重试或联系技术支持。
400	watermark.10000002	Internal service error	内部错误!	请重试或联系技术支持。
400	watermark.10000003	Failed to read the upload file	读取文件错误!	请检查文件类型和大小
400	watermark.10000004	Invalid parameter	参数错误!	请传递正确的参数
400	watermark.10000005	Watermark not found	该文件的水印未找到!	请检查该文件是否加过水印。
400	watermark.10000006	File size exceeds maximum limit	上传文件超过最大限制!	请检查文件的大小。
400	watermark.10000007	File format error	文件格式错误!	请检查文件的格式。
400	watermark.10000008	Upload file is empty	上传文件为空!	请检查文件内容是否为空。
400	watermark.10000009	Wrong password	密码错误!	请检查文件的密码是否正确。

状态码	错误码	错误信息	描述	处理措施
400	watermark.10000010	The ".doc" format file does not support the watermark feature currently	.doc格式文件暂不支持水印功能!	请检查文件的格式。
400	watermark.10000011	The ".xls" format file does not support the visible watermark feature currently	.xls格式文件暂不支持明水印功能!	请检查文件的格式。
400	watermark.10000012	The ".ppt" format file does not support the visible watermark feature currently	.ppt格式文件暂不支持明水印功能!	请检查文件的格式。
400	watermark.10000013	No watermark specified	没有指定水印内容!	请检查水印参数是否传递。
400	watermark.10000014	There are many tasks at present. Please try again later	当前任务较多，请稍后再试!	请重试或联系技术支持。
400	watermark.10000015	Blind watermark already exists	盲水印已经存在!	请检查文件是否已存在盲水印。
400	watermark.10000016	There is not enough channel capacity to embed the watermark	文件解压后超过最大限制!	请检查文件大小。
400	watermark.10000017	Image channel capacity is too small to embed watermark	图像信道容量过小，不能嵌入水印!	请重试或联系技术支持。

状态码	错误码	错误信息	描述	处理措施
400	watermark.10000018	Currently does not support embedding watermark to image below 512 pixels	当前不支持给低于512像素的图片添加水印!	请输入大于512像素的图片，或联系技术支持。