安全云脑

API 参考

文档版本 08

发布日期 2024-12-11

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。

目 录

1 使用前必读

安全云脑（SecMaster）是华为云原生的新一代安全运营中心，基于云原生安全，提供云上资产管理、安全态势管理、安全信息和事件管理、安全编排与自动响应等能力，实现提前预防风险、感知安全事件、安全事件自动化闭环。

在调用安全云脑API之前，请确保已经充分了解安全云脑相关概念，详细信息请参见**产品介绍**。

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**中查询所有服务的终端节点。

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他们进行日常管理工作。

- 用户

  由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。

  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）

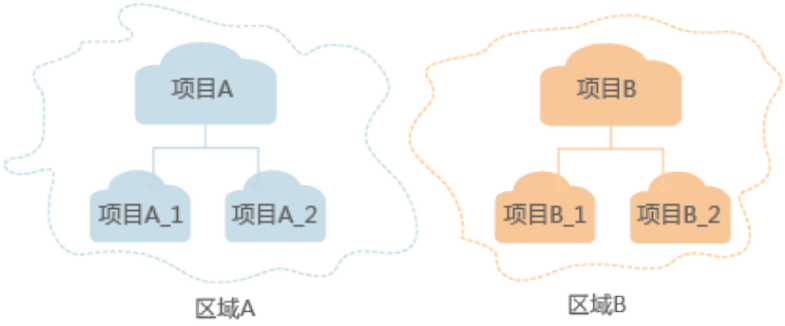
  一个AZ是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

- 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



- 企业项目

  企业项目是项目的升级版，针对企业不同项目间资源的分组和管理，是逻辑隔离。企业项目中可以包含多个区域的资源，且项目中的资源可以迁入迁出。

  关于企业项目ID的获取及企业项目特性的详细信息，请参见**企业管理服务用户指南**。

2 API 概览

通过使用安全云脑提供的接口，您可以完整的使用安全云脑的所有功能。

类型	说明
告警API接口	告警的接口，包括创建、删除、转事件等接口。
事件API接口	事件的接口，包括创建、更新、获取等接口。
威胁情报API接口	威胁情报的接口，包括创建、更新、获取等接口。
剧本API接口	剧本的接口，包括查询、创建、修改等接口。
告警规则API接口	告警规则的接口，包括创建、删除、查看、启用等接口。
剧本版本API接口	剧本版本的接口，包括查询、创建、更新等接口。
剧本规则API接口	剧本规则的接口，包括创建、查询、删除等接口。
剧本实例API接口	剧本实例的接口，包括查询、操作等接口。
剧本审核API接口	剧本审核的接口，包括审核剧本、查询剧本审核结果的接口。
事件关系API接口	事件关系的接口，包括查询、关联、取消关联接口。
数据类API接口	数据类的接口，包括查询数据类列表和字段列表接口。
流程API接口	流程的接口，包括查询流程列表接口。
数据空间API接口	数据空间的接口，包括创建数据空间接口。
管道API接口	管道的接口，包括创建管道接口。
工作空间API接口	工作空间的接口，包括新增、查询列表接口。
计量计费API接口	计量计费的接口，包括计量计费的接口。
指标查询API接口	指标查询的接口，包括批量查询指标接口。
基线API接口	基线的接口，包括搜索检查结果接口。

3 如何调用 API

3.1 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”。

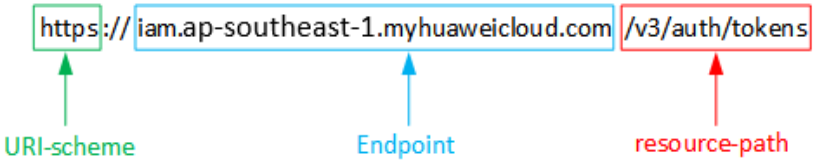
- **query-string**：

  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 3-1 URI 示意图



说明

为查看方便，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，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。

3.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

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

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

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。

3.3 返回结果

状态码

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

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

对于**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3-2**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 3-2 获取用户 Token 响应消息头

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

其中，error_code表示错误码，error_msg表示错误描述信息。

4 API

4.1 告警管理

4.1.1 搜索告警列表

功能介绍

搜索告警列表

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/alerts/search

表 4-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

请求参数

表 4-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	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）
content-type	是	String	内容类型

表 4-3 请求 Body 参数

参数	是否必选	参数类型	描述
limit	否	Integer	分页大小
offset	否	Integer	偏移量
sort_by	否	String	排序字段：create_time | update_time
order	否	String	排序方式：DESC | ASC
from_date	否	String	搜索开始时间，例如：2023-02-20T00:00:00.000Z
to_date	否	String	搜索结束时间，例如：2023-02-27T23:59:59.999Z
condition	否	**condition** object	搜索条件表达式

表 4-4 condition

参数	是否必选	参数类型	描述
conditions	否	Array of **conditions** objects	表达式列表
logics	否	Array of strings	表达式名称列表

表 4-5 conditions

参数	是否必选	参数类型	描述
name	否	String	表达式名称
data	否	Array of strings	表达式内容列表

响应参数

状态码： 200

表 4-6 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-7 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
total	Integer	告警总数
limit	Integer	分页大小
offset	Integer	偏移量
success	Boolean	是否成功
data	Array of **ListAlertDetail** objects	告警列表

表 4-8 ListAlertDetail

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
data_object	**ListAlertRsp** object	告警实体信息

参数	参数类型	描述
dataclass_ref	**dataclass_ref** object	数据类对象
format_version	Integer	格式版本
id	String	告警唯一标识，UUID格式，最大36个字符
type	String	数据类型
project_id	String	当前项目的id
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
version	Integer	版本
workspace_id	String	当前的工作空间id

表 4-9 ListAlertRsp

参数	参数类型	描述
version	String	告警对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	String	事件唯一标识，UUID格式，最大36个字符
domain_id	String	数据投递后，被委托用户的domain_id
region_id	String	数据投递后，被委托用户的region_id
workspace_id	String	当前的工作空间id
labels	String	标签，仅展示
environment	**environment** object	告警产生的环境坐标信息
data_source	**data_source** object	首次上报数据源
first_observed_time	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	String	告警标题
description	String	告警描述信息
source_url	String	告警URL链接，指向数据源产品中有关当前事件说明的页面
count	Integer	事件发生次数
confidence	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%
severity	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源
alert_type	**alert_type** object	告警分类，详细定义参考《告警类型定义》
network_list	Array of **network_list** objects	网络信息
resource_list	Array of **resource_list** objects	受影响资源

参数	参数类型	描述
remediation	**remediation** object	补救措施
verification_state	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	String	约束闭环时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
chop_phase	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
ppdr_phase	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	String	调试字段
actor	String	告警调查员

参数	参数类型	描述
owner	String	责任人、服务责任人
creator	String	创建人
close_reason	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	String	关闭评论
malware	**malware** object	恶意软件
system_info	Object	系统信息
process	Array of **process** objects	进程信息
user_info	Array of **user_info** objects	用户信息
file_info	Array of **file_info** objects	文件信息
system_alert_table	Object	告警管理列表的布局字段

表 4-10 environment

参数	参数类型	描述
vendor_type	String	环境供应商
domain_id	String	租户id
region_id	String	区域id，全局服务global
cross_workspace_id	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	String	项目id， 全局服务默认null

表 4-11 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下： 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	数据源产品所属账号的id
project_id	String	数据源产品所属项目的id
region_id	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	String	数据源产品所属公司的名称
product_name	String	数据源产品的名称
product_feature	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	String	检测模块列表

表 4-12 alert_type

参数	参数类型	描述
category	String	类别
alert_type	String	告警类型

表 4-13 network_list

参数	参数类型	描述
direction	String	方向，取值范围：IN | OUT
protocol	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	String	源IP地址
src_port	Integer	源端口，0–65535
src_domain	String	源域名

参数	参数类型	描述
src_geo	**src_geo** object	源IP的地理位置信息
dest_ip	String	目的IP地址
dest_port	String	目的端口，0–65535
dest_domain	String	目的域名
dest_geo	**dest_geo** object	目标IP的地理位置信息

表 4-14 src_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-15 dest_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-16 resource_list

参数	参数类型	描述
id	String	云服务资源id
name	String	资源名称
type	String	资源类型；引用云RMS type字段
provider	String	云服务名称；引用云RMS provider字段
region_id	String	区域；按照云regionId填写，如cn-north-1等

参数	参数类型	描述
domain_id	String	资源所属账号ID，UUID格式
project_id	String	资源所属项目ID，UUID格式
ep_id	String	企业项目id
ep_name	String	企业项目名称
tags	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-17 remediation

参数	参数类型	描述
recommendation	String	推荐处理方法
url	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-18 malware

参数	参数类型	描述
malware_family	String	恶意家族
malware_class	String	恶意软件分类

表 4-19 process

参数	参数类型	描述
process_name	String	进程名
process_path	String	进程执行文件路径
process_pid	Integer	进程id
process_uid	Integer	进程用户id
process_cmdline	String	进程命令行
process_parent_name	String	父进程名称

参数	参数类型	描述
process_parent_path	String	父进程执行文件路径
process_parent_pid	Integer	父进程id
process_parent_uid	Integer	父进程用户id
process_parent_cmdline	String	父进程命令行
process_child_name	String	子进程名称
process_child_path	String	子进程执行文件路径
process_child_pid	Integer	子进程id
process_child_uid	Integer	子进程用户id
process_child_cmdline	String	子进程命令行
process_launche_time	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-20 user_info

参数	参数类型	描述
user_id	String	用户uid
user_name	String	用户名称

表 4-21 file_info

参数	参数类型	描述
file_path	String	文件路径/名称
file_content	String	文件内容

参数	参数类型	描述
file_new_path	String	文件新路径/名称
file_hash	String	文件hash
file_md5	String	文件md5
file_sha256	String	文件sha256
file_attr	String	文件属性

表 4-22 dataclass_ref

参数	参数类型	描述
id	String	数据类唯一标识，UUID格式，最大36个字符
name	String	数据类名称

状态码： 400

表 4-23 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-24 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

查询告警列表请求样例，查询2024年1月20号到2024年1月26号，告警等级为中危且处理状态为打开的告警，按照创建时间降序排序，返回第一页，每页10条数据

```
{
  "limit" : 10,
  "offset" : 0,
  "sort_by" : "create_time",
  "order" : "DESC",
  "condition" : {
    "conditions" : [ {
```

```
    "name" : "severity",
    "data" : [ "severity", "=", "Medium" ]
   }, {
    "name" : "handle_status",
    "data" : [ "handle_status", "=", "Open" ]
   } ],
   "logics" : [ "severity", "and", "handle_status" ]
  },
  "from_date" : "2024-01-20T00:00:00.000Z+0800",
  "to_date" : "2024-01-26T23:59:59.999Z+0800"
}
```

响应示例

状态码： 200

搜索告警列表返回body体

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "total" : 41,
  "limit" : 2,
  "offset" : 1,
  "success" : true,
  "data" : [ {
    "data_object" : {
      "version" : "1.0",
      "environment" : {
        "vendor_type" : "MyXXX",
        "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
      },
      "data_source" : {
        "source_type" : 3,
        "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
      },
      "first_observed_time" : "2021-01-30T23:00:00Z+0800",
      "last_observed_time" : "2021-01-30T23:00:00Z+0800",
      "create_time" : "2021-01-30T23:00:00Z+0800",
      "arrive_time" : "2021-01-30T23:00:00Z+0800",
      "title" : "MyXXX",
      "description" : "This my XXXX",
      "source_url" : "http://xxx",
      "count" : 4,
      "confidence" : 4,
      "severity" : "TIPS",
      "criticality" : 4,
      "alert_type" : { },
      "network_list" : [ {
        "direction" : {
          "IN" : null
        },
        "protocol" : "TCP",
        "src_ip" : "192.168.0.1",
        "src_port" : "1",
        "src_domain" : "xxx",
        "dest_ip" : "192.168.0.1",
        "dest_port" : "1",
        "dest_domain" : "xxx",
        "src_geo" : {
          "latitude" : 90,
          "longitude" : 180
        },
        "dest_geo" : {
          "latitude" : 90,
```

```
        "longitude" : 180
      }
    } ],
    "resource_list" : [ {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "MyXXX",
      "type" : "MyXXX",
      "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "ep_name" : "MyXXX",
      "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    } ],
    "remediation" : {
      "recommendation" : "MyXXX",
      "url" : "MyXXX"
    },
    "verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写
Unknown",
    "handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
    "sla" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "close_time" : "2021-01-30T23:00:00Z+0800",
    "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
    "simulation" : "false",
    "actor" : "刘一博",
    "owner" : "MyXXX",
    "creator" : "MyXXX",
    "close_reason" : "误检;已解决;重复;其他",
    "close_comment" : "误检;已解决;重复;其他",
    "malware" : {
      "malware_family" : "family",
      "malware_class" : "恶意占用内存"
    },
    "system_info" : { },
    "process" : [ {
      "process_name" : "MyXXX",
      "process_path" : "MyXXX",
      "process_pid" : 123,
      "process_uid" : 123,
      "process_cmdline" : "MyXXX"
    } ],
    "user_info" : [ {
      "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "user_name" : "MyXXX"
    } ],
    "file_info" : [ {
      "file_path" : "MyXXX",
      "file_content" : "MyXXX",
      "file_new_path" : "MyXXX",
      "file_hash" : "MyXXX",
      "file_md5" : "MyXXX",
      "file_sha256" : "MyXXX",
      "file_attr" : "MyXXX"
    } ],
    "system_alert_table" : { },
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
  },
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "id" : "MyXXX",
  "version" : 123,
  "format_version" : 123,
  "dataclass_ref" : {
```

```
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX"
   }
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

查询告警列表请求样例，查询2024年1月20号到2024年1月26号，告警等级为中危且处理状态为打开的告警，按照创建时间降序排序，返回第一页，每页10条数据

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class ListAlertsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAlertsRequest request = new ListAlertsRequest();
        request.withWorkspaceId("{workspace_id}");
        DataobjectSearch body = new DataobjectSearch();
        List<String> listConditionLogics = new ArrayList<>();
        listConditionLogics.add("severity");
        listConditionLogics.add("and");
        listConditionLogics.add("handle_status");
        List<String> listConditionsData = new ArrayList<>();
        listConditionsData.add("handle_status");
        listConditionsData.add("=");
        listConditionsData.add("Open");
        List<String> listConditionsData1 = new ArrayList<>();
        listConditionsData1.add("severity");
        listConditionsData1.add("=");
        listConditionsData1.add("Medium");
        List<DataobjectSearchConditionConditions> listConditionConditions = new ArrayList<>();
        listConditionConditions.add(
```

```
            new DataobjectSearchConditionConditions()
                .withName("severity")
                .withData(listConditionsData1)
        );
        listConditionConditions.add(
            new DataobjectSearchConditionConditions()
                .withName("handle_status")
                .withData(listConditionsData)
        );
        DataobjectSearchCondition conditionbody = new DataobjectSearchCondition();
        conditionbody.withConditions(listConditionConditions)
            .withLogics(listConditionLogics);
        body.withCondition(conditionbody);
        body.withToDate("2024-01-26T23:59:59.999Z+0800");
        body.withFromDate("2024-01-20T00:00:00.000Z+0800");
        body.withOrder(DataobjectSearch.OrderEnum.fromValue("DESC"));
        body.withSortBy("create_time");
        body.withOffset(0);
        body.withLimit(10);
        request.withBody(body);
        try {
            ListAlertsResponse response = client.listAlerts(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

查询告警列表请求样例，查询2024年1月20号到2024年1月26号，告警等级为中危且处理状态为打开的告警，按照创建时间降序排序，返回第一页，每页10条数据

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListAlertsRequest()
```

```
        request.workspace_id = "{workspace_id}"
        listLogicsCondition = [
            "severity",
            "and",
            "handle_status"
        ]
        listDataConditions = [
            "handle_status",
            "=",
            "Open"
        ]
        listDataConditions1 = [
            "severity",
            "=",
            "Medium"
        ]
        listConditionsCondition = [
            DataobjectSearchConditionConditions(
                name="severity",
                data=listDataConditions1
            ),
            DataobjectSearchConditionConditions(
                name="handle_status",
                data=listDataConditions
            )
        ]
        conditionbody = DataobjectSearchCondition(
            conditions=listConditionsCondition,
            logics=listLogicsCondition
        )
        request.body = DataobjectSearch(
            condition=conditionbody,
            to_date="2024-01-26T23:59:59.999Z+0800",
            from_date="2024-01-20T00:00:00.000Z+0800",
            order="DESC",
            sort_by="create_time",
            offset=0,
            limit=10
        )
        response = client.list_alerts(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

查询告警列表请求样例，查询2024年1月20号到2024年1月26号，告警等级为中危且处理状态为打开的告警，按照创建时间降序排序，返回第一页，每页10条数据

```
package main

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

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

```
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

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

client := secmaster.NewSecMasterClient(
    secmaster.SecMasterClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ListAlertsRequest{}
request.WorkspaceId = "{workspace_id}"
var listLogicsCondition = []string{
    "severity",
    "and",
    "handle_status",
}
var listDataConditions = []string{
    "handle_status",
    "=",
    "Open",
}
var listDataConditions1 = []string{
    "severity",
    "=",
    "Medium",
}
nameConditions:= "severity"
nameConditions1:= "handle_status"
var listConditionsCondition = []model.DataobjectSearchConditionConditions{
    {
        Name: &nameConditions,
        Data: &listDataConditions1,
    },
    {
        Name: &nameConditions1,
        Data: &listDataConditions,
    },
}
conditionbody := &model.DataobjectSearchCondition{
    Conditions: &listConditionsCondition,
    Logics: &listLogicsCondition,
}
toDateDataobjectSearch:= "2024-01-26T23:59:59.999Z+0800"
fromDateDataobjectSearch:= "2024-01-20T00:00:00.000Z+0800"
orderDataobjectSearch:= model.GetDataobjectSearchOrderEnum().DESC
sortByDataobjectSearch:= "create_time"
offsetDataobjectSearch:= int32(0)
limitDataobjectSearch:= int32(10)
request.Body = &model.DataobjectSearch{
    Condition: conditionbody,
    ToDate: &toDateDataobjectSearch,
    FromDate: &fromDateDataobjectSearch,
    Order: &orderDataobjectSearch,
    SortBy: &sortByDataobjectSearch,
    Offset: &offsetDataobjectSearch,
    Limit: &limitDataobjectSearch,
}
response, err := client.ListAlerts(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	搜索告警列表返回body体
400	搜索告警列表错误返回body体

错误码

请参见**错误码**。

4.1.2 创建告警

功能介绍

创建告警

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/alerts

表 4-25 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

请求参数

表 4-26 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）
content-type	是	String	内容类型

表 4-27 请求 Body 参数

参数	是否必选	参数类型	描述
data_object	是	**Alert** object	告警实体信息

表 4-28 Alert

参数	是否必选	参数类型	描述
version	否	String	告警对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	否	String	事件唯一标识，UUID格式，最大36个字符
domain_id	否	String	数据投递后，被委托用户的domain_id
region_id	否	String	数据投递后，被委托用户的region_id
workspace_id	否	String	当前的工作空间id
labels	否	String	标签，仅展示
environment	否	**environment** object	告警产生的环境坐标信息
data_source	否	**data_source** object	首次上报数据源
first_observed_time	否	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

参数	是否必选	参数类型	描述
last_observed_time	否	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
create_time	否	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	否	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	否	String	告警标题
description	否	String	告警描述信息
source_url	否	String	告警URL链接，指向数据源产品中有关当前事件说明的页面
count	否	Integer	事件发生次数
confidence	否	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%
severity	否	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	否	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源

参数	是否必选	参数类型	描述
alert_type	否	**alert_type** object	告警分类，详细定义参考《告警类型定义》
network_list	否	Array of **network_list** objects	网络信息
resource_list	否	Array of **resource_list** objects	受影响资源
remediation	否	**remediation** object	补救措施
verification_state	否	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	否	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	否	Integer	约束闭环时间：设置风险接受持续时间。单位：小时
update_time	否	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	否	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	否	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	否	String	调试字段
actor	否	String	告警调查员

参数	是否必选	参数类型	描述
owner	否	String	责任人、服务责任人
creator	否	String	创建人
close_reason	否	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	否	String	关闭评论
malware	否	**malware** object	恶意软件
system_info	否	Object	系统信息
process	否	Array of **process** objects	进程信息
user_info	否	Array of **user_info** objects	用户信息
file_info	否	Array of **file_info** objects	文件信息
system_alert_table	否	Object	告警管理列表的布局字段

表 4-29 environment

参数	是否必选	参数类型	描述
vendor_type	否	String	环境供应商
domain_id	否	String	租户id
region_id	否	String	区域id，全局服务global
cross_workspace_id	否	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	否	String	项目id， 全局服务默认null

表 4-30 data_source

参数	是否必选	参数类型	描述
source_type	否	Integer	数据源类型，取值范围如下： 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	否	String	数据源产品所属账号的id
project_id	否	String	数据源产品所属项目的id
region_id	否	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	否	String	数据源产品所属公司的名称
product_name	否	String	数据源产品的名称
product_feature	否	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	否	String	检测模块列表

表 4-31 alert_type

参数	是否必选	参数类型	描述
category	否	String	类别
alert_type	否	String	告警类型

表 4-32 network_list

参数	是否必选	参数类型	描述
direction	否	String	方向，取值范围：IN | OUT
protocol	否	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	否	String	源IP地址
src_port	否	Integer	源端口，0–65535

参数	是否必选	参数类型	描述
src_domain	否	String	源域名
src_geo	否	src_geo object	源IP的地理位置信息
dest_ip	否	String	目的IP地址
dest_port	否	String	目的端口，0–65535
dest_domain	否	String	目的域名
dest_geo	否	dest_geo object	目标IP的地理位置信息

表 4-33 src_geo

参数	是否必选	参数类型	描述
latitude	否	Number	纬度
longitude	否	Number	经度
city_code	否	String	城市编码，Beijing | Shanghai
country_code	否	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-34 dest_geo

参数	是否必选	参数类型	描述
latitude	否	Number	纬度
longitude	否	Number	经度
city_code	否	String	城市编码，Beijing | Shanghai
country_code	否	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-35 resource_list

参数	是否必选	参数类型	描述
id	否	String	云服务资源id
name	否	String	资源名称

参数	是否必选	参数类型	描述
type	否	String	资源类型；引用云RMS type字段
provider	否	String	云服务名称；引用云RMS provider字段
region_id	否	String	区域；按照云regionId填写，如cn-north-1等
domain_id	否	String	资源所属账号ID，UUID格式
project_id	否	String	资源所属项目ID，UUID格式
ep_id	否	String	企业项目id
ep_name	否	String	企业项目名称
tags	否	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-36 remediation

参数	是否必选	参数类型	描述
recommendation	否	String	推荐处理方法
url	否	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-37 malware

参数	是否必选	参数类型	描述
malware_family	否	String	恶意家族
malware_class	否	String	恶意软件分类

表 4-38 process

参数	是否必选	参数类型	描述
process_name	否	String	进程名

参数	是否必选	参数类型	描述
process_path	否	String	进程执行文件路径
process_pid	否	Integer	进程id
process_uid	否	Integer	进程用户id
process_cmdline	否	String	进程命令行
process_parent_name	否	String	父进程名称
process_parent_path	否	String	父进程执行文件路径
process_parent_pid	否	Integer	父进程id
process_parent_uid	否	Integer	父进程用户id
process_parent_cmdline	否	String	父进程命令行
process_child_name	否	String	子进程名称
process_child_path	否	String	子进程执行文件路径
process_child_pid	否	Integer	子进程id
process_child_uid	否	Integer	子进程用户id
process_child_cmdline	否	String	子进程命令行
process_launche_time	否	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	否	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-39 user_info

参数	是否必选	参数类型	描述
user_id	否	String	用户uid
user_name	否	String	用户名称

表 4-40 file_info

参数	是否必选	参数类型	描述
file_path	否	String	文件路径/名称
file_content	否	String	文件内容
file_new_path	否	String	文件新路径/名称
file_hash	否	String	文件hash
file_md5	否	String	文件md5
file_sha256	否	String	文件sha256
file_attr	否	String	文件属性

响应参数

状态码： 200

表 4-41 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-42 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**AlertDetail** object	告警详情对象

表 4-43 AlertDetail

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
data_object	**Alert** object	告警实体信息
dataclass_ref	**dataclass_ref** object	数据类对象
format_version	Integer	格式版本
id	String	事件唯一标识，UUID格式，最大36个字符
type	String	数据类型
project_id	String	当前项目的id
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
version	Integer	版本
workspace_id	String	当前的工作空间id

表 4-44 Alert

参数	参数类型	描述
version	String	告警对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	String	事件唯一标识，UUID格式，最大36个字符
domain_id	String	数据投递后，被委托用户的domain_id
region_id	String	数据投递后，被委托用户的region_id
workspace_id	String	当前的工作空间id
labels	String	标签，仅展示
environment	**environment** object	告警产生的环境坐标信息
data_source	**data_source** object	首次上报数据源

参数	参数类型	描述
first_observed_time	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	String	告警标题
description	String	告警描述信息
source_url	String	告警URL链接，指向数据源产品中有关当前事件说明的页面
count	Integer	事件发生次数
confidence	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%
severity	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源

参数	参数类型	描述
alert_type	**alert_type** object	告警分类，详细定义参考《告警类型定义》
network_list	Array of **network_list** objects	网络信息
resource_list	Array of **resource_list** objects	受影响资源
remediation	**remediation** object	补救措施
verification_state	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	Integer	约束闭环时间：设置风险接受持续时间。单位：小时
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	String	调试字段
actor	String	告警调查员
owner	String	责任人、服务责任人
creator	String	创建人

参数	参数类型	描述
close_reason	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	String	关闭评论
malware	**malware** object	恶意软件
system_info	Object	系统信息
process	Array of **process** objects	进程信息
user_info	Array of **user_info** objects	用户信息
file_info	Array of **file_info** objects	文件信息
system_alert_table	Object	告警管理列表的布局字段

表 4-45 environment

参数	参数类型	描述
vendor_type	String	环境供应商
domain_id	String	租户id
region_id	String	区域id，全局服务global
cross_workspace_id	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	String	项目id， 全局服务默认null

表 4-46 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下: 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	数据源产品所属账号的id

参数	参数类型	描述
project_id	String	数据源产品所属项目的id
region_id	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	String	数据源产品所属公司的名称
product_name	String	数据源产品的名称
product_feature	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	String	检测模块列表

表 4-47 alert_type

参数	参数类型	描述
category	String	类别
alert_type	String	告警类型

表 4-48 network_list

参数	参数类型	描述
direction	String	方向，取值范围：IN | OUT
protocol	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	String	源IP地址
src_port	Integer	源端口，0–65535
src_domain	String	源域名
src_geo	**src_geo** object	源IP的地理位置信息
dest_ip	String	目的IP地址
dest_port	String	目的端口，0–65535
dest_domain	String	目的域名
dest_geo	**dest_geo** object	目标IP的地理位置信息

表 4-49 src_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-50 dest_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-51 resource_list

参数	参数类型	描述
id	String	云服务资源id
name	String	资源名称
type	String	资源类型；引用云RMS type字段
provider	String	云服务名称；引用云RMS provider字段
region_id	String	区域；按照云regionId填写，如cn-north-1等
domain_id	String	资源所属账号ID，UUID格式
project_id	String	资源所属项目ID，UUID格式
ep_id	String	企业项目id
ep_name	String	企业项目名称
tags	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-52 remediation

参数	参数类型	描述
recommendation	String	推荐处理方法
url	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-53 malware

参数	参数类型	描述
malware_family	String	恶意家族
malware_class	String	恶意软件分类

表 4-54 process

参数	参数类型	描述
process_name	String	进程名
process_path	String	进程执行文件路径
process_pid	Integer	进程id
process_uid	Integer	进程用户id
process_cmdline	String	进程命令行
process_parent_name	String	父进程名称
process_parent_path	String	父进程执行文件路径
process_parent_pid	Integer	父进程id
process_parent_uid	Integer	父进程用户id
process_parent_cmdline	String	父进程命令行
process_child_name	String	子进程名称
process_child_path	String	子进程执行文件路径
process_child_pid	Integer	子进程id

参数	参数类型	描述
process_child_uid	Integer	子进程用户id
process_child_cmdline	String	子进程命令行
process_launche_time	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-55 user_info

参数	参数类型	描述
user_id	String	用户uid
user_name	String	用户名称

表 4-56 file_info

参数	参数类型	描述
file_path	String	文件路径/名称
file_content	String	文件内容
file_new_path	String	文件新路径/名称
file_hash	String	文件hash
file_md5	String	文件md5
file_sha256	String	文件sha256
file_attr	String	文件属性

表 4-57 dataclass_ref

参数	参数类型	描述
id	String	数据类唯一标识，UUID格式，最大36个字符
name	String	数据类名称

状态码： 400

表 4-58 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-59 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

创建一条告警，告警名称为MyXXX，标签为MyXXX，URL为http://xxx，发生次数为4次，置信度为4，严重等级为tips。

```
{
  "data_object" : {
    "version" : "1.0",
    "environment" : {
      "vendor_type" : "MyXXX",
      "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    },
    "data_source" : {
      "source_type" : 3,
      "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "product_name" : "test",
      "product_feature" : "test"
    },
    "first_observed_time" : "2021-01-30T23:00:00Z+0800",
    "last_observed_time" : "2021-01-30T23:00:00Z+0800",
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "arrive_time" : "2021-01-30T23:00:00Z+0800",
    "title" : "MyXXX",
    "labels" : "MyXXX",
    "description" : "This my XXXX",
    "source_url" : "http://xxx",
    "count" : 4,
    "confidence" : 4,
    "severity" : "TIPS",
    "criticality" : 4,
    "alert_type" : { },
    "network_list" : [ {
      "direction" : {
        "IN" : null
      },
      "protocol" : "TCP",
      "src_ip" : "192.168.0.1",
      "src_port" : "1",
      "src_domain" : "xxx",
```

```
    "dest_ip" : "192.168.0.1",
    "dest_port" : "1",
    "dest_domain" : "xxx",
    "src_geo" : {
      "latitude" : 90,
      "longitude" : 180
    },
    "dest_geo" : {
      "latitude" : 90,
      "longitude" : 180
    }
  } ],
  "resource_list" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "type" : "MyXXX",
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "ep_name" : "MyXXX",
    "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  } ],
  "remediation" : {
    "recommendation" : "MyXXX",
    "url" : "MyXXX"
  },
  "verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写Unknown",
  "handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
  "sla" : 60000,
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "close_time" : "2021-01-30T23:00:00Z+0800",
  "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
  "simulation" : "false",
  "actor" : "刘一博",
  "owner" : "MyXXX",
  "creator" : "MyXXX",
  "close_reason" : "误检;已解决;重复;其他",
  "close_comment" : "误检;已解决;重复;其他",
  "malware" : {
    "malware_family" : "family",
    "malware_class" : "恶意占用内存"
  },
  "system_info" : { },
  "process" : [ {
    "process_name" : "MyXXX",
    "process_path" : "MyXXX",
    "process_pid" : 123,
    "process_uid" : 123,
    "process_cmdline" : "MyXXX"
  } ],
  "user_info" : [ {
    "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "user_name" : "MyXXX"
  } ],
  "file_info" : [ {
    "file_path" : "MyXXX",
    "file_content" : "MyXXX",
    "file_new_path" : "MyXXX",
    "file_hash" : "MyXXX",
    "file_md5" : "MyXXX",
    "file_sha256" : "MyXXX",
    "file_attr" : "MyXXX"
  } ],
  "system_alert_table" : { },
  "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
```



```
  }
}
```

响应示例

状态码： 200

创建告警返回body体

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "data" : {
    "data_object" : {
      "version" : "1.0",
      "environment" : {
        "vendor_type" : "MyXXX",
        "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
      },
      "data_source" : {
        "source_type" : 3,
        "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
      },
      "first_observed_time" : "2021-01-30T23:00:00Z+0800",
      "last_observed_time" : "2021-01-30T23:00:00Z+0800",
      "create_time" : "2021-01-30T23:00:00Z+0800",
      "arrive_time" : "2021-01-30T23:00:00Z+0800",
      "title" : "MyXXX",
      "description" : "This my XXXX",
      "source_url" : "http://xxx",
      "count" : 4,
      "confidence" : 4,
      "severity" : "TIPS",
      "criticality" : 4,
      "alert_type" : { },
      "network_list" : [ {
        "direction" : {
          "IN" : null
        },
        "protocol" : "TCP",
        "src_ip" : "192.168.0.1",
        "src_port" : "1",
        "src_domain" : "xxx",
        "dest_ip" : "192.168.0.1",
        "dest_port" : "1",
        "dest_domain" : "xxx",
        "src_geo" : {
          "latitude" : 90,
          "longitude" : 180
        },
        "dest_geo" : {
          "latitude" : 90,
          "longitude" : 180
        }
      } ],
      "resource_list" : [ {
        "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "name" : "MyXXX",
        "type" : "MyXXX",
        "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "ep_name" : "MyXXX",
        "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
```

```
    } ],
    "remediation" : {
      "recommendation" : "MyXXX",
      "url" : "MyXXX"
    },
    "verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写
Unknown",
    "handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
    "sla" : 60000,
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "close_time" : "2021-01-30T23:00:00Z+0800",
    "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
    "simulation" : "false",
    "actor" : "刘一博",
    "owner" : "MyXXX",
    "creator" : "MyXXX",
    "close_reason" : "误检;已解决;重复;其他",
    "close_comment" : "误检;已解决;重复;其他",
    "malware" : {
      "malware_family" : "family",
      "malware_class" : "恶意占用内存"
    },
    "system_info" : { },
    "process" : [ {
      "process_name" : "MyXXX",
      "process_path" : "MyXXX",
      "process_pid" : 123,
      "process_uid" : 123,
      "process_cmdline" : "MyXXX"
    } ],
    "user_info" : [ {
      "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "user_name" : "MyXXX"
    } ],
    "file_info" : [ {
      "file_path" : "MyXXX",
      "file_content" : "MyXXX",
      "file_new_path" : "MyXXX",
      "file_hash" : "MyXXX",
      "file_md5" : "MyXXX",
      "file_sha256" : "MyXXX",
      "file_attr" : "MyXXX"
    } ],
    "system_alert_table" : { },
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
  },
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "id" : "MyXXX",
  "version" : 123,
  "format_version" : 123,
  "dataclass_ref" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX"
  }
 }
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一条告警，告警名称为MyXXX，标签为MyXXX，URL为http://xxx，发生次数为4次，置信度为4，严重等级为tips。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class CreateAlertSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateAlertRequest request = new CreateAlertRequest();
        request.withWorkspaceId("{workspace_id}");
        CreateAlertRequestBody body = new CreateAlertRequestBody();
        List<AlertFileInfo> listDataObjectFileInfo = new ArrayList<>();
        listDataObjectFileInfo.add(
            new AlertFileInfo()
                .withFilePath("MyXXX")
                .withFileContent("MyXXX")
                .withFileNewPath("MyXXX")
                .withFileHash("MyXXX")
                .withFileMd5("MyXXX")
                .withFileSha256("MyXXX")
                .withFileAttr("MyXXX")
        );
        List<AlertUserInfo> listDataObjectUserInfo = new ArrayList<>();
        listDataObjectUserInfo.add(
            new AlertUserInfo()
                .withUserId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
                .withUserName("MyXXX")
        );
        List<AlertProcess> listDataObjectProcess = new ArrayList<>();
        listDataObjectProcess.add(
            new AlertProcess()
                .withProcessName("MyXXX")
                .withProcessPath("MyXXX")
                .withProcessPid(123)
                .withProcessUid(123)
                .withProcessCmdline("MyXXX")
        );
```

```
AlertMalware malwareDataObject = new AlertMalware();
malwareDataObject.withMalwareFamily("family")
    .withMalwareClass("恶意占用内存");
AlertRemediation remediationDataObject = new AlertRemediation();
remediationDataObject.withRecommendation("MyXXX")
    .withUrl("MyXXX");
List<AlertResourceList> listDataObjectResourceList = new ArrayList<>();
listDataObjectResourceList.add(
    new AlertResourceList()
        .withId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withName("MyXXX")
        .withType("MyXXX")
        .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withEpId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withEpName("MyXXX")
        .withTags("909494e3-558e-46b6-a9eb-07a8e18ca62f")
);
AlertDestGeo destGeoNetworkList = new AlertDestGeo();
destGeoNetworkList.withLatitude(java.math.BigDecimal.valueOf(90))
    .withLongitude(java.math.BigDecimal.valueOf(180));
AlertSrcGeo srcGeoNetworkList = new AlertSrcGeo();
srcGeoNetworkList.withLatitude(java.math.BigDecimal.valueOf(90))
    .withLongitude(java.math.BigDecimal.valueOf(180));
List<AlertNetworkList> listDataObjectNetworkList = new ArrayList<>();
listDataObjectNetworkList.add(
    new AlertNetworkList()
        .withDirection(AlertNetworkList.DirectionEnum.fromValue("{}"))
        .withProtocol("TCP")
        .withSrcIp("192.168.0.1")
        .withSrcPort(1)
        .withSrcDomain("xxx")
        .withSrcGeo(srcGeoNetworkList)
        .withDestIp("192.168.0.1")
        .withDestPort("1")
        .withDestDomain("xxx")
        .withDestGeo(destGeoNetworkList)
);
AlertDataSource dataSourceDataObject = new AlertDataSource();
dataSourceDataObject.withSourceType(3)
    .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
    .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
    .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
    .withProductName("test")
    .withProductFeature("test");
AlertEnvironment environmentDataObject = new AlertEnvironment();
environmentDataObject.withVendorType("MyXXX")
    .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
    .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
    .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
Alert dataObjectbody = new Alert();
dataObjectbody.withVersion("1.0")
    .withId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
    .withWorkspaceId("909494e3-558e-46b6-a9eb-07a8e18ca620")
    .withLabels("MyXXX")
    .withEnvironment(environmentDataObject)
    .withDataSource(dataSourceDataObject)
    .withFirstObservedTime("2021-01-30T23:00:00Z+0800")
    .withLastObservedTime("2021-01-30T23:00:00Z+0800")
    .withCreateTime("2021-01-30T23:00:00Z+0800")
    .withArriveTime("2021-01-30T23:00:00Z+0800")
    .withTitle("MyXXX")
    .withDescription("This my XXXX")
    .withSourceUrl("http://xxx")
    .withCount(4)
    .withConfidence(4)
    .withSeverity(Alert.SeverityEnum.fromValue("TIPS"))
    .withCriticality(4)
```

```
            .withNetworkList(listDataObjectNetworkList)
            .withResourceList(listDataObjectResourceList)
            .withRemediation(remediationDataObject)
            .withVerificationState(Alert.VerificationStateEnum.fromValue("Unknown – 未知,True_Positive – 确认,False_Positive – 误报。默认填写Unknown"))
            .withHandleStatus(Alert.HandleStatusEnum.fromValue("Open – 打开,Block – 阻塞,Closed – 关闭。默认填写Open"))
            .withSla(60000)
            .withUpdateTime("2021-01-30T23:00:00Z+0800")
            .withCloseTime("2021-01-30T23:00:00Z+0800")
            .withIpdrrPhase(Alert.IpdrrPhaseEnum.fromValue("Prepartion|Detection and Analysis|Containm,Eradication& Recovery| Post-Incident-Activity"))
            .withSimulation("false")
            .withActor("刘一博")
            .withOwner("MyXXX")
            .withCreator("MyXXX")
            .withCloseReason(Alert.CloseReasonEnum.fromValue("误检;已解决;重复;其他"))
            .withCloseComment("误检;已解决;重复;其他")
            .withMalware(malwareDataObject)
            .withSystemInfo(new Object())
            .withProcess(listDataObjectProcess)
            .withUserInfo(listDataObjectUserInfo)
            .withFileInfo(listDataObjectFileInfo)
            .withSystemAlertTable(new Object());
        body.withDataObject(dataObjectbody);
        request.withBody(body);
        try {
            CreateAlertResponse response = client.createAlert(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

创建一条告警，告警名称为MyXXX，标签为MyXXX，URL为http://xxx，发生次数为4次，置信度为4，严重等级为tips。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

    client = SecMasterClient.new_builder() \
```



```
    .with_credentials(credentials) \
    .with_region(SecMasterRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateAlertRequest()
    request.workspace_id = "{workspace_id}"
    listFileInfoDataObject = [
        AlertFileInfo(
            file_path="MyXXX",
            file_content="MyXXX",
            file_new_path="MyXXX",
            file_hash="MyXXX",
            file_md5="MyXXX",
            file_sha256="MyXXX",
            file_attr="MyXXX"
        )
    ]
    listUserInfoDataObject = [
        AlertUserInfo(
            user_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            user_name="MyXXX"
        )
    ]
    listProcessDataObject = [
        AlertProcess(
            process_name="MyXXX",
            process_path="MyXXX",
            process_pid=123,
            process_uid=123,
            process_cmdline="MyXXX"
        )
    ]
    malwareDataObject = AlertMalware(
        malware_family="family",
        malware_class="恶意占用内存"
    )
    remediationDataObject = AlertRemediation(
        recommendation="MyXXX",
        url="MyXXX"
    )
    listResourceListDataObject = [
        AlertResourceList(
            id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            name="MyXXX",
            type="MyXXX",
            region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            ep_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            ep_name="MyXXX",
            tags="909494e3-558e-46b6-a9eb-07a8e18ca62f"
        )
    ]
    destGeoNetworkList = AlertDestGeo(
        latitude=90,
        longitude=180
    )
    srcGeoNetworkList = AlertSrcGeo(
        latitude=90,
        longitude=180
    )
    listNetworkListDataObject = [
        AlertNetworkList(
            direction="{}",
            protocol="TCP",
            src_ip="192.168.0.1",
            src_port=1,
            src_domain="xxx",
```

```
            src_geo=srcGeoNetworkList,
            dest_ip="192.168.0.1",
            dest_port="1",
            dest_domain="xxx",
            dest_geo=destGeoNetworkList
        )
    ]
    dataSourceDataObject = AlertDataSource(
        source_type=3,
        domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        product_name="test",
        product_feature="test"
    )
    environmentDataObject = AlertEnvironment(
        vendor_type="MyXXX",
        domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f"
    )
    dataObjectbody = Alert(
        version="1.0",
        id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        workspace_id="909494e3-558e-46b6-a9eb-07a8e18ca620",
        labels="MyXXX",
        environment=environmentDataObject,
        data_source=dataSourceDataObject,
        first_observed_time="2021-01-30T23:00:00Z+0800",
        last_observed_time="2021-01-30T23:00:00Z+0800",
        create_time="2021-01-30T23:00:00Z+0800",
        arrive_time="2021-01-30T23:00:00Z+0800",
        title="MyXXX",
        description="This my XXXX",
        source_url="http://xxx",
        count=4,
        confidence=4,
        severity="TIPS",
        criticality=4,
        network_list=listNetworkListDataObject,
        resource_list=listResourceListDataObject,
        remediation=remediationDataObject,
        verification_state="Unknown – 未知,True_Positive – 确认,False_Positive – 误报。默认填写Unknown",
        handle_status="Open – 打开,Block – 阻塞,Closed – 关闭。默认填写Open",
        sla=60000,
        update_time="2021-01-30T23:00:00Z+0800",
        close_time="2021-01-30T23:00:00Z+0800",
        ipdrr_phase="Prepartion|Detection and Analysis|Containm,Eradication& Recovery| Post-Incident-
Activity",
        simulation="false",
        actor="刘一博",
        owner="MyXXX",
        creator="MyXXX",
        close_reason="误检;已解决;重复;其他",
        close_comment="误检;已解决;重复;其他",
        malware=malwareDataObject,
        system_info={},
        process=listProcessDataObject,
        user_info=listUserInfoDataObject,
        file_info=listFileInfoDataObject,
        system_alert_table={}
    )
    request.body = CreateAlertRequestBody(
        data_object=dataObjectbody
    )
    response = client.create_alert(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

创建一条告警，告警名称为MyXXX，标签为MyXXX，URL为http://xxx，发生次数为4次，置信度为4，严重等级为tips。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateAlertRequest{}
    request.WorkspaceId = "{workspace_id}"
    filePathFileInfo:= "MyXXX"
    fileContentFileInfo:= "MyXXX"
    fileNewPathFileInfo:= "MyXXX"
    fileHashFileInfo:= "MyXXX"
    fileMd5FileInfo:= "MyXXX"
    fileSha256FileInfo:= "MyXXX"
    fileAttrFileInfo:= "MyXXX"
    var listFileInfoDataObject = []model.AlertFileInfo{
        {
            FilePath: &filePathFileInfo,
            FileContent: &fileContentFileInfo,
            FileNewPath: &fileNewPathFileInfo,
            FileHash: &fileHashFileInfo,
            FileMd5: &fileMd5FileInfo,
            FileSha256: &fileSha256FileInfo,
            FileAttr: &fileAttrFileInfo,
        },
    }
    userIdUserInfo:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    userNameUserInfo:= "MyXXX"
    var listUserInfoDataObject = []model.AlertUserInfo{
        {
            UserId: &userIdUserInfo,
            UserName: &userNameUserInfo,
        },
    }
```

```
processNameProcess:= "MyXXX"
processPathProcess:= "MyXXX"
processPidProcess:= int32(123)
processUidProcess:= int32(123)
processCmdlineProcess:= "MyXXX"
var listProcessDataObject = []model.AlertProcess{
    {
        ProcessName: &processNameProcess,
        ProcessPath: &processPathProcess,
        ProcessPid: &processPidProcess,
        ProcessUid: &processUidProcess,
        ProcessCmdline: &processCmdlineProcess,
    },
}
malwareFamilyMalware:= "family"
malwareClassMalware:= "恶意占用内存"
malwareDataObject := &model.AlertMalware{
    MalwareFamily: &malwareFamilyMalware,
    MalwareClass: &malwareClassMalware,
}
recommendationRemediation:= "MyXXX"
urlRemediation:= "MyXXX"
remediationDataObject := &model.AlertRemediation{
    Recommendation: &recommendationRemediation,
    Url: &urlRemediation,
}
idResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
nameResourceList:= "MyXXX"
typeResourceList:= "MyXXX"
regionIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
domainIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
projectIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
epIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
epNameResourceList:= "MyXXX"
tagsResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
var listResourceListDataObject = []model.AlertResourceList{
    {
        Id: &idResourceList,
        Name: &nameResourceList,
        Type: &typeResourceList,
        RegionId: &regionIdResourceList,
        DomainId: &domainIdResourceList,
        ProjectId: &projectIdResourceList,
        EpId: &epIdResourceList,
        EpName: &epNameResourceList,
        Tags: &tagsResourceList,
    },
}
latitudeDestGeo:= float32(90)
longitudeDestGeo:= float32(180)
destGeoNetworkList := &model.AlertDestGeo{
    Latitude: &latitudeDestGeo,
    Longitude: &longitudeDestGeo,
}
latitudeSrcGeo:= float32(90)
longitudeSrcGeo:= float32(180)
srcGeoNetworkList := &model.AlertSrcGeo{
    Latitude: &latitudeSrcGeo,
    Longitude: &longitudeSrcGeo,
}
directionNetworkList:= model.GetAlertNetworkListDirectionEnum().{}
protocolNetworkList:= "TCP"
srcIpNetworkList:= "192.168.0.1"
srcPortNetworkList:= int32(1)
srcDomainNetworkList:= "xxx"
destIpNetworkList:= "192.168.0.1"
destPortNetworkList:= "1"
destDomainNetworkList:= "xxx"
var listNetworkListDataObject = []model.AlertNetworkList{
```

```
        {
            Direction: &directionNetworkList,
            Protocol: &protocolNetworkList,
            SrcIp: &srcIpNetworkList,
            SrcPort: &srcPortNetworkList,
            SrcDomain: &srcDomainNetworkList,
            SrcGeo: srcGeoNetworkList,
            DestIp: &destIpNetworkList,
            DestPort: &destPortNetworkList,
            DestDomain: &destDomainNetworkList,
            DestGeo: destGeoNetworkList,
        },
    }
    sourceTypeDataSource:= int32(3)
    domainIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    projectIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    regionIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    productNameDataSource:= "test"
    productFeatureDataSource:= "test"
    dataSourceDataObject := &model.AlertDataSource{
        SourceType: &sourceTypeDataSource,
        DomainId: &domainIdDataSource,
        ProjectId: &projectIdDataSource,
        RegionId: &regionIdDataSource,
        ProductName: &productNameDataSource,
        ProductFeature: &productFeatureDataSource,
    }
    vendorTypeEnvironment:= "MyXXX"
    domainIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    regionIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    projectIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    environmentDataObject := &model.AlertEnvironment{
        VendorType: &vendorTypeEnvironment,
        DomainId: &domainIdEnvironment,
        RegionId: &regionIdEnvironment,
        ProjectId: &projectIdEnvironment,
    }
    versionDataObject:= "1.0"
    idDataObject:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    workspaceIdDataObject:= "909494e3-558e-46b6-a9eb-07a8e18ca620"
    labelsDataObject:= "MyXXX"
    firstObservedTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    lastObservedTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    createTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    arriveTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    titleDataObject:= "MyXXX"
    descriptionDataObject:= "This my XXXX"
    sourceUrlDataObject:= "http://xxx"
    countDataObject:= int32(4)
    confidenceDataObject:= int32(4)
    severityDataObject:= model.GetAlertSeverityEnum().TIPS
    criticalityDataObject:= int32(4)
    verificationStateDataObject:= model.GetAlertVerificationStateEnum().UNKNOWN_-_未
知,TRUE_POSITIVE_-_确认,FALSE_POSITIVE_-_误报。默认填写UNKNOWN
    handleStatusDataObject:= model.GetAlertHandleStatusEnum().OPEN_-_打开,BLOCK_-_阻塞,CLOSED_-_关
闭。默认填写OPEN
    slaDataObject:= int32(60000)
    updateTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    closeTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    ipdrrPhaseDataObject:= model.GetAlertIpdrrPhaseEnum().PREPARTION|DETECTION_AND_ANALYSIS|
CONTAINM,ERADICATION&_RECOVERY|_POST_INCIDENT_ACTIVITY
    simulationDataObject:= "false"
    actorDataObject:= "刘一博"
    ownerDataObject:= "MyXXX"
    creatorDataObject:= "MyXXX"
    closeReasonDataObject:= model.GetAlertCloseReasonEnum().误检;已解决;重复;其他
    closeCommentDataObject:= "误检;已解决;重复;其他"
    var systemInfoDataObject interface{} = make(map[string]string)
    var systemAlertTableDataObject interface{} = make(map[string]string)
```

```
    dataObjectbody := &model.Alert{
        Version: &versionDataObject,
        Id: &idDataObject,
        WorkspaceId: &workspaceIdDataObject,
        Labels: &labelsDataObject,
        Environment: environmentDataObject,
        DataSource: dataSourceDataObject,
        FirstObservedTime: &firstObservedTimeDataObject,
        LastObservedTime: &lastObservedTimeDataObject,
        CreateTime: &createTimeDataObject,
        ArriveTime: &arriveTimeDataObject,
        Title: &titleDataObject,
        Description: &descriptionDataObject,
        SourceUrl: &sourceUrlDataObject,
        Count: &countDataObject,
        Confidence: &confidenceDataObject,
        Severity: &severityDataObject,
        Criticality: &criticalityDataObject,
        NetworkList: &listNetworkListDataObject,
        ResourceList: &listResourceListDataObject,
        Remediation: remediationDataObject,
        VerificationState: &verificationStateDataObject,
        HandleStatus: &handleStatusDataObject,
        Sla: &slaDataObject,
        UpdateTime: &updateTimeDataObject,
        CloseTime: &closeTimeDataObject,
        IpdrrPhase: &ipdrrPhaseDataObject,
        Simulation: &simulationDataObject,
        Actor: &actorDataObject,
        Owner: &ownerDataObject,
        Creator: &creatorDataObject,
        CloseReason: &closeReasonDataObject,
        CloseComment: &closeCommentDataObject,
        Malware: malwareDataObject,
        SystemInfo: &systemInfoDataObject,
        Process: &listProcessDataObject,
        UserInfo: &listUserInfoDataObject,
        FileInfo: &listFileInfoDataObject,
        SystemAlertTable: &systemAlertTableDataObject,
    }
    request.Body = &model.CreateAlertRequestBody{
        DataObject: dataObjectbody,
    }
    response, err := client.CreateAlert(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	创建告警返回body体
400	创建告警错误返回body体

错误码

请参见**错误码**。

4.1.3 删除告警

功能介绍

删除告警

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/workspaces/{workspace_id}/soc/alerts

表 4-60 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

请求参数

表 4-61 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）
content-type	是	String	内容类型

表 4-62 请求 Body 参数

参数	是否必选	参数类型	描述
batch_ids	否	Array of strings	删除告警的ID列表

响应参数

状态码： 200

表 4-63 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-64 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**BatchOperateAlertResult** object	批量操作告警返回对象

表 4-65 BatchOperateAlertResult

参数	参数类型	描述
error_ids	Array of strings	失败id
success_ids	Array of strings	成功id

状态码： 400

表 4-66 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-67 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

删除id为909494e3-558e-46b6-a9eb-07a8e18ca621的告警

```
{
  "batch_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ]
}
```

响应示例

状态码： 200

删除告警返回body体

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "data" : {
    "success_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ],
    "error_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ]
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

删除id为909494e3-558e-46b6-a9eb-07a8e18ca621的告警

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class DeleteAlertSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
```

```
        DeleteAlertRequest request = new DeleteAlertRequest();
        request.withWorkspaceId("{workspace_id}");
        DeleteAlertRequestBody body = new DeleteAlertRequestBody();
        List<String> listbodyBatchIds = new ArrayList<>();
        listbodyBatchIds.add("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        body.withBatchIds(listbodyBatchIds);
        request.withBody(body);
        try {
            DeleteAlertResponse response = client.deleteAlert(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

删除id为909494e3-558e-46b6-a9eb-07a8e18ca621的告警

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteAlertRequest()
        request.workspace_id = "{workspace_id}"
        listBatchIdsbody = [
            "909494e3-558e-46b6-a9eb-07a8e18ca62f"
        ]
        request.body = DeleteAlertRequestBody(
            batch_ids=listBatchIdsbody
        )
        response = client.delete_alert(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

删除id为909494e3-558e-46b6-a9eb-07a8e18ca621的告警

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteAlertRequest{}
    request.WorkspaceId = "{workspace_id}"
    var listBatchIdsbody = []string{
        "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    }
    request.Body = &model.DeleteAlertRequestBody{
        BatchIds: &listBatchIdsbody,
    }
    response, err := client.DeleteAlert(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	删除告警返回body体
400	删除告警错误返回body体

错误码

请参见**错误码**。

4.1.4 告警转事件

功能介绍

告警转事件

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/alerts/batch-order

表 4-68 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

请求参数

表 4-69 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）
content-type	是	String	内容类型

表 4-70 请求 Body 参数

参数	是否必选	参数类型	描述
ids	否	Array of strings	转事件的告警id列表
incident_content	否	**incident_content** object	事件内容

表 4-71 incident_content

参数	是否必选	参数类型	描述
title	否	String	事件名称
incident_type	否	**incident_type** object	事件类型

表 4-72 incident_type

参数	是否必选	参数类型	描述
id	否	String	事件类型id
category	否	String	事件类型父类
incident_type	否	String	事件类型子类

响应参数

状态码： 200

表 4-73 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-74 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**BatchOperateAlertResult** object	批量操作告警返回对象

表 4-75 BatchOperateAlertResult

参数	参数类型	描述
error_ids	Array of strings	失败id
success_ids	Array of strings	成功id

状态码： 400

表 4-76 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-77 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

将一条告警转为事件，告警ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，事件ID为909494e3-558e-46b6-a9eb-07a8e18ca621，告警状态为已关闭，是否标记为证据为否。

```
{
  "ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ],
  "incident_content" : {
    "title" : "XXX",
    "incident_type" : {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "category" : "DDoS攻击",
      "incident_type" : "DNS协议攻击"
    }
  }
}
```

响应示例

状态码： 200

告警转事件返回body体

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "data" : {
    "error_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ],
    "success_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ]
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

将一条告警转为事件，告警ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，事件ID为909494e3-558e-46b6-a9eb-07a8e18ca621，告警状态为已关闭，是否标记为证据为否。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class CreateBatchOrderAlertsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateBatchOrderAlertsRequest request = new CreateBatchOrderAlertsRequest();
        request.withWorkspaceId("{workspace_id}");
        OrderAlert body = new OrderAlert();
        OrderAlertIncidentContentIncidentType incidentTypeIncidentContent = new OrderAlertIncidentContentIncidentType();
        incidentTypeIncidentContent.withId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withCategory("DDoS攻击")
            .withIncidentType("DNS协议攻击");
        OrderAlertIncidentContent incidentContentbody = new OrderAlertIncidentContent();
        incidentContentbody.withTitle("XXX")
            .withIncidentType(incidentTypeIncidentContent);
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        body.withIncidentContent(incidentContentbody);
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            CreateBatchOrderAlertsResponse response = client.createBatchOrderAlerts(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

将一条告警转为事件，告警ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，事件ID为909494e3-558e-46b6-a9eb-07a8e18ca621，告警状态为已关闭，是否标记为证据为否。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateBatchOrderAlertsRequest()
        request.workspace_id = "{workspace_id}"
        incidentTypeIncidentContent = OrderAlertIncidentContentIncidentType(
            id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            category="DDoS攻击",
            incident_type="DNS协议攻击"
        )
        incidentContentbody = OrderAlertIncidentContent(
            title="XXX",
            incident_type=incidentTypeIncidentContent
        )
        listIdsbody = [
            "909494e3-558e-46b6-a9eb-07a8e18ca62f"
        ]
        request.body = OrderAlert(
            incident_content=incidentContentbody,
            ids=listIdsbody
        )
        response = client.create_batch_order_alerts(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

将一条告警转为事件，告警ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，事件ID为909494e3-558e-46b6-a9eb-07a8e18ca621，告警状态为已关闭，是否标记为证据为否。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateBatchOrderAlertsRequest{}
    request.WorkspaceId = "{workspace_id}"
    idIncidentType:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    categoryIncidentType:= "DDoS攻击"
    incidentTypeIncidentType:= "DNS协议攻击"
    incidentTypeIncidentContent := &model.OrderAlertIncidentContentIncidentType{
        Id: &idIncidentType,
        Category: &categoryIncidentType,
        IncidentType: &incidentTypeIncidentType,
    }
    titleIncidentContent:= "XXX"
    incidentContentbody := &model.OrderAlertIncidentContent{
        Title: &titleIncidentContent,
        IncidentType: incidentTypeIncidentContent,
    }
    var listIdsbody = []string{
        "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    }
    request.Body = &model.OrderAlert{
        IncidentContent: incidentContentbody,
        Ids: &listIdsbody,
    }
    response, err := client.CreateBatchOrderAlerts(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	告警转事件返回body体
400	告警转事件错误返回body体

错误码

请参见**错误码**。

4.1.5 获取告警详情

功能介绍

获取告警详情

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/alerts/{alert_id}

表 4-78 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id
alert_id	是	String	告警ID

请求参数

表 4-79 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）
content-type	是	String	内容类型

响应参数

状态码： 200

表 4-80 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-81 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**AlertDetail** object	告警详情对象

表 4-82 AlertDetail

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
data_object	**Alert** object	告警实体信息
dataclass_ref	**dataclass_ref** object	数据类对象
format_version	Integer	格式版本

参数	参数类型	描述
id	String	事件唯一标识，UUID格式，最大36个字符
type	String	数据类型
project_id	String	当前项目的id
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
version	Integer	版本
workspace_id	String	当前的工作空间id

表 4-83 Alert

参数	参数类型	描述
version	String	告警对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	String	事件唯一标识，UUID格式，最大36个字符
domain_id	String	数据投递后，被委托用户的domain_id
region_id	String	数据投递后，被委托用户的region_id
workspace_id	String	当前的工作空间id
labels	String	标签，仅展示
environment	**environment** object	告警产生的环境坐标信息
data_source	**data_source** object	首次上报数据源
first_observed_time	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	String	告警标题
description	String	告警描述信息
source_url	String	告警URL链接，指向数据源产品中有关当前事件说明的页面
count	Integer	事件发生次数
confidence	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%
severity	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源
alert_type	**alert_type** object	告警分类，详细定义参考《告警类型定义》
network_list	Array of **network_list** objects	网络信息
resource_list	Array of **resource_list** objects	受影响资源

参数	参数类型	描述
remediation	**remediation** object	补救措施
verification_state	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	Integer	约束闭环时间：设置风险接受持续时间。单位：小时
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	String	调试字段
actor	String	告警调查员
owner	String	责任人、服务责任人
creator	String	创建人
close_reason	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	String	关闭评论

参数	参数类型	描述
malware	**malware** object	恶意软件
system_info	Object	系统信息
process	Array of **process** objects	进程信息
user_info	Array of **user_info** objects	用户信息
file_info	Array of **file_info** objects	文件信息
system_alert_table	Object	告警管理列表的布局字段

表 4-84 environment

参数	参数类型	描述
vendor_type	String	环境供应商
domain_id	String	租户id
region_id	String	区域id，全局服务global
cross_workspace_id	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	String	项目id， 全局服务默认null

表 4-85 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下： 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	数据源产品所属账号的id
project_id	String	数据源产品所属项目的id
region_id	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	String	数据源产品所属公司的名称
product_name	String	数据源产品的名称

参数	参数类型	描述
product_feature	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	String	检测模块列表

表 4-86 alert_type

参数	参数类型	描述
category	String	类别
alert_type	String	告警类型

表 4-87 network_list

参数	参数类型	描述
direction	String	方向，取值范围：IN | OUT
protocol	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	String	源IP地址
src_port	Integer	源端口，0–65535
src_domain	String	源域名
src_geo	**src_geo** object	源IP的地理位置信息
dest_ip	String	目的IP地址
dest_port	String	目的端口，0–65535
dest_domain	String	目的域名
dest_geo	**dest_geo** object	目标IP的地理位置信息

表 4-88 src_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度

参数	参数类型	描述
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-89 dest_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-90 resource_list

参数	参数类型	描述
id	String	云服务资源id
name	String	资源名称
type	String	资源类型；引用云RMS type字段
provider	String	云服务名称；引用云RMS provider字段
region_id	String	区域；按照云regionId填写，如cn-north-1等
domain_id	String	资源所属账号ID，UUID格式
project_id	String	资源所属项目ID，UUID格式
ep_id	String	企业项目id
ep_name	String	企业项目名称
tags	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-91 remediation

参数	参数类型	描述
recommendation	String	推荐处理方法
url	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-92 malware

参数	参数类型	描述
malware_family	String	恶意家族
malware_class	String	恶意软件分类

表 4-93 process

参数	参数类型	描述
process_name	String	进程名
process_path	String	进程执行文件路径
process_pid	Integer	进程id
process_uid	Integer	进程用户id
process_cmdline	String	进程命令行
process_parent_name	String	父进程名称
process_parent_path	String	父进程执行文件路径
process_parent_pid	Integer	父进程id
process_parent_uid	Integer	父进程用户id
process_parent_cmdline	String	父进程命令行
process_child_name	String	子进程名称
process_child_path	String	子进程执行文件路径
process_child_pid	Integer	子进程id

参数	参数类型	描述
process_child_uid	Integer	子进程用户id
process_child_cmdline	String	子进程命令行
process_launche_time	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-94 user_info

参数	参数类型	描述
user_id	String	用户uid
user_name	String	用户名称

表 4-95 file_info

参数	参数类型	描述
file_path	String	文件路径/名称
file_content	String	文件内容
file_new_path	String	文件新路径/名称
file_hash	String	文件hash
file_md5	String	文件md5
file_sha256	String	文件sha256
file_attr	String	文件属性

表 4-96 dataclass_ref

参数	参数类型	描述
id	String	数据类唯一标识，UUID格式，最大36个字符
name	String	数据类名称

状态码： 400

表 4-97 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-98 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

获取告警详情返回body体

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "data" : {
    "data_object" : {
      "version" : "1.0",
      "environment" : {
        "vendor_type" : "MyXXX",
        "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
      },
      "data_source" : {
        "source_type" : 3,
        "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
      },
      "first_observed_time" : "2021-01-30T23:00:00Z+0800",
      "last_observed_time" : "2021-01-30T23:00:00Z+0800",
      "create_time" : "2021-01-30T23:00:00Z+0800",
      "arrive_time" : "2021-01-30T23:00:00Z+0800",
      "title" : "MyXXX",
      "description" : "This my XXXX",
      "source_url" : "http://xxx",
      "count" : "4",
      "confidence" : 4,
      "severity" : "TIPS",
      "criticality" : 4,
      "alert_type" : { },
      "network_list" : [ {
        "direction" : {
```

```
      "IN" : null
    },
    "protocol" : "TCP",
    "src_ip" : "192.168.0.1",
    "src_port" : "1",
    "src_domain" : "xxx",
    "dest_ip" : "192.168.0.1",
    "dest_port" : "1",
    "dest_domain" : "xxx",
    "src_geo" : {
      "latitude" : 90,
      "longitude" : 180
    },
    "dest_geo" : {
      "latitude" : 90,
      "longitude" : 180
    }
  } ],
  "resource_list" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "type" : "MyXXX",
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "ep_name" : "MyXXX",
    "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  } ],
  "remediation" : {
    "recommendation" : "MyXXX",
    "url" : "MyXXX"
  },
  "verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写
Unknown",
  "handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
  "sla" : 60000,
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "close_time" : "2021-01-30T23:00:00Z+0800",
  "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
  "simulation" : "false",
  "actor" : "刘一博",
  "owner" : "MyXXX",
  "creator" : "MyXXX",
  "close_reason" : "误检;已解决;重复;其他",
  "close_comment" : "误检;已解决;重复;其他",
  "malware" : {
    "malware_family" : "family",
    "malware_class" : "恶意占用内存"
  },
  "system_info" : { },
  "process" : [ {
    "process_name" : "MyXXX",
    "process_path" : "MyXXX",
    "process_pid" : 123,
    "process_uid" : 123,
    "process_cmdline" : "MyXXX"
  } ],
  "user_info" : [ {
    "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "user_name" : "MyXXX"
  } ],
  "file_info" : [ {
    "file_path" : "MyXXX",
    "file_content" : "MyXXX",
    "file_new_path" : "MyXXX",
    "file_hash" : "MyXXX",
    "file_md5" : "MyXXX",
```

```
      "file_sha256" : "MyXXX",
      "file_attr" : "MyXXX"
    } ],
    "system_alert_table" : { },
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
  },
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "id" : "MyXXX",
  "version" : 11,
  "format_version" : 11,
  "dataclass_ref" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX"
  }
 }
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowAlertSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowAlertRequest request = new ShowAlertRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withAlertId("{alert_id}");
        try {
            ShowAlertResponse response = client.showAlert(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowAlertRequest()
        request.workspace_id = "{workspace_id}"
        request.alert_id = "{alert_id}"
        response = client.show_alert(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

```
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAlertRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.AlertId = "{alert_id}"
    response, err := client.ShowAlert(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	获取告警详情返回body体
400	获取告警详情错误返回body体

错误码

请参见**错误码**。

4.1.6 更新告警

功能介绍

编辑告警，根据实际修改的属性更新，未修改的列不更新

调用方法

请参见**如何调用API**。

URI

PUT /v1/{project_id}/workspaces/{workspace_id}/soc/alerts/{alert_id}

表 4-99 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id
alert_id	是	String	告警ID

请求参数

表 4-100 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）
content-type	是	String	内容类型

表 4-101 请求 Body 参数

参数	是否必选	参数类型	描述
batch_ids	否	Array of strings	更新告警的ID列表
data_object	否	**Alert** object	告警实体信息

表 4-102 Alert

参数	是否必选	参数类型	描述
version	否	String	告警对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	否	String	事件唯一标识，UUID格式，最大36个字符
domain_id	否	String	数据投递后，被委托用户的domain_id
region_id	否	String	数据投递后，被委托用户的region_id
workspace_id	否	String	当前的工作空间id

参数	是否必选	参数类型	描述
labels	否	String	标签，仅展示
environment	否	**environment** object	告警产生的环境坐标信息
data_source	否	**data_source** object	首次上报数据源
first_observed_time	否	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	否	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
create_time	否	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	否	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	否	String	告警标题
description	否	String	告警描述信息
source_url	否	String	告警URL链接，指向数据源产品中有关当前事件说明的页面
count	否	Integer	事件发生次数
confidence	否	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%

参数	是否必选	参数类型	描述
severity	否	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	否	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源
alert_type	否	**alert_type** object	告警分类，详细定义参考《告警类型定义》
network_list	否	Array of **network_list** objects	网络信息
resource_list	否	Array of **resource_list** objects	受影响资源
remediation	否	**remediation** object	补救措施
verification_state	否	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	否	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	否	Integer	约束闭环时间：设置风险接受持续时间。单位：小时

参数	是否必选	参数类型	描述
update_time	否	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	否	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	否	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	否	String	调试字段
actor	否	String	告警调查员
owner	否	String	责任人、服务责任人
creator	否	String	创建人
close_reason	否	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	否	String	关闭评论
malware	否	**malware** object	恶意软件
system_info	否	Object	系统信息
process	否	Array of **process** objects	进程信息
user_info	否	Array of **user_info** objects	用户信息
file_info	否	Array of **file_info** objects	文件信息

参数	是否必选	参数类型	描述
system_alert_table	否	Object	告警管理列表的布局字段

表 4-103 environment

参数	是否必选	参数类型	描述
vendor_type	否	String	环境供应商
domain_id	否	String	租户id
region_id	否	String	区域id，全局服务global
cross_workspace_id	否	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	否	String	项目id， 全局服务默认null

表 4-104 data_source

参数	是否必选	参数类型	描述
source_type	否	Integer	数据源类型，取值范围如下： 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	否	String	数据源产品所属账号的id
project_id	否	String	数据源产品所属项目的id
region_id	否	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	否	String	数据源产品所属公司的名称
product_name	否	String	数据源产品的名称
product_feature	否	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	否	String	检测模块列表

表 4-105 alert_type

参数	是否必选	参数类型	描述
category	否	String	类别
alert_type	否	String	告警类型

表 4-106 network_list

参数	是否必选	参数类型	描述
direction	否	String	方向，取值范围：IN | OUT
protocol	否	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	否	String	源IP地址
src_port	否	Integer	源端口，0–65535
src_domain	否	String	源域名
src_geo	否	src_geo object	源IP的地理位置信息
dest_ip	否	String	目的IP地址
dest_port	否	String	目的端口，0–65535
dest_domain	否	String	目的域名
dest_geo	否	dest_geo object	目标IP的地理位置信息

表 4-107 src_geo

参数	是否必选	参数类型	描述
latitude	否	Number	纬度
longitude	否	Number	经度
city_code	否	String	城市编码，Beijing | Shanghai
country_code	否	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-108 dest_geo

参数	是否必选	参数类型	描述
latitude	否	Number	纬度
longitude	否	Number	经度
city_code	否	String	城市编码，Beijing | Shanghai
country_code	否	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-109 resource_list

参数	是否必选	参数类型	描述
id	否	String	云服务资源id
name	否	String	资源名称
type	否	String	资源类型；引用云RMS type字段
provider	否	String	云服务名称；引用云RMS provider字段
region_id	否	String	区域；按照云regionId填写，如cn-north-1等
domain_id	否	String	资源所属账号ID，UUID格式
project_id	否	String	资源所属项目ID，UUID格式
ep_id	否	String	企业项目id
ep_name	否	String	企业项目名称
tags	否	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-110 remediation

参数	是否必选	参数类型	描述
recommendation	否	String	推荐处理方法

参数	是否必选	参数类型	描述
url	否	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-111 malware

参数	是否必选	参数类型	描述
malware_family	否	String	恶意家族
malware_class	否	String	恶意软件分类

表 4-112 process

参数	是否必选	参数类型	描述
process_name	否	String	进程名
process_path	否	String	进程执行文件路径
process_pid	否	Integer	进程id
process_uid	否	Integer	进程用户id
process_cmdline	否	String	进程命令行
process_parent_name	否	String	父进程名称
process_parent_path	否	String	父进程执行文件路径
process_parent_pid	否	Integer	父进程id
process_parent_uid	否	Integer	父进程用户id
process_parent_cmdline	否	String	父进程命令行
process_child_name	否	String	子进程名称
process_child_path	否	String	子进程执行文件路径
process_child_pid	否	Integer	子进程id

参数	是否必选	参数类型	描述
process_child_uid	否	Integer	子进程用户id
process_child_cmdline	否	String	子进程命令行
process_launche_time	否	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	否	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-113 user_info

参数	是否必选	参数类型	描述
user_id	否	String	用户uid
user_name	否	String	用户名称

表 4-114 file_info

参数	是否必选	参数类型	描述
file_path	否	String	文件路径/名称
file_content	否	String	文件内容
file_new_path	否	String	文件新路径/名称
file_hash	否	String	文件hash
file_md5	否	String	文件md5
file_sha256	否	String	文件sha256
file_attr	否	String	文件属性

响应参数

状态码： 200

表 4-115 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-116 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**AlertDetail** object	告警详情对象

表 4-117 AlertDetail

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
data_object	**Alert** object	告警实体信息
dataclass_ref	**dataclass_ref** object	数据类对象
format_version	Integer	格式版本
id	String	事件唯一标识，UUID格式，最大36个字符
type	String	数据类型
project_id	String	当前项目的id
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
version	Integer	版本
workspace_id	String	当前的工作空间id

表 4-118 Alert

参数	参数类型	描述
version	String	告警对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	String	事件唯一标识，UUID格式，最大36个字符
domain_id	String	数据投递后，被委托用户的domain_id
region_id	String	数据投递后，被委托用户的region_id
workspace_id	String	当前的工作空间id
labels	String	标签，仅展示
environment	**environment** object	告警产生的环境坐标信息
data_source	**data_source** object	首次上报数据源
first_observed_time	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	String	告警标题
description	String	告警描述信息
source_url	String	告警URL链接，指向数据源产品中有关当前事件说明的页面
count	Integer	事件发生次数
confidence	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%

参数	参数类型	描述
severity	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源
alert_type	**alert_type** object	告警分类，详细定义参考《告警类型定义》
network_list	Array of **network_list** objects	网络信息
resource_list	Array of **resource_list** objects	受影响资源
remediation	**remediation** object	补救措施
verification_state	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	Integer	约束闭环时间：设置风险接受持续时间。单位：小时

参数	参数类型	描述
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	String	调试字段
actor	String	告警调查员
owner	String	责任人、服务责任人
creator	String	创建人
close_reason	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	String	关闭评论
malware	**malware** object	恶意软件
system_info	Object	系统信息
process	Array of **process** objects	进程信息
user_info	Array of **user_info** objects	用户信息
file_info	Array of **file_info** objects	文件信息
system_alert_table	Object	告警管理列表的布局字段

表 4-119 environment

参数	参数类型	描述
vendor_type	String	环境供应商

参数	参数类型	描述
domain_id	String	租户id
region_id	String	区域id，全局服务global
cross_workspace_id	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	String	项目id， 全局服务默认null

表 4-120 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下： 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	数据源产品所属账号的id
project_id	String	数据源产品所属项目的id
region_id	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	String	数据源产品所属公司的名称
product_name	String	数据源产品的名称
product_feature	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	String	检测模块列表

表 4-121 alert_type

参数	参数类型	描述
category	String	类别
alert_type	String	告警类型

表 4-122 network_list

参数	参数类型	描述
direction	String	方向，取值范围：IN | OUT

参数	参数类型	描述
protocol	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	String	源IP地址
src_port	Integer	源端口，0–65535
src_domain	String	源域名
src_geo	**src_geo** object	源IP的地理位置信息
dest_ip	String	目的IP地址
dest_port	String	目的端口，0–65535
dest_domain	String	目的域名
dest_geo	**dest_geo** object	目标IP的地理位置信息

表 4-123 src_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-124 dest_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-125 resource_list

参数	参数类型	描述
id	String	云服务资源id
name	String	资源名称
type	String	资源类型；引用云RMS type字段
provider	String	云服务名称；引用云RMS provider字段
region_id	String	区域；按照云regionId填写，如cn-north-1等
domain_id	String	资源所属账号ID，UUID格式
project_id	String	资源所属项目ID，UUID格式
ep_id	String	企业项目id
ep_name	String	企业项目名称
tags	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-126 remediation

参数	参数类型	描述
recommendation	String	推荐处理方法
url	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-127 malware

参数	参数类型	描述
malware_family	String	恶意家族
malware_class	String	恶意软件分类

表 4-128 process

参数	参数类型	描述
process_name	String	进程名

参数	参数类型	描述
process_path	String	进程执行文件路径
process_pid	Integer	进程id
process_uid	Integer	进程用户id
process_cmdline	String	进程命令行
process_parent_name	String	父进程名称
process_parent_path	String	父进程执行文件路径
process_parent_pid	Integer	父进程id
process_parent_uid	Integer	父进程用户id
process_parent_cmdline	String	父进程命令行
process_child_name	String	子进程名称
process_child_path	String	子进程执行文件路径
process_child_pid	Integer	子进程id
process_child_uid	Integer	子进程用户id
process_child_cmdline	String	子进程命令行
process_launche_time	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-129 user_info

参数	参数类型	描述
user_id	String	用户uid
user_name	String	用户名称

表 4-130 file_info

参数	参数类型	描述
file_path	String	文件路径/名称
file_content	String	文件内容
file_new_path	String	文件新路径/名称
file_hash	String	文件hash
file_md5	String	文件md5
file_sha256	String	文件sha256
file_attr	String	文件属性

表 4-131 dataclass_ref

参数	参数类型	描述
id	String	数据类唯一标识，UUID格式，最大36个字符
name	String	数据类名称

状态码： 400

表 4-132 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-133 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

更新一条告警，告警名称为MyXXX，URL为http://xxx，发生次数为4次，置信度为4，严重等级为tips。

```
{
  "data_object" : {
```

```
  "version" : "1.0",
  "environment" : {
    "vendor_type" : "MyXXX",
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  },
  "data_source" : {
    "source_type" : 3,
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  },
  "first_observed_time" : "2021-01-30T23:00:00Z+0800",
  "last_observed_time" : "2021-01-30T23:00:00Z+0800",
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "arrive_time" : "2021-01-30T23:00:00Z+0800",
  "title" : "MyXXX",
  "description" : "This my XXXX",
  "source_url" : "http://xxx",
  "count" : 4,
  "confidence" : 4,
  "severity" : "TIPS",
  "criticality" : 4,
  "alert_type" : { },
  "network_list" : [ {
    "direction" : {
      "IN" : null
    },
    "protocol" : "TCP",
    "src_ip" : "192.168.0.1",
    "src_port" : "1",
    "src_domain" : "xxx",
    "dest_ip" : "192.168.0.1",
    "dest_port" : "1",
    "dest_domain" : "xxx",
    "src_geo" : {
      "latitude" : 90,
      "longitude" : 180
    },
    "dest_geo" : {
      "latitude" : 90,
      "longitude" : 180
    }
  } ],
  "resource_list" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "type" : "MyXXX",
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "ep_name" : "MyXXX",
    "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  } ],
  "remediation" : {
    "recommendation" : "MyXXX",
    "url" : "MyXXX"
  },
  "verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写Unknown",
  "handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
  "sla" : 60000,
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "close_time" : "2021-01-30T23:00:00Z+0800",
  "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
  "simulation" : "false",
  "actor" : "刘一博",
```

```
  "owner" : "MyXXX",
  "creator" : "MyXXX",
  "close_reason" : "误检;已解决;重复;其他",
  "close_comment" : "误检;已解决;重复;其他",
  "malware" : {
    "malware_family" : "family",
    "malware_class" : "恶意占用内存"
  },
  "system_info" : { },
  "process" : [ {
    "process_name" : "MyXXX",
    "process_path" : "MyXXX",
    "process_pid" : 123,
    "process_uid" : 123,
    "process_cmdline" : "MyXXX"
  } ],
  "user_info" : [ {
    "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "user_name" : "MyXXX"
  } ],
  "file_info" : [ {
    "file_path" : "MyXXX",
    "file_content" : "MyXXX",
    "file_new_path" : "MyXXX",
    "file_hash" : "MyXXX",
    "file_md5" : "MyXXX",
    "file_sha256" : "MyXXX",
    "file_attr" : "MyXXX"
  } ],
  "system_alert_table" : { },
  "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
 }
}
```

响应示例

状态码： 200

更新告警返回body体

```
{
 "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
 "message" : "Error message",
 "data" : {
   "data_object" : {
     "version" : "1.0",
     "environment" : {
       "vendor_type" : "MyXXX",
       "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
       "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
       "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
     },
     "data_source" : {
       "source_type" : 3,
       "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
       "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
       "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
     },
     "first_observed_time" : "2021-01-30T23:00:00Z+0800",
     "last_observed_time" : "2021-01-30T23:00:00Z+0800",
     "create_time" : "2021-01-30T23:00:00Z+0800",
     "arrive_time" : "2021-01-30T23:00:00Z+0800",
     "title" : "MyXXX",
     "description" : "This my XXXX",
     "source_url" : "http://xxx",
     "count" : 4,
     "confidence" : 4,
     "severity" : "TIPS",
```

```
    "criticality" : 4,
    "alert_type" : { },
    "network_list" : [ {
      "direction" : {
        "IN" : null
      },
      "protocol" : "TCP",
      "src_ip" : "192.168.0.1",
      "src_port" : "1",
      "src_domain" : "xxx",
      "dest_ip" : "192.168.0.1",
      "dest_port" : "1",
      "dest_domain" : "xxx",
      "src_geo" : {
        "latitude" : 90,
        "longitude" : 180
      },
      "dest_geo" : {
        "latitude" : 90,
        "longitude" : 180
      }
    } ],
    "resource_list" : [ {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "MyXXX",
      "type" : "MyXXX",
      "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "ep_name" : "MyXXX",
      "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    } ],
    "remediation" : {
      "recommendation" : "MyXXX",
      "url" : "MyXXX"
    },
    "verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写
Unknown",
    "handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
    "sla" : 60000,
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "close_time" : "2021-01-30T23:00:00Z+0800",
    "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
    "simulation" : "false",
    "actor" : "刘一博",
    "owner" : "MyXXX",
    "creator" : "MyXXX",
    "close_reason" : "误检;已解决;重复;其他",
    "close_comment" : "误检;已解决;重复;其他",
    "malware" : {
      "malware_family" : "family",
      "malware_class" : "恶意占用内存"
    },
    "system_info" : { },
    "process" : [ {
      "process_name" : "MyXXX",
      "process_path" : "MyXXX",
      "process_pid" : 123,
      "process_uid" : 123,
      "process_cmdline" : "MyXXX"
    } ],
    "user_info" : [ {
      "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "user_name" : "MyXXX"
    } ],
    "file_info" : [ {
      "file_path" : "MyXXX",
```

```
      "file_content" : "MyXXX",
      "file_new_path" : "MyXXX",
      "file_hash" : "MyXXX",
      "file_md5" : "MyXXX",
      "file_sha256" : "MyXXX",
      "file_attr" : "MyXXX"
     } ],
     "system_alert_table" : { },
     "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
     "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
   },
   "create_time" : "2021-01-30T23:00:00Z+0800",
   "update_time" : "2021-01-30T23:00:00Z+0800",
   "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
   "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
   "id" : "MyXXX",
   "version" : 11,
   "format_version" : 11,
   "dataclass_ref" : {
     "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
     "name" : "MyXXX"
   }
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新一条告警，告警名称为MyXXX，URL为http://xxx，发生次数为4次，置信度为4，严重等级为tips。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class ChangeAlertSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
```

```
        .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
        .build();
ChangeAlertRequest request = new ChangeAlertRequest();
request.withWorkspaceId("{workspace_id}");
request.withAlertId("{alert_id}");
ChangeAlertRequestBody body = new ChangeAlertRequestBody();
List<AlertFileInfo> listDataObjectFileInfo = new ArrayList<>();
listDataObjectFileInfo.add(
    new AlertFileInfo()
        .withFilePath("MyXXX")
        .withFileContent("MyXXX")
        .withFileNewPath("MyXXX")
        .withFileHash("MyXXX")
        .withFileMd5("MyXXX")
        .withFileSha256("MyXXX")
        .withFileAttr("MyXXX")
);
List<AlertUserInfo> listDataObjectUserInfo = new ArrayList<>();
listDataObjectUserInfo.add(
    new AlertUserInfo()
        .withUserId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withUserName("MyXXX")
);
List<AlertProcess> listDataObjectProcess = new ArrayList<>();
listDataObjectProcess.add(
    new AlertProcess()
        .withProcessName("MyXXX")
        .withProcessPath("MyXXX")
        .withProcessPid(123)
        .withProcessUid(123)
        .withProcessCmdline("MyXXX")
);
AlertMalware malwareDataObject = new AlertMalware();
malwareDataObject.withMalwareFamily("family")
    .withMalwareClass("恶意占用内存");
AlertRemediation remediationDataObject = new AlertRemediation();
remediationDataObject.withRecommendation("MyXXX")
    .withUrl("MyXXX");
List<AlertResourceList> listDataObjectResourceList = new ArrayList<>();
listDataObjectResourceList.add(
    new AlertResourceList()
        .withId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withName("MyXXX")
        .withType("MyXXX")
        .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withEpId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withEpName("MyXXX")
        .withTags("909494e3-558e-46b6-a9eb-07a8e18ca62f")
);
AlertDestGeo destGeoNetworkList = new AlertDestGeo();
destGeoNetworkList.withLatitude(java.math.BigDecimal.valueOf(90))
    .withLongitude(java.math.BigDecimal.valueOf(180));
AlertSrcGeo srcGeoNetworkList = new AlertSrcGeo();
srcGeoNetworkList.withLatitude(java.math.BigDecimal.valueOf(90))
    .withLongitude(java.math.BigDecimal.valueOf(180));
List<AlertNetworkList> listDataObjectNetworkList = new ArrayList<>();
listDataObjectNetworkList.add(
    new AlertNetworkList()
        .withDirection(AlertNetworkList.DirectionEnum.fromValue("{}"))
        .withProtocol("TCP")
        .withSrcIp("192.168.0.1")
        .withSrcPort(1)
        .withSrcDomain("xxx")
        .withSrcGeo(srcGeoNetworkList)
        .withDestIp("192.168.0.1")
        .withDestPort("1")
        .withDestDomain("xxx")
```

```
            .withDestGeo(destGeoNetworkList)
    );
    AlertDataSource dataSourceDataObject = new AlertDataSource();
    dataSourceDataObject.withSourceType(3)
        .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
    AlertEnvironment environmentDataObject = new AlertEnvironment();
    environmentDataObject.withVendorType("MyXXX")
        .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
    Alert dataObjectbody = new Alert();
    dataObjectbody.withVersion("1.0")
        .withId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withWorkspaceId("909494e3-558e-46b6-a9eb-07a8e18ca620")
        .withEnvironment(environmentDataObject)
        .withDataSource(dataSourceDataObject)
        .withFirstObservedTime("2021-01-30T23:00:00Z+0800")
        .withLastObservedTime("2021-01-30T23:00:00Z+0800")
        .withCreateTime("2021-01-30T23:00:00Z+0800")
        .withArriveTime("2021-01-30T23:00:00Z+0800")
        .withTitle("MyXXX")
        .withDescription("This my XXXX")
        .withSourceUrl("http://xxx")
        .withCount(4)
        .withConfidence(4)
        .withSeverity(Alert.SeverityEnum.fromValue("TIPS"))
        .withCriticality(4)
        .withNetworkList(listDataObjectNetworkList)
        .withResourceList(listDataObjectResourceList)
        .withRemediation(remediationDataObject)
        .withVerificationState(Alert.VerificationStateEnum.fromValue("Unknown – 未知,True_Positive – 确
认,False_Positive – 误报。默认填写Unknown"))
        .withHandleStatus(Alert.HandleStatusEnum.fromValue("Open – 打开,Block – 阻塞,Closed – 关闭。默
认填写Open"))
        .withSla(60000)
        .withUpdateTime("2021-01-30T23:00:00Z+0800")
        .withCloseTime("2021-01-30T23:00:00Z+0800")
        .withIpdrrPhase(Alert.IpdrrPhaseEnum.fromValue("Prepartion|Detection and Analysis|
Containm,Eradication& Recovery| Post-Incident-Activity"))
        .withSimulation("false")
        .withActor("刘一博")
        .withOwner("MyXXX")
        .withCreator("MyXXX")
        .withCloseReason(Alert.CloseReasonEnum.fromValue("误检;已解决;重复;其他"))
        .withCloseComment("误检;已解决;重复;其他")
        .withMalware(malwareDataObject)
        .withSystemInfo(new Object())
        .withProcess(listDataObjectProcess)
        .withUserInfo(listDataObjectUserInfo)
        .withFileInfo(listDataObjectFileInfo)
        .withSystemAlertTable(new Object());
    body.withDataObject(dataObjectbody);
    request.withBody(body);
    try {
        ChangeAlertResponse response = client.changeAlert(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

更新一条告警，告警名称为MyXXX，URL为http://xxx，发生次数为4次，置信度为4，严重等级为tips。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ChangeAlertRequest()
        request.workspace_id = "{workspace_id}"
        request.alert_id = "{alert_id}"
        listFileInfoDataObject = [
            AlertFileInfo(
                file_path="MyXXX",
                file_content="MyXXX",
                file_new_path="MyXXX",
                file_hash="MyXXX",
                file_md5="MyXXX",
                file_sha256="MyXXX",
                file_attr="MyXXX"
            )
        ]
        listUserInfoDataObject = [
            AlertUserInfo(
                user_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
                user_name="MyXXX"
            )
        ]
        listProcessDataObject = [
            AlertProcess(
                process_name="MyXXX",
                process_path="MyXXX",
                process_pid=123,
                process_uid=123,
                process_cmdline="MyXXX"
            )
        ]
        malwareDataObject = AlertMalware(
            malware_family="family",
            malware_class="恶意占用内存"
        )
        remediationDataObject = AlertRemediation(
            recommendation="MyXXX",
```

```
    url="MyXXX"
)
listResourceListDataObject = [
    AlertResourceList(
        id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        name="MyXXX",
        type="MyXXX",
        region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        ep_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        ep_name="MyXXX",
        tags="909494e3-558e-46b6-a9eb-07a8e18ca62f"
    )
]
destGeoNetworkList = AlertDestGeo(
    latitude=90,
    longitude=180
)
srcGeoNetworkList = AlertSrcGeo(
    latitude=90,
    longitude=180
)
listNetworkListDataObject = [
    AlertNetworkList(
        direction="{}",
        protocol="TCP",
        src_ip="192.168.0.1",
        src_port=1,
        src_domain="xxx",
        src_geo=srcGeoNetworkList,
        dest_ip="192.168.0.1",
        dest_port="1",
        dest_domain="xxx",
        dest_geo=destGeoNetworkList
    )
]
dataSourceDataObject = AlertDataSource(
    source_type=3,
    domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f"
)
environmentDataObject = AlertEnvironment(
    vendor_type="MyXXX",
    domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f"
)
dataObjectbody = Alert(
    version="1.0",
    id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    workspace_id="909494e3-558e-46b6-a9eb-07a8e18ca620",
    environment=environmentDataObject,
    data_source=dataSourceDataObject,
    first_observed_time="2021-01-30T23:00:00Z+0800",
    last_observed_time="2021-01-30T23:00:00Z+0800",
    create_time="2021-01-30T23:00:00Z+0800",
    arrive_time="2021-01-30T23:00:00Z+0800",
    title="MyXXX",
    description="This my XXXX",
    source_url="http://xxx",
    count=4,
    confidence=4,
    severity="TIPS",
    criticality=4,
    network_list=listNetworkListDataObject,
    resource_list=listResourceListDataObject,
    remediation=remediationDataObject,
```

```
            verification_state="Unknown – 未知,True_Positive – 确认,False_Positive – 误报。默认填写Unknown",
            handle_status="Open – 打开,Block – 阻塞,Closed – 关闭。默认填写Open",
            sla=60000,
            update_time="2021-01-30T23:00:00Z+0800",
            close_time="2021-01-30T23:00:00Z+0800",
            ipdrr_phase="Prepartion|Detection and Analysis|Containm,Eradication& Recovery| Post-Incident-
Activity",
            simulation="false",
            actor="刘一博",
            owner="MyXXX",
            creator="MyXXX",
            close_reason="误检;已解决;重复;其他",
            close_comment="误检;已解决;重复;其他",
            malware=malwareDataObject,
            system_info={},
            process=listProcessDataObject,
            user_info=listUserInfoDataObject,
            file_info=listFileInfoDataObject,
            system_alert_table={}
        )
        request.body = ChangeAlertRequestBody(
            data_object=dataObjectbody
        )
        response = client.change_alert(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

更新一条告警，告警名称为MyXXX，URL为http://xxx，发生次数为4次，置信度为4，严重等级为tips。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeAlertRequest{}
```



```
request.WorkspaceId = "{workspace_id}"
request.AlertId = "{alert_id}"
filePathFileInfo:= "MyXXX"
fileContentFileInfo:= "MyXXX"
fileNewPathFileInfo:= "MyXXX"
fileHashFileInfo:= "MyXXX"
fileMd5FileInfo:= "MyXXX"
fileSha256FileInfo:= "MyXXX"
fileAttrFileInfo:= "MyXXX"
var listFileInfoDataObject = []model.AlertFileInfo{
    {
        FilePath: &filePathFileInfo,
        FileContent: &fileContentFileInfo,
        FileNewPath: &fileNewPathFileInfo,
        FileHash: &fileHashFileInfo,
        FileMd5: &fileMd5FileInfo,
        FileSha256: &fileSha256FileInfo,
        FileAttr: &fileAttrFileInfo,
    },
}
userIdUserInfo:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
userNameUserInfo:= "MyXXX"
var listUserInfoDataObject = []model.AlertUserInfo{
    {
        UserId: &userIdUserInfo,
        UserName: &userNameUserInfo,
    },
}
processNameProcess:= "MyXXX"
processPathProcess:= "MyXXX"
processPidProcess:= int32(123)
processUidProcess:= int32(123)
processCmdlineProcess:= "MyXXX"
var listProcessDataObject = []model.AlertProcess{
    {
        ProcessName: &processNameProcess,
        ProcessPath: &processPathProcess,
        ProcessPid: &processPidProcess,
        ProcessUid: &processUidProcess,
        ProcessCmdline: &processCmdlineProcess,
    },
}
malwareFamilyMalware:= "family"
malwareClassMalware:= "恶意占用内存"
malwareDataObject := &model.AlertMalware{
    MalwareFamily: &malwareFamilyMalware,
    MalwareClass: &malwareClassMalware,
}
recommendationRemediation:= "MyXXX"
urlRemediation:= "MyXXX"
remediationDataObject := &model.AlertRemediation{
    Recommendation: &recommendationRemediation,
    Url: &urlRemediation,
}
idResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
nameResourceList:= "MyXXX"
typeResourceList:= "MyXXX"
regionIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
domainIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
projectIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
epIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
epNameResourceList:= "MyXXX"
tagsResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
var listResourceListDataObject = []model.AlertResourceList{
    {
        Id: &idResourceList,
        Name: &nameResourceList,
        Type: &typeResourceList,
        RegionId: &regionIdResourceList,
```

```
        DomainId: &domainIdResourceList,
        ProjectId: &projectIdResourceList,
        EpId: &epIdResourceList,
        EpName: &epNameResourceList,
        Tags: &tagsResourceList,
    },
}
latitudeDestGeo:= float32(90)
longitudeDestGeo:= float32(180)
destGeoNetworkList := &model.AlertDestGeo{
    Latitude: &latitudeDestGeo,
    Longitude: &longitudeDestGeo,
}
latitudeSrcGeo:= float32(90)
longitudeSrcGeo:= float32(180)
srcGeoNetworkList := &model.AlertSrcGeo{
    Latitude: &latitudeSrcGeo,
    Longitude: &longitudeSrcGeo,
}
directionNetworkList:= model.GetAlertNetworkListDirectionEnum().{}
protocolNetworkList:= "TCP"
srcIpNetworkList:= "192.168.0.1"
srcPortNetworkList:= int32(1)
srcDomainNetworkList:= "xxx"
destIpNetworkList:= "192.168.0.1"
destPortNetworkList:= "1"
destDomainNetworkList:= "xxx"
var listNetworkListDataObject = []model.AlertNetworkList{
    {
        Direction: &directionNetworkList,
        Protocol: &protocolNetworkList,
        SrcIp: &srcIpNetworkList,
        SrcPort: &srcPortNetworkList,
        SrcDomain: &srcDomainNetworkList,
        SrcGeo: srcGeoNetworkList,
        DestIp: &destIpNetworkList,
        DestPort: &destPortNetworkList,
        DestDomain: &destDomainNetworkList,
        DestGeo: destGeoNetworkList,
    },
}
sourceTypeDataSource:= int32(3)
domainIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
projectIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
regionIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
dataSourceDataObject := &model.AlertDataSource{
    SourceType: &sourceTypeDataSource,
    DomainId: &domainIdDataSource,
    ProjectId: &projectIdDataSource,
    RegionId: &regionIdDataSource,
}
vendorTypeEnvironment:= "MyXXX"
domainIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
regionIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
projectIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
environmentDataObject := &model.AlertEnvironment{
    VendorType: &vendorTypeEnvironment,
    DomainId: &domainIdEnvironment,
    RegionId: &regionIdEnvironment,
    ProjectId: &projectIdEnvironment,
}
versionDataObject:= "1.0"
idDataObject:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
workspaceIdDataObject:= "909494e3-558e-46b6-a9eb-07a8e18ca620"
firstObservedTimeDataObject:= "2021-01-30T23:00:00Z+0800"
lastObservedTimeDataObject:= "2021-01-30T23:00:00Z+0800"
createTimeDataObject:= "2021-01-30T23:00:00Z+0800"
arriveTimeDataObject:= "2021-01-30T23:00:00Z+0800"
titleDataObject:= "MyXXX"
```

```
    descriptionDataObject:= "This my XXXX"
    sourceUrlDataObject:= "http://xxx"
    countDataObject:= int32(4)
    confidenceDataObject:= int32(4)
    severityDataObject:= model.GetAlertSeverityEnum().TIPS
    criticalityDataObject:= int32(4)
    verificationStateDataObject:= model.GetAlertVerificationStateEnum().UNKNOWN_-_未
知,TRUE_POSITIVE_-_确认,FALSE_POSITIVE_-_误报。默认填写UNKNOWN
    handleStatusDataObject:= model.GetAlertHandleStatusEnum().OPEN_-_打开,BLOCK_-_阻塞,CLOSED_-_关
闭。默认填写OPEN
    slaDataObject:= int32(60000)
    updateTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    closeTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    ipdrrPhaseDataObject:= model.GetAlertIpdrrPhaseEnum().PREPARTION|DETECTION_AND_ANALYSIS|
CONTAINM,ERADICATION&_RECOVERY|_POST_INCIDENT_ACTIVITY
    simulationDataObject:= "false"
    actorDataObject:= "刘一博"
    ownerDataObject:= "MyXXX"
    creatorDataObject:= "MyXXX"
    closeReasonDataObject:= model.GetAlertCloseReasonEnum().误检;已解决;重复;其他
    closeCommentDataObject:= "误检;已解决;重复;其他"
    var systemInfoDataObject interface{} = make(map[string]string)
    var systemAlertTableDataObject interface{} = make(map[string]string)
    dataObjectbody := &model.Alert{
        Version: &versionDataObject,
        Id: &idDataObject,
        WorkspaceId: &workspaceIdDataObject,
        Environment: environmentDataObject,
        DataSource: dataSourceDataObject,
        FirstObservedTime: &firstObservedTimeDataObject,
        LastObservedTime: &lastObservedTimeDataObject,
        CreateTime: &createTimeDataObject,
        ArriveTime: &arriveTimeDataObject,
        Title: &titleDataObject,
        Description: &descriptionDataObject,
        SourceUrl: &sourceUrlDataObject,
        Count: &countDataObject,
        Confidence: &confidenceDataObject,
        Severity: &severityDataObject,
        Criticality: &criticalityDataObject,
        NetworkList: &listNetworkListDataObject,
        ResourceList: &listResourceListDataObject,
        Remediation: remediationDataObject,
        VerificationState: &verificationStateDataObject,
        HandleStatus: &handleStatusDataObject,
        Sla: &slaDataObject,
        UpdateTime: &updateTimeDataObject,
        CloseTime: &closeTimeDataObject,
        IpdrrPhase: &ipdrrPhaseDataObject,
        Simulation: &simulationDataObject,
        Actor: &actorDataObject,
        Owner: &ownerDataObject,
        Creator: &creatorDataObject,
        CloseReason: &closeReasonDataObject,
        CloseComment: &closeCommentDataObject,
        Malware: malwareDataObject,
        SystemInfo: &systemInfoDataObject,
        Process: &listProcessDataObject,
        UserInfo: &listUserInfoDataObject,
        FileInfo: &listFileInfoDataObject,
        SystemAlertTable: &systemAlertTableDataObject,
    }
    request.Body = &model.ChangeAlertRequestBody{
        DataObject: dataObjectbody,
    }
    response, err := client.ChangeAlert(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	更新告警返回body体
400	更新告警错误返回body体

错误码

请参见**错误码**。

4.2 事件管理

4.2.1 搜索事件列表

功能介绍

搜索事件列表

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/incidents/search

表 4-134 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

请求参数

表 4-135 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）
content-type	是	String	内容类型

表 4-136 请求 Body 参数

参数	是否必选	参数类型	描述
limit	否	Integer	分页大小
offset	否	Integer	偏移量
sort_by	否	String	排序字段：create_time | update_time
order	否	String	排序方式：DESC | ASC
from_date	否	String	搜索开始时间，例如：2023-02-20T00:00:00.000Z
to_date	否	String	搜索结束时间，例如：2023-02-27T23:59:59.999Z
condition	否	**condition** object	搜索条件表达式

表 4-137 condition

参数	是否必选	参数类型	描述
conditions	否	Array of **conditions** objects	表达式列表
logics	否	Array of strings	表达式名称列表

表 4-138 conditions

参数	是否必选	参数类型	描述
name	否	String	表达式名称
data	否	Array of strings	表达式内容列表

响应参数

状态码： 200

表 4-139 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-140 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
total	Integer	事件总数
limit	Integer	分页大小
offset	Integer	偏移量
success	Boolean	是否成功
data	Array of **IncidentDetail** objects	事件列表

表 4-141 IncidentDetail

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
data_object	**Incident** object	事件实体信息

参数	参数类型	描述
dataclass_ref	**dataclass_ref** object	数据类对象
format_version	Integer	格式版本
id	String	事件唯一标识，UUID格式，最大36个字符
project_id	String	当前项目的id
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为告警发生时区，无法解析时区的时间，默认时区填东八区
version	Integer	版本
workspace_id	String	当前的工作空间id

表 4-142 Incident

参数	参数类型	描述
version	String	事件对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	String	事件唯一标识，UUID格式，最大36个字符
domain_id	String	数据投递后，被委托用户的domain_id
region_id	String	数据投递后，被委托用户的region_id
workspace_id	String	当前的工作空间id
labels	String	标签，仅展示
environment	**environment** object	事件产生的环境坐标信息
data_source	**data_source** object	首次上报数据源
first_observed_time	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	String	事件标题
description	String	事件描述信息
source_url	String	事件URL链接，指向数据源产品中有关当前事件说明的页面
count	Integer	事件发生次数
confidence	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%
severity	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源
incident_type	**incident_type** object	事件分类，详细定义参考《告警事件类型定义》
network_list	Array of **network_list** objects	网络信息
resource_list	Array of **resource_list** objects	受影响资源

参数	参数类型	描述
remediation	**remediation** object	补救措施
verification_state	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	Integer	约束闭环时间：设置风险接受持续时间。单位：小时
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	String	调试字段
actor	String	事件调查员
owner	String	责任人、服务责任人
creator	String	创建人
close_reason	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	String	关闭评论

参数	参数类型	描述
malware	**malware** object	恶意软件
system_info	Object	系统信息
process	Array of **process** objects	进程信息
user_info	Array of **user_info** objects	用户信息
file_info	Array of **file_info** objects	文件信息
system_alert_table	Object	事件管理列表的布局字段

表 4-143 environment

参数	参数类型	描述
vendor_type	String	环境供应商
domain_id	String	租户id
region_id	String	区域id，全局服务global
cross_workspace_id	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	String	项目id， 全局服务默认null

表 4-144 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下： 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	数据源产品所属账号的id
project_id	String	数据源产品所属项目的id
region_id	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	String	数据源产品所属公司的名称
product_name	String	数据源产品的名称

参数	参数类型	描述
product_feature	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	String	检测模块列表

表 4-145 incident_type

参数	参数类型	描述
category	String	类别
incident_type	String	事件类型

表 4-146 network_list

参数	参数类型	描述
direction	String	方向，取值范围：IN | OUT
protocol	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	String	源IP地址
src_port	Integer	源端口，0–65535
src_domain	String	源域名
src_geo	**src_geo** object	源IP的地理位置信息
dest_ip	String	目的IP地址
dest_port	String	目的端口，0–65535
dest_domain	String	目的域名
dest_geo	**dest_geo** object	目标IP的地理位置信息

表 4-147 src_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度

参数	参数类型	描述
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-148 dest_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-149 resource_list

参数	参数类型	描述
id	String	云服务资源id
name	String	资源名称
type	String	资源类型；引用云RMS type字段
provider	String	云服务名称；引用云RMS provider字段
region_id	String	区域；按照云regionId填写，如cn-north-1等
domain_id	String	资源所属账号ID，UUID格式
project_id	String	资源所属项目ID，UUID格式
ep_id	String	企业项目id
ep_name	String	企业项目名称
tags	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-150 remediation

参数	参数类型	描述
recommendation	String	推荐处理方法
url	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-151 malware

参数	参数类型	描述
malware_family	String	恶意家族
malware_class	String	恶意软件分类

表 4-152 process

参数	参数类型	描述
process_name	String	进程名
process_path	String	进程执行文件路径
process_pid	Integer	进程id
process_uid	Integer	进程用户id
process_cmdline	String	进程命令行
process_parent_name	String	父进程名称
process_parent_path	String	父进程执行文件路径
process_parent_pid	Integer	父进程id
process_parent_uid	Integer	父进程用户id
process_parent_cmdline	String	父进程命令行
process_child_name	String	子进程名称
process_child_path	String	子进程执行文件路径
process_child_pid	Integer	子进程id

参数	参数类型	描述
process_child_uid	Integer	子进程用户id
process_child_cmdline	String	子进程命令行
process_launche_time	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-153 user_info

参数	参数类型	描述
user_id	String	用户uid
user_name	String	用户名称

表 4-154 file_info

参数	参数类型	描述
file_path	String	文件路径/名称
file_content	String	文件内容
file_new_path	String	文件新路径/名称
file_hash	String	文件hash
file_md5	String	文件md5
file_sha256	String	文件sha256
file_attr	String	文件属性

表 4-155 dataclass_ref

参数	参数类型	描述
id	String	数据类唯一标识，UUID格式，最大36个字符
name	String	数据类名称

状态码： 400

表 4-156 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-157 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

查询事件列表请求样例，查询2024年1月20号到2024年1月26号，事件等级为中危且处理状态为打开的事件，按照创建时间降序排序，返回第一页，每页10条数据

```
{
  "limit" : 10,
  "offset" : 0,
  "sort_by" : "create_time",
  "order" : "DESC",
  "condition" : {
    "conditions" : [ {
      "name" : "severity",
      "data" : [ "severity", "=", "Medium" ]
    }, {
      "name" : "handle_status",
      "data" : [ "handle_status", "=", "Open" ]
    } ],
    "logics" : [ "severity", "and", "handle_status" ]
  },
  "from_date" : "2024-01-20T00:00:00.000Z+0800",
  "to_date" : "2024-01-26T23:59:59.999Z+0800"
}
```

响应示例

状态码： 200

搜索事件列表返回body体

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "total" : 41,
  "limit" : 2,
  "offset" : 1,
  "success" : true,
  "data" : [ {
    "data_object" : {
      "version" : "1.0",
      "environment" : {
        "vendor_type" : "MyXXX",
```

```
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  },
  "data_source" : {
    "source_type" : 3,
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  },
  "first_observed_time" : "2021-01-30T23:00:00Z+0800",
  "last_observed_time" : "2021-01-30T23:00:00Z+0800",
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "arrive_time" : "2021-01-30T23:00:00Z+0800",
  "title" : "MyXXX",
  "description" : "This my XXXX",
  "source_url" : "http://xxx",
  "count" : 4,
  "confidence" : 4,
  "severity" : "TIPS",
  "criticality" : 4,
  "incident_type" : { },
  "network_list" : [ {
    "direction" : {
      "IN" : null
    },
    "protocol" : "TCP",
    "src_ip" : "192.168.0.1",
    "src_port" : "1",
    "src_domain" : "xxx",
    "dest_ip" : "192.168.0.1",
    "dest_port" : "1",
    "dest_domain" : "xxx",
    "src_geo" : {
      "latitude" : 90,
      "longitude" : 180
    },
    "dest_geo" : {
      "latitude" : 90,
      "longitude" : 180
    }
  } ],
  "resource_list" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "type" : "MyXXX",
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "ep_name" : "MyXXX",
    "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  } ],
  "remediation" : {
    "recommendation" : "MyXXX",
    "url" : "MyXXX"
  },
  "verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写
Unknown",
  "handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
  "sla" : 60000,
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "close_time" : "2021-01-30T23:00:00Z+0800",
  "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
  "simulation" : "false",
  "actor" : "刘一博",
  "owner" : "MyXXX",
  "creator" : "MyXXX",
```

```
    "close_reason" : "误检;已解决;重复;其他",
    "close_comment" : "误检;已解决;重复;其他",
    "malware" : {
      "malware_family" : "family",
      "malware_class" : "恶意占用内存"
    },
    "system_info" : { },
    "process" : [ {
      "process_name" : "MyXXX",
      "process_path" : "MyXXX",
      "process_pid" : 123,
      "process_uid" : 123,
      "process_cmdline" : "MyXXX"
    } ],
    "user_info" : [ {
      "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "user_name" : "MyXXX"
    } ],
    "file_info" : [ {
      "file_path" : "MyXXX",
      "file_content" : "MyXXX",
      "file_new_path" : "MyXXX",
      "file_hash" : "MyXXX",
      "file_md5" : "MyXXX",
      "file_sha256" : "MyXXX",
      "file_attr" : "MyXXX"
    } ],
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
  },
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
 } ]
}
```


SDK 代码示例

SDK代码示例如下。

Java

查询事件列表请求样例，查询2024年1月20号到2024年1月26号，事件等级为中危且处理状态为打开的事件，按照创建时间降序排序，返回第一页，每页10条数据

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class ListIncidentsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListIncidentsRequest request = new ListIncidentsRequest();
        request.withWorkspaceId("{workspace_id}");
        DataobjectSearch body = new DataobjectSearch();
        List<String> listConditionLogics = new ArrayList<>();
        listConditionLogics.add("severity");
        listConditionLogics.add("and");
        listConditionLogics.add("handle_status");
        List<String> listConditionsData = new ArrayList<>();
        listConditionsData.add("handle_status");
        listConditionsData.add("=");
        listConditionsData.add("Open");
        List<String> listConditionsData1 = new ArrayList<>();
        listConditionsData1.add("severity");
        listConditionsData1.add("=");
        listConditionsData1.add("Medium");
        List<DataobjectSearchConditionConditions> listConditionConditions = new ArrayList<>();
        listConditionConditions.add(
            new DataobjectSearchConditionConditions()
                .withName("severity")
                .withData(listConditionsData1)
        );
        listConditionConditions.add(
            new DataobjectSearchConditionConditions()
                .withName("handle_status")
                .withData(listConditionsData)
        );
        DataobjectSearchCondition conditionbody = new DataobjectSearchCondition();
        conditionbody.withConditions(listConditionConditions)
            .withLogics(listConditionLogics);
        body.withCondition(conditionbody);
        body.withToDate("2024-01-26T23:59:59.999Z+0800");
        body.withFromDate("2024-01-20T00:00:00.000Z+0800");
        body.withOrder(DataobjectSearch.OrderEnum.fromValue("DESC"));
        body.withSortBy("create_time");
        body.withOffset(0);
        body.withLimit(10);
        request.withBody(body);
        try {
            ListIncidentsResponse response = client.listIncidents(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

查询事件列表请求样例，查询2024年1月20号到2024年1月26号，事件等级为中危且处理状态为打开的事件，按照创建时间降序排序，返回第一页，每页10条数据

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListIncidentsRequest()
        request.workspace_id = "{workspace_id}"
        listLogicsCondition = [
            "severity",
            "and",
            "handle_status"
        ]
        listDataConditions = [
            "handle_status",
            "=",
            "Open"
        ]
        listDataConditions1 = [
            "severity",
            "=",
            "Medium"
        ]
        listConditionsCondition = [
            DataobjectSearchConditionConditions(
                name="severity",
                data=listDataConditions1
            ),
            DataobjectSearchConditionConditions(
                name="handle_status",
                data=listDataConditions
            )
        ]
        conditionbody = DataobjectSearchCondition(
            conditions=listConditionsCondition,
            logics=listLogicsCondition
        )
        request.body = DataobjectSearch(
            condition=conditionbody,
            to_date="2024-01-26T23:59:59.999Z+0800",
            from_date="2024-01-20T00:00:00.000Z+0800",
            order="DESC",
            sort_by="create_time",
            offset=0,
```

```
        limit=10
    )
    response = client.list_incidents(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

查询事件列表请求样例，查询2024年1月20号到2024年1月26号，事件等级为中危且处理状态为打开的事件，按照创建时间降序排序，返回第一页，每页10条数据

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListIncidentsRequest{}
    request.WorkspaceId = "{workspace_id}"
    var listLogicsCondition = []string{
        "severity",
        "and",
        "handle_status",
    }
    var listDataConditions = []string{
        "handle_status",
        "=",
        "Open",
    }
    var listDataConditions1 = []string{
        "severity",
        "=",
        "Medium",
    }
    nameConditions:= "severity"
    nameConditions1:= "handle_status"
    var listConditionsCondition = []model.DataobjectSearchConditionConditions{
        {
            Name: &nameConditions,
```

```
            Data: &listDataConditions1,
        },
        {
            Name: &nameConditions1,
            Data: &listDataConditions,
        },
    }
    conditionbody := &model.DataobjectSearchCondition{
        Conditions: &listConditionsCondition,
        Logics: &listLogicsCondition,
    }
    toDateDataobjectSearch:= "2024-01-26T23:59:59.999Z+0800"
    fromDateDataobjectSearch:= "2024-01-20T00:00:00.000Z+0800"
    orderDataobjectSearch:= model.GetDataobjectSearchOrderEnum().DESC
    sortByDataobjectSearch:= "create_time"
    offsetDataobjectSearch:= int32(0)
    limitDataobjectSearch:= int32(10)
    request.Body = &model.DataobjectSearch{
        Condition: conditionbody,
        ToDate: &toDateDataobjectSearch,
        FromDate: &fromDateDataobjectSearch,
        Order: &orderDataobjectSearch,
        SortBy: &sortByDataobjectSearch,
        Offset: &offsetDataobjectSearch,
        Limit: &limitDataobjectSearch,
    }
    response, err := client.ListIncidents(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	搜索事件列表返回body体
400	搜索事件列表错误返回body体

错误码

请参见**错误码**。

4.2.2 创建事件

功能介绍

创建事件

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/incidents

表 4-158 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

请求参数

表 4-159 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）
content-type	是	String	内容类型

表 4-160 请求 Body 参数

参数	是否必选	参数类型	描述
data_object	否	**Incident** object	事件实体信息

表 4-161 Incident

参数	是否必选	参数类型	描述
version	否	String	事件对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	否	String	事件唯一标识，UUID格式，最大36个字符
domain_id	否	String	数据投递后，被委托用户的domain_id
region_id	否	String	数据投递后，被委托用户的region_id

参数	是否必选	参数类型	描述
workspace_id	否	String	当前的工作空间id
labels	否	String	标签，仅展示
environment	否	**environment** object	事件产生的环境坐标信息
data_source	否	**data_source** object	首次上报数据源
first_observed_time	否	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	否	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
create_time	否	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	否	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	否	String	事件标题
description	否	String	事件描述信息
source_url	否	String	事件URL链接，指向数据源产品中有关当前事件说明的页面
count	否	Integer	事件发生次数
confidence	否	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%

参数	是否必选	参数类型	描述
severity	否	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	否	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源
incident_type	否	**incident_type** object	事件分类，详细定义参考《告警事件类型定义》
network_list	否	Array of **network_list** objects	网络信息
resource_list	否	Array of **resource_list** objects	受影响资源
remediation	否	**remediation** object	补救措施
verification_state	否	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	否	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	否	Integer	约束闭环时间：设置风险接受持续时间。单位：小时

参数	是否必选	参数类型	描述
update_time	否	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	否	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	否	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	否	String	调试字段
actor	否	String	事件调查员
owner	否	String	责任人、服务责任人
creator	否	String	创建人
close_reason	否	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	否	String	关闭评论
malware	否	**malware** object	恶意软件
system_info	否	Object	系统信息
process	否	Array of **process** objects	进程信息
user_info	否	Array of **user_info** objects	用户信息
file_info	否	Array of **file_info** objects	文件信息

参数	是否必选	参数类型	描述
system_alert_table	否	Object	事件管理列表的布局字段

表 4-162 environment

参数	是否必选	参数类型	描述
vendor_type	否	String	环境供应商
domain_id	否	String	租户id
region_id	否	String	区域id，全局服务global
cross_workspace_id	否	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	否	String	项目id， 全局服务默认null

表 4-163 data_source

参数	是否必选	参数类型	描述
source_type	否	Integer	数据源类型，取值范围如下： 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	否	String	数据源产品所属账号的id
project_id	否	String	数据源产品所属项目的id
region_id	否	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	否	String	数据源产品所属公司的名称
product_name	否	String	数据源产品的名称
product_feature	否	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	否	String	检测模块列表

表 4-164 incident_type

参数	是否必选	参数类型	描述
category	否	String	类别
incident_type	否	String	事件类型

表 4-165 network_list

参数	是否必选	参数类型	描述
direction	否	String	方向，取值范围：IN | OUT
protocol	否	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	否	String	源IP地址
src_port	否	Integer	源端口，0–65535
src_domain	否	String	源域名
src_geo	否	src_geo object	源IP的地理位置信息
dest_ip	否	String	目的IP地址
dest_port	否	String	目的端口，0–65535
dest_domain	否	String	目的域名
dest_geo	否	dest_geo object	目标IP的地理位置信息

表 4-166 src_geo

参数	是否必选	参数类型	描述
latitude	否	Number	纬度
longitude	否	Number	经度
city_code	否	String	城市编码，Beijing | Shanghai
country_code	否	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-167 dest_geo

参数	是否必选	参数类型	描述
latitude	否	Number	纬度
longitude	否	Number	经度
city_code	否	String	城市编码，Beijing | Shanghai
country_code	否	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-168 resource_list

参数	是否必选	参数类型	描述
id	否	String	云服务资源id
name	否	String	资源名称
type	否	String	资源类型；引用云RMS type字段
provider	否	String	云服务名称；引用云RMS provider字段
region_id	否	String	区域；按照云regionId填写，如cn-north-1等
domain_id	否	String	资源所属账号ID，UUID格式
project_id	否	String	资源所属项目ID，UUID格式
ep_id	否	String	企业项目id
ep_name	否	String	企业项目名称
tags	否	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-169 remediation

参数	是否必选	参数类型	描述
recommendation	否	String	推荐处理方法

参数	是否必选	参数类型	描述
url	否	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-170 malware

参数	是否必选	参数类型	描述
malware_family	否	String	恶意家族
malware_class	否	String	恶意软件分类

表 4-171 process

参数	是否必选	参数类型	描述
process_name	否	String	进程名
process_path	否	String	进程执行文件路径
process_pid	否	Integer	进程id
process_uid	否	Integer	进程用户id
process_cmdline	否	String	进程命令行
process_parent_name	否	String	父进程名称
process_parent_path	否	String	父进程执行文件路径
process_parent_pid	否	Integer	父进程id
process_parent_uid	否	Integer	父进程用户id
process_parent_cmdline	否	String	父进程命令行
process_child_name	否	String	子进程名称
process_child_path	否	String	子进程执行文件路径
process_child_pid	否	Integer	子进程id

参数	是否必选	参数类型	描述
process_child_uid	否	Integer	子进程用户id
process_child_cmdline	否	String	子进程命令行
process_launche_time	否	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	否	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-172 user_info

参数	是否必选	参数类型	描述
user_id	否	String	用户uid
user_name	否	String	用户名称

表 4-173 file_info

参数	是否必选	参数类型	描述
file_path	否	String	文件路径/名称
file_content	否	String	文件内容
file_new_path	否	String	文件新路径/名称
file_hash	否	String	文件hash
file_md5	否	String	文件md5
file_sha256	否	String	文件sha256
file_attr	否	String	文件属性

响应参数

状态码： 200

表 4-174 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-175 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**IncidentDetail** object	事件详情对象

表 4-176 IncidentDetail

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
data_object	**Incident** object	事件实体信息
dataclass_ref	**dataclass_ref** object	数据类对象
format_version	Integer	格式版本
id	String	事件唯一标识，UUID格式，最大36个字符
project_id	String	当前项目的id
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为告警发生时区，无法解析时区的时间，默认时区填东八区
version	Integer	版本
workspace_id	String	当前的工作空间id

表 4-177 Incident

参数	参数类型	描述
version	String	事件对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	String	事件唯一标识，UUID格式，最大36个字符
domain_id	String	数据投递后，被委托用户的domain_id
region_id	String	数据投递后，被委托用户的region_id
workspace_id	String	当前的工作空间id
labels	String	标签，仅展示
environment	**environment** object	事件产生的环境坐标信息
data_source	**data_source** object	首次上报数据源
first_observed_time	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	String	事件标题
description	String	事件描述信息
source_url	String	事件URL链接，指向数据源产品中有关当前事件说明的页面
count	Integer	事件发生次数
confidence	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%

参数	参数类型	描述
severity	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源
incident_type	**incident_type** object	事件分类，详细定义参考《告警事件类型定义》
network_list	Array of **network_list** objects	网络信息
resource_list	Array of **resource_list** objects	受影响资源
remediation	**remediation** object	补救措施
verification_state	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	Integer	约束闭环时间：设置风险接受持续时间。单位：小时

参数	参数类型	描述
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	String	调试字段
actor	String	事件调查员
owner	String	责任人、服务责任人
creator	String	创建人
close_reason	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	String	关闭评论
malware	**malware** object	恶意软件
system_info	Object	系统信息
process	Array of **process** objects	进程信息
user_info	Array of **user_info** objects	用户信息
file_info	Array of **file_info** objects	文件信息
system_alert_table	Object	事件管理列表的布局字段

表 4-178 environment

参数	参数类型	描述
vendor_type	String	环境供应商

参数	参数类型	描述
domain_id	String	租户id
region_id	String	区域id，全局服务global
cross_workspace_id	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	String	项目id， 全局服务默认null

表 4-179 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下： 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	数据源产品所属账号的id
project_id	String	数据源产品所属项目的id
region_id	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	String	数据源产品所属公司的名称
product_name	String	数据源产品的名称
product_feature	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	String	检测模块列表

表 4-180 incident_type

参数	参数类型	描述
category	String	类别
incident_type	String	事件类型

表 4-181 network_list

参数	参数类型	描述
direction	String	方向，取值范围：IN | OUT

参数	参数类型	描述
protocol	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	String	源IP地址
src_port	Integer	源端口，0–65535
src_domain	String	源域名
src_geo	**src_geo** object	源IP的地理位置信息
dest_ip	String	目的IP地址
dest_port	String	目的端口，0–65535
dest_domain	String	目的域名
dest_geo	**dest_geo** object	目标IP的地理位置信息

表 4-182 src_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-183 dest_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-184 resource_list

参数	参数类型	描述
id	String	云服务资源id
name	String	资源名称
type	String	资源类型；引用云RMS type字段
provider	String	云服务名称；引用云RMS provider字段
region_id	String	区域；按照云regionId填写，如cn-north-1等
domain_id	String	资源所属账号ID，UUID格式
project_id	String	资源所属项目ID，UUID格式
ep_id	String	企业项目id
ep_name	String	企业项目名称
tags	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-185 remediation

参数	参数类型	描述
recommendation	String	推荐处理方法
url	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-186 malware

参数	参数类型	描述
malware_family	String	恶意家族
malware_class	String	恶意软件分类

表 4-187 process

参数	参数类型	描述
process_name	String	进程名

参数	参数类型	描述
process_path	String	进程执行文件路径
process_pid	Integer	进程id
process_uid	Integer	进程用户id
process_cmdline	String	进程命令行
process_parent_name	String	父进程名称
process_parent_path	String	父进程执行文件路径
process_parent_pid	Integer	父进程id
process_parent_uid	Integer	父进程用户id
process_parent_cmdline	String	父进程命令行
process_child_name	String	子进程名称
process_child_path	String	子进程执行文件路径
process_child_pid	Integer	子进程id
process_child_uid	Integer	子进程用户id
process_child_cmdline	String	子进程命令行
process_launche_time	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-188 user_info

参数	参数类型	描述
user_id	String	用户uid
user_name	String	用户名称

表 4-189 file_info

参数	参数类型	描述
file_path	String	文件路径/名称
file_content	String	文件内容
file_new_path	String	文件新路径/名称
file_hash	String	文件hash
file_md5	String	文件md5
file_sha256	String	文件sha256
file_attr	String	文件属性

表 4-190 dataclass_ref

参数	参数类型	描述
id	String	数据类唯一标识，UUID格式，最大36个字符
name	String	数据类名称

状态码： 400

表 4-191 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-192 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

创建一条事件，事件标题为MyXXX，标签为MyXXX，严重级别为tips，发生次数为4次。

```
{
  "data_object" : {
```

```
"version" : "1.0",
"environment" : {
  "vendor_type" : "MyXXX",
  "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
},
"data_source" : {
  "source_type" : 3,
  "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "product_name" : "test",
  "product_feature" : "test"
},
"first_observed_time" : "2021-01-30T23:00:00Z+0800",
"last_observed_time" : "2021-01-30T23:00:00Z+0800",
"create_time" : "2021-01-30T23:00:00Z+0800",
"arrive_time" : "2021-01-30T23:00:00Z+0800",
"title" : "MyXXX",
"labels" : "MyXXX",
"description" : "This my XXXX",
"source_url" : "http://xxx",
"count" : 4,
"confidence" : 4,
"severity" : "TIPS",
"criticality" : 4,
"incident_type" : {
  "incident_type" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "category" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
},
"network_list" : [ {
  "direction" : {
    "IN" : null
  },
  "protocol" : "TCP",
  "src_ip" : "192.168.0.1",
  "src_port" : "1",
  "src_domain" : "xxx",
  "dest_ip" : "192.168.0.1",
  "dest_port" : "1",
  "dest_domain" : "xxx",
  "src_geo" : {
    "latitude" : 90,
    "longitude" : 180
  },
  "dest_geo" : {
    "latitude" : 90,
    "longitude" : 180
  }
} ],
"resource_list" : [ {
  "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "name" : "MyXXX",
  "type" : "MyXXX",
  "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "ep_name" : "MyXXX",
  "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
} ],
"remediation" : {
  "recommendation" : "MyXXX",
  "url" : "MyXXX"
},
"verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写Unknown",
"handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
"sla" : 60000,
```



```
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "close_time" : "2021-01-30T23:00:00Z+0800",
    "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
    "simulation" : "false",
    "actor" : "刘一博",
    "owner" : "MyXXX",
    "creator" : "MyXXX",
    "close_reason" : "误检;已解决;重复;其他",
    "close_comment" : "误检;已解决;重复;其他",
    "malware" : {
      "malware_family" : "family",
      "malware_class" : "恶意占用内存"
    },
    "system_info" : { },
    "process" : [ {
      "process_name" : "MyXXX",
      "process_path" : "MyXXX",
      "process_pid" : 123,
      "process_uid" : 123,
      "process_cmdline" : "MyXXX"
    } ],
    "user_info" : [ {
      "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "user_name" : "MyXXX"
    } ],
    "file_info" : [ {
      "file_path" : "MyXXX",
      "file_content" : "MyXXX",
      "file_new_path" : "MyXXX",
      "file_hash" : "MyXXX",
      "file_md5" : "MyXXX",
      "file_sha256" : "MyXXX",
      "file_attr" : "MyXXX"
    } ],
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
  }
}
```

响应示例

状态码： 200

创建事件返回body体

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "data" : {
    "data_object" : {
      "version" : "1.0",
      "environment" : {
        "vendor_type" : "MyXXX",
        "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
      },
      "data_source" : {
        "source_type" : 3,
        "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
      },
      "first_observed_time" : "2021-01-30T23:00:00Z+0800",
      "last_observed_time" : "2021-01-30T23:00:00Z+0800",
      "create_time" : "2021-01-30T23:00:00Z+0800",
      "arrive_time" : "2021-01-30T23:00:00Z+0800",
      "title" : "MyXXX",
```

```
    "description" : "This my XXXX",
    "source_url" : "http://xxx",
    "count" : 4,
    "confidence" : 4,
    "severity" : "TIPS",
    "criticality" : 4,
    "incident_type" : { },
    "network_list" : [ {
      "direction" : {
        "IN" : null
      },
      "protocol" : "TCP",
      "src_ip" : "192.168.0.1",
      "src_port" : "1",
      "src_domain" : "xxx",
      "dest_ip" : "192.168.0.1",
      "dest_port" : "1",
      "dest_domain" : "xxx",
      "src_geo" : {
        "latitude" : 90,
        "longitude" : 180
      },
      "dest_geo" : {
        "latitude" : 90,
        "longitude" : 180
      }
    } ],
    "resource_list" : [ {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "MyXXX",
      "type" : "MyXXX",
      "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "ep_name" : "MyXXX",
      "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    } ],
    "remediation" : {
      "recommendation" : "MyXXX",
      "url" : "MyXXX"
    },
    "verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写
Unknown",
    "handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
    "sla" : 60000,
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "close_time" : "2021-01-30T23:00:00Z+0800",
    "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
    "simulation" : "false",
    "actor" : "刘一博",
    "owner" : "MyXXX",
    "creator" : "MyXXX",
    "close_reason" : "误检;已解决;重复;其他",
    "close_comment" : "误检;已解决;重复;其他",
    "malware" : {
      "malware_family" : "family",
      "malware_class" : "恶意占用内存"
    },
    "system_info" : { },
    "process" : [ {
      "process_name" : "MyXXX",
      "process_path" : "MyXXX",
      "process_pid" : 123,
      "process_uid" : 123,
      "process_cmdline" : "MyXXX"
    } ],
    "user_info" : [ {
```



```
      "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "user_name" : "MyXXX"
    } ],
    "file_info" : [ {
      "file_path" : "MyXXX",
      "file_content" : "MyXXX",
      "file_new_path" : "MyXXX",
      "file_hash" : "MyXXX",
      "file_md5" : "MyXXX",
      "file_sha256" : "MyXXX",
      "file_attr" : "MyXXX"
    } ],
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
  },
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
 }
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一条事件，事件标题为MyXXX，标签为MyXXX，严重级别为tips，发生次数为4次。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class CreateIncidentSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateIncidentRequest request = new CreateIncidentRequest();
```

```
request.withWorkspaceId("{workspace_id}");
CreateIncidentRequestBody body = new CreateIncidentRequestBody();
List<IncidentFileInfo> listDataObjectFileInfo = new ArrayList<>();
listDataObjectFileInfo.add(
    new IncidentFileInfo()
        .withFilePath("MyXXX")
        .withFileContent("MyXXX")
        .withFileNewPath("MyXXX")
        .withFileHash("MyXXX")
        .withFileMd5("MyXXX")
        .withFileSha256("MyXXX")
        .withFileAttr("MyXXX")
);
List<IncidentUserInfo> listDataObjectUserInfo = new ArrayList<>();
listDataObjectUserInfo.add(
    new IncidentUserInfo()
        .withUserId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withUserName("MyXXX")
);
List<IncidentProcess> listDataObjectProcess = new ArrayList<>();
listDataObjectProcess.add(
    new IncidentProcess()
        .withProcessName("MyXXX")
        .withProcessPath("MyXXX")
        .withProcessPid(123)
        .withProcessUid(123)
        .withProcessCmdline("MyXXX")
);
IncidentMalware malwareDataObject = new IncidentMalware();
malwareDataObject.withMalwareFamily("family")
    .withMalwareClass("恶意占用内存");
IncidentRemediation remediationDataObject = new IncidentRemediation();
remediationDataObject.withRecommendation("MyXXX")
    .withUrl("MyXXX");
List<IncidentResourceList> listDataObjectResourceList = new ArrayList<>();
listDataObjectResourceList.add(
    new IncidentResourceList()
        .withId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withName("MyXXX")
        .withType("MyXXX")
        .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withEpId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withEpName("MyXXX")
        .withTags("909494e3-558e-46b6-a9eb-07a8e18ca62f")
);
IncidentDestGeo destGeoNetworkList = new IncidentDestGeo();
destGeoNetworkList.withLatitude(java.math.BigDecimal.valueOf(90))
    .withLongitude(java.math.BigDecimal.valueOf(180));
IncidentSrcGeo srcGeoNetworkList = new IncidentSrcGeo();
srcGeoNetworkList.withLatitude(java.math.BigDecimal.valueOf(90))
    .withLongitude(java.math.BigDecimal.valueOf(180));
List<IncidentNetworkList> listDataObjectNetworkList = new ArrayList<>();
listDataObjectNetworkList.add(
    new IncidentNetworkList()
        .withDirection(IncidentNetworkList.DirectionEnum.fromValue("{}"))
        .withProtocol("TCP")
        .withSrcIp("192.168.0.1")
        .withSrcPort(1)
        .withSrcDomain("xxx")
        .withSrcGeo(srcGeoNetworkList)
        .withDestIp("192.168.0.1")
        .withDestPort("1")
        .withDestDomain("xxx")
        .withDestGeo(destGeoNetworkList)
);
IncidentIncidentType incidentTypeDataObject = new IncidentIncidentType();
incidentTypeDataObject.withCategory("909494e3-558e-46b6-a9eb-07a8e18ca62f")
```

```
            .withIncidentType("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        IncidentDataSource dataSourceDataObject = new IncidentDataSource();
        dataSourceDataObject.withSourceType(3)
            .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withProductName("test")
            .withProductFeature("test");
        IncidentEnvironment environmentDataObject = new IncidentEnvironment();
        environmentDataObject.withVendorType("MyXXX")
            .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        Incident dataObjectbody = new Incident();
        dataObjectbody.withVersion("1.0")
            .withId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withWorkspaceId("909494e3-558e-46b6-a9eb-07a8e18ca620")
            .withLabels("MyXXX")
            .withEnvironment(environmentDataObject)
            .withDataSource(dataSourceDataObject)
            .withFirstObservedTime("2021-01-30T23:00:00Z+0800")
            .withLastObservedTime("2021-01-30T23:00:00Z+0800")
            .withCreateTime("2021-01-30T23:00:00Z+0800")
            .withArriveTime("2021-01-30T23:00:00Z+0800")
            .withTitle("MyXXX")
            .withDescription("This my XXXX")
            .withSourceUrl("http://xxx")
            .withCount(4)
            .withConfidence(4)
            .withSeverity(Incident.SeverityEnum.fromValue("TIPS"))
            .withCriticality(4)
            .withIncidentType(incidentTypeDataObject)
            .withNetworkList(listDataObjectNetworkList)
            .withResourceList(listDataObjectResourceList)
            .withRemediation(remediationDataObject)
            .withVerificationState(Incident.VerificationStateEnum.fromValue("Unknown – 未知,True_Positive – 确
认,False_Positive – 误报。默认填写Unknown"))
            .withHandleStatus(Incident.HandleStatusEnum.fromValue("Open – 打开,Block – 阻塞,Closed – 关
闭。默认填写Open"))
            .withSla(60000)
            .withUpdateTime("2021-01-30T23:00:00Z+0800")
            .withCloseTime("2021-01-30T23:00:00Z+0800")
            .withIpdrrPhase(Incident.IpdrrPhaseEnum.fromValue("Prepartion|Detection and Analysis|
Containm,Eradication& Recovery| Post-Incident-Activity"))
            .withSimulation("false")
            .withActor("刘一博")
            .withOwner("MyXXX")
            .withCreator("MyXXX")
            .withCloseReason(Incident.CloseReasonEnum.fromValue("误检;已解决;重复;其他"))
            .withCloseComment("误检;已解决;重复;其他")
            .withMalware(malwareDataObject)
            .withSystemInfo(new Object())
            .withProcess(listDataObjectProcess)
            .withUserInfo(listDataObjectUserInfo)
            .withFileInfo(listDataObjectFileInfo);
        body.withDataObject(dataObjectbody);
        request.withBody(body);
        try {
            CreateIncidentResponse response = client.createIncident(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

创建一条事件，事件标题为MyXXX，标签为MyXXX，严重级别为tips，发生次数为4次。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateIncidentRequest()
        request.workspace_id = "{workspace_id}"
        listFileInfoDataObject = [
            IncidentFileInfo(
                file_path="MyXXX",
                file_content="MyXXX",
                file_new_path="MyXXX",
                file_hash="MyXXX",
                file_md5="MyXXX",
                file_sha256="MyXXX",
                file_attr="MyXXX"
            )
        ]
        listUserInfoDataObject = [
            IncidentUserInfo(
                user_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
                user_name="MyXXX"
            )
        ]
        listProcessDataObject = [
            IncidentProcess(
                process_name="MyXXX",
                process_path="MyXXX",
                process_pid=123,
                process_uid=123,
                process_cmdline="MyXXX"
            )
        ]
        malwareDataObject = IncidentMalware(
            malware_family="family",
            malware_class="恶意占用内存"
        )
        remediationDataObject = IncidentRemediation(
```

```
    recommendation="MyXXX",
    url="MyXXX"
)
listResourceListDataObject = [
    IncidentResourceList(
        id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        name="MyXXX",
        type="MyXXX",
        region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        ep_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        ep_name="MyXXX",
        tags="909494e3-558e-46b6-a9eb-07a8e18ca62f"
    )
]
destGeoNetworkList = IncidentDestGeo(
    latitude=90,
    longitude=180
)
srcGeoNetworkList = IncidentSrcGeo(
    latitude=90,
    longitude=180
)
listNetworkListDataObject = [
    IncidentNetworkList(
        direction="{}",
        protocol="TCP",
        src_ip="192.168.0.1",
        src_port=1,
        src_domain="xxx",
        src_geo=srcGeoNetworkList,
        dest_ip="192.168.0.1",
        dest_port="1",
        dest_domain="xxx",
        dest_geo=destGeoNetworkList
    )
]
incidentTypeDataObject = IncidentIncidentType(
    category="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    incident_type="909494e3-558e-46b6-a9eb-07a8e18ca62f"
)
dataSourceDataObject = IncidentDataSource(
    source_type=3,
    domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    product_name="test",
    product_feature="test"
)
environmentDataObject = IncidentEnvironment(
    vendor_type="MyXXX",
    domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f"
)
dataObjectbody = Incident(
    version="1.0",
    id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
    workspace_id="909494e3-558e-46b6-a9eb-07a8e18ca620",
    labels="MyXXX",
    environment=environmentDataObject,
    data_source=dataSourceDataObject,
    first_observed_time="2021-01-30T23:00:00Z+0800",
    last_observed_time="2021-01-30T23:00:00Z+0800",
    create_time="2021-01-30T23:00:00Z+0800",
    arrive_time="2021-01-30T23:00:00Z+0800",
    title="MyXXX",
    description="This my XXXX",
```

```
            source_url="http://xxx",
            count=4,
            confidence=4,
            severity="TIPS",
            criticality=4,
            incident_type=incidentTypeDataObject,
            network_list=listNetworkListDataObject,
            resource_list=listResourceListDataObject,
            remediation=remediationDataObject,
            verification_state="Unknown – 未知,True_Positive – 确认,False_Positive – 误报。默认填写Unknown",
            handle_status="Open – 打开,Block – 阻塞,Closed – 关闭。默认填写Open",
            sla=60000,
            update_time="2021-01-30T23:00:00Z+0800",
            close_time="2021-01-30T23:00:00Z+0800",
            ipdrr_phase="Prepartion|Detection and Analysis|Containm,Eradication& Recovery| Post-Incident-
Activity",
            simulation="false",
            actor="刘一博",
            owner="MyXXX",
            creator="MyXXX",
            close_reason="误检;已解决;重复;其他",
            close_comment="误检;已解决;重复;其他",
            malware=malwareDataObject,
            system_info={},
            process=listProcessDataObject,
            user_info=listUserInfoDataObject,
            file_info=listFileInfoDataObject
        )
        request.body = CreateIncidentRequestBody(
            data_object=dataObjectbody
        )
        response = client.create_incident(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

创建一条事件，事件标题为MyXXX，标签为MyXXX，严重级别为tips，发生次数为4次。

```
package main

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

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

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

```
client := secmaster.NewSecMasterClient(
    secmaster.SecMasterClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateIncidentRequest{}
request.WorkspaceId = "{workspace_id}"
filePathFileInfo:= "MyXXX"
fileContentFileInfo:= "MyXXX"
fileNewPathFileInfo:= "MyXXX"
fileHashFileInfo:= "MyXXX"
fileMd5FileInfo:= "MyXXX"
fileSha256FileInfo:= "MyXXX"
fileAttrFileInfo:= "MyXXX"
var listFileInfoDataObject = []model.IncidentFileInfo{
    {
        FilePath: &filePathFileInfo,
        FileContent: &fileContentFileInfo,
        FileNewPath: &fileNewPathFileInfo,
        FileHash: &fileHashFileInfo,
        FileMd5: &fileMd5FileInfo,
        FileSha256: &fileSha256FileInfo,
        FileAttr: &fileAttrFileInfo,
    },
}
userIdUserInfo:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
userNameUserInfo:= "MyXXX"
var listUserInfoDataObject = []model.IncidentUserInfo{
    {
        UserId: &userIdUserInfo,
        UserName: &userNameUserInfo,
    },
}
processNameProcess:= "MyXXX"
processPathProcess:= "MyXXX"
processPidProcess:= int32(123)
processUidProcess:= int32(123)
processCmdlineProcess:= "MyXXX"
var listProcessDataObject = []model.IncidentProcess{
    {
        ProcessName: &processNameProcess,
        ProcessPath: &processPathProcess,
        ProcessPid: &processPidProcess,
        ProcessUid: &processUidProcess,
        ProcessCmdline: &processCmdlineProcess,
    },
}
malwareFamilyMalware:= "family"
malwareClassMalware:= "恶意占用内存"
malwareDataObject := &model.IncidentMalware{
    MalwareFamily: &malwareFamilyMalware,
    MalwareClass: &malwareClassMalware,
}
recommendationRemediation:= "MyXXX"
urlRemediation:= "MyXXX"
remediationDataObject := &model.IncidentRemediation{
    Recommendation: &recommendationRemediation,
    Url: &urlRemediation,
}
idResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
nameResourceList:= "MyXXX"
typeResourceList:= "MyXXX"
regionIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
domainIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
projectIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
epIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
epNameResourceList:= "MyXXX"
```

```
tagsResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
var listResourceListDataObject = []model.IncidentResourceList{
    {
        Id: &idResourceList,
        Name: &nameResourceList,
        Type: &typeResourceList,
        RegionId: &regionIdResourceList,
        DomainId: &domainIdResourceList,
        ProjectId: &projectIdResourceList,
        EpId: &epIdResourceList,
        EpName: &epNameResourceList,
        Tags: &tagsResourceList,
    },
}
latitudeDestGeo:= float32(90)
longitudeDestGeo:= float32(180)
destGeoNetworkList := &model.IncidentDestGeo{
    Latitude: &latitudeDestGeo,
    Longitude: &longitudeDestGeo,
}
latitudeSrcGeo:= float32(90)
longitudeSrcGeo:= float32(180)
srcGeoNetworkList := &model.IncidentSrcGeo{
    Latitude: &latitudeSrcGeo,
    Longitude: &longitudeSrcGeo,
}
directionNetworkList:= model.GetIncidentNetworkListDirectionEnum().{}
protocolNetworkList:= "TCP"
srcIpNetworkList:= "192.168.0.1"
srcPortNetworkList:= int32(1)
srcDomainNetworkList:= "xxx"
destIpNetworkList:= "192.168.0.1"
destPortNetworkList:= "1"
destDomainNetworkList:= "xxx"
var listNetworkListDataObject = []model.IncidentNetworkList{
    {
        Direction: &directionNetworkList,
        Protocol: &protocolNetworkList,
        SrcIp: &srcIpNetworkList,
        SrcPort: &srcPortNetworkList,
        SrcDomain: &srcDomainNetworkList,
        SrcGeo: srcGeoNetworkList,
        DestIp: &destIpNetworkList,
        DestPort: &destPortNetworkList,
        DestDomain: &destDomainNetworkList,
        DestGeo: destGeoNetworkList,
    },
}
categoryIncidentType:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
incidentTypeIncidentType:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
incidentTypeDataObject := &model.IncidentIncidentType{
    Category: &categoryIncidentType,
    IncidentType: &incidentTypeIncidentType,
}
sourceTypeDataSource:= int32(3)
domainIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
projectIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
regionIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
productNameDataSource:= "test"
productFeatureDataSource:= "test"
dataSourceDataObject := &model.IncidentDataSource{
    SourceType: &sourceTypeDataSource,
    DomainId: &domainIdDataSource,
    ProjectId: &projectIdDataSource,
    RegionId: &regionIdDataSource,
    ProductName: &productNameDataSource,
    ProductFeature: &productFeatureDataSource,
}
vendorTypeEnvironment:= "MyXXX"
```

```
    domainIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    regionIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    projectIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    environmentDataObject := &model.IncidentEnvironment{
        VendorType: &vendorTypeEnvironment,
        DomainId: &domainIdEnvironment,
        RegionId: &regionIdEnvironment,
        ProjectId: &projectIdEnvironment,
    }
    versionDataObject:= "1.0"
    idDataObject:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    workspaceIdDataObject:= "909494e3-558e-46b6-a9eb-07a8e18ca620"
    labelsDataObject:= "MyXXX"
    firstObservedTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    lastObservedTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    createTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    arriveTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    titleDataObject:= "MyXXX"
    descriptionDataObject:= "This my XXXX"
    sourceUrlDataObject:= "http://xxx"
    countDataObject:= int32(4)
    confidenceDataObject:= int32(4)
    severityDataObject:= model.GetIncidentSeverityEnum().TIPS
    criticalityDataObject:= int32(4)
    verificationStateDataObject:= model.GetIncidentVerificationStateEnum().UNKNOWN_-_未
知,TRUE_POSITIVE_-_确认,FALSE_POSITIVE_-_误报。默认填写UNKNOWN
    handleStatusDataObject:= model.GetIncidentHandleStatusEnum().OPEN_-_打开,BLOCK_-_阻塞,CLOSED_-
_关闭。默认填写OPEN
    slaDataObject:= int32(60000)
    updateTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    closeTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    ipdrrPhaseDataObject:= model.GetIncidentIpdrrPhaseEnum().PREPARTION|DETECTION_AND_ANALYSIS|
CONTAINM,ERADICATION&_RECOVERY|_POST_INCIDENT_ACTIVITY
    simulationDataObject:= "false"
    actorDataObject:= "刘一博"
    ownerDataObject:= "MyXXX"
    creatorDataObject:= "MyXXX"
    closeReasonDataObject:= model.GetIncidentCloseReasonEnum().误检;已解决;重复;其他
    closeCommentDataObject:= "误检;已解决;重复;其他"
    var systemInfoDataObject interface{} = make(map[string]string)
    dataObjectbody := &model.Incident{
        Version: &versionDataObject,
        Id: &idDataObject,
        WorkspaceId: &workspaceIdDataObject,
        Labels: &labelsDataObject,
        Environment: environmentDataObject,
        DataSource: dataSourceDataObject,
        FirstObservedTime: &firstObservedTimeDataObject,
        LastObservedTime: &lastObservedTimeDataObject,
        CreateTime: &createTimeDataObject,
        ArriveTime: &arriveTimeDataObject,
        Title: &titleDataObject,
        Description: &descriptionDataObject,
        SourceUrl: &sourceUrlDataObject,
        Count: &countDataObject,
        Confidence: &confidenceDataObject,
        Severity: &severityDataObject,
        Criticality: &criticalityDataObject,
        IncidentType: incidentTypeDataObject,
        NetworkList: &listNetworkListDataObject,
        ResourceList: &listResourceListDataObject,
        Remediation: remediationDataObject,
        VerificationState: &verificationStateDataObject,
        HandleStatus: &handleStatusDataObject,
        Sla: &slaDataObject,
        UpdateTime: &updateTimeDataObject,
        CloseTime: &closeTimeDataObject,
        IpdrrPhase: &ipdrrPhaseDataObject,
        Simulation: &simulationDataObject,
```

```
        Actor: &actorDataObject,
        Owner: &ownerDataObject,
        Creator: &creatorDataObject,
        CloseReason: &closeReasonDataObject,
        CloseComment: &closeCommentDataObject,
        Malware: malwareDataObject,
        SystemInfo: &systemInfoDataObject,
        Process: &listProcessDataObject,
        UserInfo: &listUserInfoDataObject,
        FileInfo: &listFileInfoDataObject,
    }
    request.Body = &model.CreateIncidentRequestBody{
        DataObject: dataObjectbody,
    }
    response, err := client.CreateIncident(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	创建事件返回body体
400	创建事件错误返回body体

错误码

请参见**错误码**。

4.2.3 删除事件

功能介绍

删除事件

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/workspaces/{workspace_id}/soc/incidents

表 4-193 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

请求参数

表 4-194 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）
content-type	是	String	内容类型

表 4-195 请求 Body 参数

参数	是否必选	参数类型	描述
batch_ids	否	Array of strings	删除事件的ID列表

响应参数

状态码： 200

表 4-196 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-197 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息

参数	参数类型	描述
data	**data** object	批量删除事件返回对象

表 4-198 data

参数	参数类型	描述
error_ids	Array of strings	失败id
success_ids	Array of strings	成功id

状态码： 400

表 4-199 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-200 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

删除id为909494e3-558e-46b6-a9eb-07a8e18ca621的事件

```
{
  "batch_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ]
}
```

响应示例

状态码： 200

事件删除结果

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "data" : {
    "error_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ],
    "success_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ]
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

删除id为909494e3-558e-46b6-a9eb-07a8e18ca621的事件

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class DeleteIncidentSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteIncidentRequest request = new DeleteIncidentRequest();
        request.withWorkspaceId("{workspace_id}");
        DeleteIncidentRequestBody body = new DeleteIncidentRequestBody();
        List<String> listbodyBatchIds = new ArrayList<>();
        listbodyBatchIds.add("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        body.withBatchIds(listbodyBatchIds);
        request.withBody(body);
        try {
            DeleteIncidentResponse response = client.deleteIncident(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

删除id为909494e3-558e-46b6-a9eb-07a8e18ca621的事件

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteIncidentRequest()
        request.workspace_id = "{workspace_id}"
        listBatchIdsbody = [
            "909494e3-558e-46b6-a9eb-07a8e18ca62f"
        ]
        request.body = DeleteIncidentRequestBody(
            batch_ids=listBatchIdsbody
        )
        response = client.delete_incident(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

删除id为909494e3-558e-46b6-a9eb-07a8e18ca621的事件

```
package main

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

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

```
    projectId := "{project_id}"

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteIncidentRequest{}
    request.WorkspaceId = "{workspace_id}"
    var listBatchIdsbody = []string{
        "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    }
    request.Body = &model.DeleteIncidentRequestBody{
        BatchIds: &listBatchIdsbody,
    }
    response, err := client.DeleteIncident(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	事件删除结果
400	删除事件错误返回body体

错误码

请参见**错误码**。

4.2.4 获取事件详情

功能介绍

获取事件详情

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/incidents/{incident_id}

表 4-201 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id
incident_id	是	String	事件ID

请求参数

表 4-202 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）
content-type	是	String	内容类型

响应参数

状态码： 200

表 4-203 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**IncidentDetail** object	事件详情对象

表 4-204 IncidentDetail

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
data_object	**Incident** object	事件实体信息
dataclass_ref	**dataclass_ref** object	数据类对象
format_version	Integer	格式版本
id	String	事件唯一标识，UUID格式，最大36个字符
project_id	String	当前项目的id
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为告警发生时区，无法解析时区的时间，默认时区填东八区
version	Integer	版本
workspace_id	String	当前的工作空间id

表 4-205 Incident

参数	参数类型	描述
version	String	事件对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	String	事件唯一标识，UUID格式，最大36个字符
domain_id	String	数据投递后，被委托用户的domain_id
region_id	String	数据投递后，被委托用户的region_id
workspace_id	String	当前的工作空间id
labels	String	标签，仅展示
environment	**environment** object	事件产生的环境坐标信息
data_source	**data_source** object	首次上报数据源

参数	参数类型	描述
first_observed_time	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	String	事件标题
description	String	事件描述信息
source_url	String	事件URL链接，指向数据源产品中有关当前事件说明的页面
count	Integer	事件发生次数
confidence	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%
severity	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源

参数	参数类型	描述
incident_type	**incident_type** object	事件分类，详细定义参考《告警事件类型定义》
network_list	Array of **network_list** objects	网络信息
resource_list	Array of **resource_list** objects	受影响资源
remediation	**remediation** object	补救措施
verification_state	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	Integer	约束闭环时间：设置风险接受持续时间。单位：小时
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	String	调试字段
actor	String	事件调查员
owner	String	责任人、服务责任人
creator	String	创建人

参数	参数类型	描述
close_reason	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	String	关闭评论
malware	**malware** object	恶意软件
system_info	Object	系统信息
process	Array of **process** objects	进程信息
user_info	Array of **user_info** objects	用户信息
file_info	Array of **file_info** objects	文件信息
system_alert_table	Object	事件管理列表的布局字段

表 4-206 environment

参数	参数类型	描述
vendor_type	String	环境供应商
domain_id	String	租户id
region_id	String	区域id，全局服务global
cross_workspace_id	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	String	项目id， 全局服务默认null

表 4-207 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下: 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	数据源产品所属账号的id

参数	参数类型	描述
project_id	String	数据源产品所属项目的id
region_id	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	String	数据源产品所属公司的名称
product_name	String	数据源产品的名称
product_feature	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	String	检测模块列表

表 4-208 incident_type

参数	参数类型	描述
category	String	类别
incident_type	String	事件类型

表 4-209 network_list

参数	参数类型	描述
direction	String	方向，取值范围：IN | OUT
protocol	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	String	源IP地址
src_port	Integer	源端口，0–65535
src_domain	String	源域名
src_geo	**src_geo** object	源IP的地理位置信息
dest_ip	String	目的IP地址
dest_port	String	目的端口，0–65535
dest_domain	String	目的域名
dest_geo	**dest_geo** object	目标IP的地理位置信息

表 4-210 src_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-211 dest_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-212 resource_list

参数	参数类型	描述
id	String	云服务资源id
name	String	资源名称
type	String	资源类型；引用云RMS type字段
provider	String	云服务名称；引用云RMS provider字段
region_id	String	区域；按照云regionId填写，如cn-north-1等
domain_id	String	资源所属账号ID，UUID格式
project_id	String	资源所属项目ID，UUID格式
ep_id	String	企业项目id
ep_name	String	企业项目名称
tags	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-213 remediation

参数	参数类型	描述
recommendation	String	推荐处理方法
url	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-214 malware

参数	参数类型	描述
malware_family	String	恶意家族
malware_class	String	恶意软件分类

表 4-215 process

参数	参数类型	描述
process_name	String	进程名
process_path	String	进程执行文件路径
process_pid	Integer	进程id
process_uid	Integer	进程用户id
process_cmdline	String	进程命令行
process_parent_name	String	父进程名称
process_parent_path	String	父进程执行文件路径
process_parent_pid	Integer	父进程id
process_parent_uid	Integer	父进程用户id
process_parent_cmdline	String	父进程命令行
process_child_name	String	子进程名称
process_child_path	String	子进程执行文件路径
process_child_pid	Integer	子进程id

参数	参数类型	描述
process_child_uid	Integer	子进程用户id
process_child_cmdline	String	子进程命令行
process_launche_time	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-216 user_info

参数	参数类型	描述
user_id	String	用户uid
user_name	String	用户名称

表 4-217 file_info

参数	参数类型	描述
file_path	String	文件路径/名称
file_content	String	文件内容
file_new_path	String	文件新路径/名称
file_hash	String	文件hash
file_md5	String	文件md5
file_sha256	String	文件sha256
file_attr	String	文件属性

表 4-218 dataclass_ref

参数	参数类型	描述
id	String	数据类唯一标识，UUID格式，最大36个字符
name	String	数据类名称

状态码： 400

表 4-219 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

获取事件详情返回body体

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "data" : {
    "data_object" : {
      "version" : "1.0",
      "environment" : {
        "vendor_type" : "MyXXX",
        "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
      },
      "data_source" : {
        "source_type" : 3,
        "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
      },
      "first_observed_time" : "2021-01-30T23:00:00Z+0800",
      "last_observed_time" : "2021-01-30T23:00:00Z+0800",
      "create_time" : "2021-01-30T23:00:00Z+0800",
      "arrive_time" : "2021-01-30T23:00:00Z+0800",
      "title" : "MyXXX",
      "description" : "This my XXXX",
      "source_url" : "http://xxx",
      "count" : "4",
      "confidence" : 4,
      "severity" : "TIPS",
      "criticality" : 4,
      "incident_type" : { },
      "network_list" : [ {
        "direction" : {
          "IN" : null
        },
        "protocol" : "TCP",
        "src_ip" : "192.168.0.1",
        "src_port" : "1",
        "src_domain" : "xxx",
        "dest_ip" : "192.168.0.1",
        "dest_port" : "1",
        "dest_domain" : "xxx",
        "src_geo" : {
          "latitude" : 90,
```

```
        "longitude" : 180
      },
      "dest_geo" : {
        "latitude" : 90,
        "longitude" : 180
      }
    } ],
    "resource_list" : [ {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "MyXXX",
      "type" : "MyXXX",
      "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "ep_name" : "MyXXX",
      "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    } ],
    "remediation" : {
      "recommendation" : "MyXXX",
      "url" : "MyXXX"
    },
    "verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写
Unknown",
    "handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
    "sla" : 60000,
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "close_time" : "2021-01-30T23:00:00Z+0800",
    "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
    "simulation" : "false",
    "actor" : "刘一博",
    "owner" : "MyXXX",
    "creator" : "MyXXX",
    "close_reason" : "误检;已解决;重复;其他",
    "close_comment" : "误检;已解决;重复;其他",
    "malware" : {
      "malware_family" : "family",
      "malware_class" : "恶意占用内存"
    },
    "system_info" : { },
    "process" : [ {
      "process_name" : "MyXXX",
      "process_path" : "MyXXX",
      "process_pid" : 123,
      "process_uid" : 123,
      "process_cmdline" : "MyXXX"
    } ],
    "user_info" : [ {
      "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "user_name" : "MyXXX"
    } ],
    "file_info" : [ {
      "file_path" : "MyXXX",
      "file_content" : "MyXXX",
      "file_new_path" : "MyXXX",
      "file_hash" : "MyXXX",
      "file_md5" : "MyXXX",
      "file_sha256" : "MyXXX",
      "file_attr" : "MyXXX"
    } ],
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
  },
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
```

```
 }
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowIncidentSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowIncidentRequest request = new ShowIncidentRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withIncidentId("{incident_id}");
        try {
            ShowIncidentResponse response = client.showIncident(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowIncidentRequest()
        request.workspace_id = "{workspace_id}"
        request.incident_id = "{incident_id}"
        response = client.show_incident(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowIncidentRequest{}
```

```
    request.WorkspaceId = "{workspace_id}"
    request.IncidentId = "{incident_id}"
    response, err := client.ShowIncident(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	获取事件详情返回body体
400	获取事件详情错误返回body体

错误码

请参见**错误码**。

4.2.5 更新事件

功能介绍

编辑事件，根据实际修改的属性更新，未修改的列不更新

调用方法

请参见**如何调用API**。

URI

PUT /v1/{project_id}/workspaces/{workspace_id}/soc/incidents/{incident_id}

表 4-220 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id
incident_id	是	String	事件ID

请求参数

表 4-221 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）
content-type	是	String	内容类型

表 4-222 请求 Body 参数

参数	是否必选	参数类型	描述
batch_ids	否	Array of strings	更新事件的ID列表
data_object	否	**Incident** object	事件实体信息

表 4-223 Incident

参数	是否必选	参数类型	描述
version	否	String	事件对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	否	String	事件唯一标识，UUID格式，最大36个字符
domain_id	否	String	数据投递后，被委托用户的domain_id
region_id	否	String	数据投递后，被委托用户的region_id
workspace_id	否	String	当前的工作空间id
labels	否	String	标签，仅展示
environment	否	**environment** object	事件产生的环境坐标信息
data_source	否	**data_source** object	首次上报数据源

参数	是否必选	参数类型	描述
first_observed_time	否	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	否	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
create_time	否	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	否	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	否	String	事件标题
description	否	String	事件描述信息
source_url	否	String	事件URL链接，指向数据源产品中有关当前事件说明的页面
count	否	Integer	事件发生次数
confidence	否	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%
severity	否	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害

参数	是否必选	参数类型	描述
criticality	否	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源
incident_type	否	**incident_type** object	事件分类，详细定义参考《告警事件类型定义》
network_list	否	Array of **network_list** objects	网络信息
resource_list	否	Array of **resource_list** objects	受影响资源
remediation	否	**remediation** object	补救措施
verification_state	否	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	否	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	否	Integer	约束闭环时间：设置风险接受持续时间。单位：小时
update_time	否	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	否	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

参数	是否必选	参数类型	描述
ipdrr_phase	否	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	否	String	调试字段
actor	否	String	事件调查员
owner	否	String	责任人、服务责任人
creator	否	String	创建人
close_reason	否	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	否	String	关闭评论
malware	否	**malware** object	恶意软件
system_info	否	Object	系统信息
process	否	Array of **process** objects	进程信息
user_info	否	Array of **user_info** objects	用户信息
file_info	否	Array of **file_info** objects	文件信息
system_alert_table	否	Object	事件管理列表的布局字段

表 4-224 environment

参数	是否必选	参数类型	描述
vendor_type	否	String	环境供应商
domain_id	否	String	租户id
region_id	否	String	区域id，全局服务global

参数	是否必选	参数类型	描述
cross_workspace_id	否	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	否	String	项目id， 全局服务默认null

表 4-225 data_source

参数	是否必选	参数类型	描述
source_type	否	Integer	数据源类型，取值范围如下： 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	否	String	数据源产品所属账号的id
project_id	否	String	数据源产品所属项目的id
region_id	否	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	否	String	数据源产品所属公司的名称
product_name	否	String	数据源产品的名称
product_feature	否	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	否	String	检测模块列表

表 4-226 incident_type

参数	是否必选	参数类型	描述
category	否	String	类别
incident_type	否	String	事件类型

表 4-227 network_list

参数	是否必选	参数类型	描述
direction	否	String	方向，取值范围：IN | OUT

参数	是否必选	参数类型	描述
protocol	否	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	否	String	源IP地址
src_port	否	Integer	源端口，0–65535
src_domain	否	String	源域名
src_geo	否	**src_geo** object	源IP的地理位置信息
dest_ip	否	String	目的IP地址
dest_port	否	String	目的端口，0–65535
dest_domain	否	String	目的域名
dest_geo	否	**dest_geo** object	目标IP的地理位置信息

表 4-228 src_geo

参数	是否必选	参数类型	描述
latitude	否	Number	纬度
longitude	否	Number	经度
city_code	否	String	城市编码，Beijing | Shanghai
country_code	否	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-229 dest_geo

参数	是否必选	参数类型	描述
latitude	否	Number	纬度
longitude	否	Number	经度
city_code	否	String	城市编码，Beijing | Shanghai

参数	是否必选	参数类型	描述
country_code	否	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-230 resource_list

参数	是否必选	参数类型	描述
id	否	String	云服务资源id
name	否	String	资源名称
type	否	String	资源类型；引用云RMS type字段
provider	否	String	云服务名称；引用云RMS provider字段
region_id	否	String	区域；按照云regionId填写，如cn-north-1等
domain_id	否	String	资源所属账号ID，UUID格式
project_id	否	String	资源所属项目ID，UUID格式
ep_id	否	String	企业项目id
ep_name	否	String	企业项目名称
tags	否	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-231 remediation

参数	是否必选	参数类型	描述
recommendation	否	String	推荐处理方法
url	否	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-232 malware

参数	是否必选	参数类型	描述
malware_family	否	String	恶意家族
malware_class	否	String	恶意软件分类

表 4-233 process

参数	是否必选	参数类型	描述
process_name	否	String	进程名
process_path	否	String	进程执行文件路径
process_pid	否	Integer	进程id
process_uid	否	Integer	进程用户id
process_cmdline	否	String	进程命令行
process_parent_name	否	String	父进程名称
process_parent_path	否	String	父进程执行文件路径
process_parent_pid	否	Integer	父进程id
process_parent_uid	否	Integer	父进程用户id
process_parent_cmdline	否	String	父进程命令行
process_child_name	否	String	子进程名称
process_child_path	否	String	子进程执行文件路径
process_child_pid	否	Integer	子进程id
process_child_uid	否	Integer	子进程用户id
process_child_cmdline	否	String	子进程命令行

参数	是否必选	参数类型	描述
process_launche_time	否	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	否	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-234 user_info

参数	是否必选	参数类型	描述
user_id	否	String	用户uid
user_name	否	String	用户名称

表 4-235 file_info

参数	是否必选	参数类型	描述
file_path	否	String	文件路径/名称
file_content	否	String	文件内容
file_new_path	否	String	文件新路径/名称
file_hash	否	String	文件hash
file_md5	否	String	文件md5
file_sha256	否	String	文件sha256
file_attr	否	String	文件属性

响应参数

状态码： 200

表 4-236 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-237 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**IncidentDetail** object	事件详情对象

表 4-238 IncidentDetail

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
data_object	**Incident** object	事件实体信息
dataclass_ref	**dataclass_ref** object	数据类对象
format_version	Integer	格式版本
id	String	事件唯一标识，UUID格式，最大36个字符
project_id	String	当前项目的id
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为告警发生时区，无法解析时区的时间，默认时区填东八区
version	Integer	版本
workspace_id	String	当前的工作空间id

表 4-239 Incident

参数	参数类型	描述
version	String	事件对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	String	事件唯一标识，UUID格式，最大36个字符
domain_id	String	数据投递后，被委托用户的domain_id
region_id	String	数据投递后，被委托用户的region_id

参数	参数类型	描述
workspace_id	String	当前的工作空间id
labels	String	标签，仅展示
environment	**environment** object	事件产生的环境坐标信息
data_source	**data_source** object	首次上报数据源
first_observed_time	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	String	事件标题
description	String	事件描述信息
source_url	String	事件URL链接，指向数据源产品中有关当前事件说明的页面
count	Integer	事件发生次数
confidence	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%

参数	参数类型	描述
severity	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源
incident_type	**incident_type** object	事件分类，详细定义参考《告警事件类型定义》
network_list	Array of **network_list** objects	网络信息
resource_list	Array of **resource_list** objects	受影响资源
remediation	**remediation** object	补救措施
verification_state	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	Integer	约束闭环时间：设置风险接受持续时间。单位：小时

参数	参数类型	描述
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	String	调试字段
actor	String	事件调查员
owner	String	责任人、服务责任人
creator	String	创建人
close_reason	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	String	关闭评论
malware	**malware** object	恶意软件
system_info	Object	系统信息
process	Array of **process** objects	进程信息
user_info	Array of **user_info** objects	用户信息
file_info	Array of **file_info** objects	文件信息
system_alert_table	Object	事件管理列表的布局字段

表 4-240 environment

参数	参数类型	描述
vendor_type	String	环境供应商

参数	参数类型	描述
domain_id	String	租户id
region_id	String	区域id，全局服务global
cross_workspace_id	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	String	项目id， 全局服务默认null

表 4-241 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下： 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	数据源产品所属账号的id
project_id	String	数据源产品所属项目的id
region_id	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	String	数据源产品所属公司的名称
product_name	String	数据源产品的名称
product_feature	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	String	检测模块列表

表 4-242 incident_type

参数	参数类型	描述
category	String	类别
incident_type	String	事件类型

表 4-243 network_list

参数	参数类型	描述
direction	String	方向，取值范围：IN | OUT

参数	参数类型	描述
protocol	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	String	源IP地址
src_port	Integer	源端口，0–65535
src_domain	String	源域名
src_geo	**src_geo** object	源IP的地理位置信息
dest_ip	String	目的IP地址
dest_port	String	目的端口，0–65535
dest_domain	String	目的域名
dest_geo	**dest_geo** object	目标IP的地理位置信息

表 4-244 src_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-245 dest_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-246 resource_list

参数	参数类型	描述
id	String	云服务资源id
name	String	资源名称
type	String	资源类型；引用云RMS type字段
provider	String	云服务名称；引用云RMS provider字段
region_id	String	区域；按照云regionId填写，如cn-north-1等
domain_id	String	资源所属账号ID，UUID格式
project_id	String	资源所属项目ID，UUID格式
ep_id	String	企业项目id
ep_name	String	企业项目名称
tags	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-247 remediation

参数	参数类型	描述
recommendation	String	推荐处理方法
url	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-248 malware

参数	参数类型	描述
malware_family	String	恶意家族
malware_class	String	恶意软件分类

表 4-249 process

参数	参数类型	描述
process_name	String	进程名

参数	参数类型	描述
process_path	String	进程执行文件路径
process_pid	Integer	进程id
process_uid	Integer	进程用户id
process_cmdline	String	进程命令行
process_parent_name	String	父进程名称
process_parent_path	String	父进程执行文件路径
process_parent_pid	Integer	父进程id
process_parent_uid	Integer	父进程用户id
process_parent_cmdline	String	父进程命令行
process_child_name	String	子进程名称
process_child_path	String	子进程执行文件路径
process_child_pid	Integer	子进程id
process_child_uid	Integer	子进程用户id
process_child_cmdline	String	子进程命令行
process_launche_time	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-250 user_info

参数	参数类型	描述
user_id	String	用户uid
user_name	String	用户名称

表 4-251 file_info

参数	参数类型	描述
file_path	String	文件路径/名称
file_content	String	文件内容
file_new_path	String	文件新路径/名称
file_hash	String	文件hash
file_md5	String	文件md5
file_sha256	String	文件sha256
file_attr	String	文件属性

表 4-252 dataclass_ref

参数	参数类型	描述
id	String	数据类唯一标识，UUID格式，最大36个字符
name	String	数据类名称

状态码： 400

表 4-253 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-254 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

更新一条事件，事件标题为MyXXX，URL链接为http://xxx，发生次数为4次，置信度为4。

```
{
  "data_object" : {
```

```
  "version" : "1.0",
  "environment" : {
    "vendor_type" : "MyXXX",
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  },
  "data_source" : {
    "source_type" : 3,
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  },
  "first_observed_time" : "2021-01-30T23:00:00Z+0800",
  "last_observed_time" : "2021-01-30T23:00:00Z+0800",
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "arrive_time" : "2021-01-30T23:00:00Z+0800",
  "title" : "MyXXX",
  "description" : "This my XXXX",
  "source_url" : "http://xxx",
  "count" : 4,
  "confidence" : 4,
  "severity" : "TIPS",
  "criticality" : 4,
  "incident_type" : { },
  "network_list" : [ {
    "direction" : {
      "IN" : null
    },
    "protocol" : "TCP",
    "src_ip" : "192.168.0.1",
    "src_port" : "1",
    "src_domain" : "xxx",
    "dest_ip" : "192.168.0.1",
    "dest_port" : "1",
    "dest_domain" : "xxx",
    "src_geo" : {
      "latitude" : 90,
      "longitude" : 180
    },
    "dest_geo" : {
      "latitude" : 90,
      "longitude" : 180
    }
  } ],
  "resource_list" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "type" : "MyXXX",
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "ep_name" : "MyXXX",
    "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  } ],
  "remediation" : {
    "recommendation" : "MyXXX",
    "url" : "MyXXX"
  },
  "verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写Unknown",
  "handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
  "sla" : 60000,
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "close_time" : "2021-01-30T23:00:00Z+0800",
  "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
  "simulation" : "false",
  "actor" : "刘一博",
```

```
  "owner" : "MyXXX",
  "creator" : "MyXXX",
  "close_reason" : "误检;已解决;重复;其他",
  "close_comment" : "误检;已解决;重复;其他",
  "malware" : {
    "malware_family" : "family",
    "malware_class" : "恶意占用内存"
  },
  "system_info" : { },
  "process" : [ {
    "process_name" : "MyXXX",
    "process_path" : "MyXXX",
    "process_pid" : 123,
    "process_uid" : 123,
    "process_cmdline" : "MyXXX"
  } ],
  "user_info" : [ {
    "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "user_name" : "MyXXX"
  } ],
  "file_info" : [ {
    "file_path" : "MyXXX",
    "file_content" : "MyXXX",
    "file_new_path" : "MyXXX",
    "file_hash" : "MyXXX",
    "file_md5" : "MyXXX",
    "file_sha256" : "MyXXX",
    "file_attr" : "MyXXX"
  } ],
  "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
 }
}
```

响应示例

状态码： 200

更新事件返回body体

```
{
 "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
 "message" : "Error message",
 "data" : {
  "data_object" : {
   "version" : "1.0",
   "environment" : {
    "vendor_type" : "MyXXX",
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
   },
   "data_source" : {
    "source_type" : 3,
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
   },
   "first_observed_time" : "2021-01-30T23:00:00Z+0800",
   "last_observed_time" : "2021-01-30T23:00:00Z+0800",
   "create_time" : "2021-01-30T23:00:00Z+0800",
   "arrive_time" : "2021-01-30T23:00:00Z+0800",
   "title" : "MyXXX",
   "description" : "This my XXXX",
   "source_url" : "http://xxx",
   "count" : 4,
   "confidence" : 4,
   "severity" : "TIPS",
   "criticality" : 4,
```

```
    "incident_type" : { },
    "network_list" : [ {
      "direction" : {
        "IN" : null
      },
      "protocol" : "TCP",
      "src_ip" : "192.168.0.1",
      "src_port" : "1",
      "src_domain" : "xxx",
      "dest_ip" : "192.168.0.1",
      "dest_port" : "1",
      "dest_domain" : "xxx",
      "src_geo" : {
        "latitude" : 90,
        "longitude" : 180
      },
      "dest_geo" : {
        "latitude" : 90,
        "longitude" : 180
      }
    } ],
    "resource_list" : [ {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "MyXXX",
      "type" : "MyXXX",
      "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "ep_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "ep_name" : "MyXXX",
      "tags" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    } ],
    "remediation" : {
      "recommendation" : "MyXXX",
      "url" : "MyXXX"
    },
    "verification_state" : "Unknown – 未知，True_Positive – 确认，False_Positive – 误报。默认填写
Unknown",
    "handle_status" : "Open – 打开，Block – 阻塞，Closed – 关闭。默认填写Open",
    "sla" : 60000,
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "close_time" : "2021-01-30T23:00:00Z+0800",
    "ipdrr_phase" : "Prepartion|Detection and Analysis|Containm，Eradication& Recovery| Post-Incident-
Activity",
    "simulation" : "false",
    "actor" : "刘一博",
    "owner" : "MyXXX",
    "creator" : "MyXXX",
    "close_reason" : "误检;已解决;重复;其他",
    "close_comment" : "误检;已解决;重复;其他",
    "malware" : {
      "malware_family" : "family",
      "malware_class" : "恶意占用内存"
    },
    "system_info" : { },
    "process" : [ {
      "process_name" : "MyXXX",
      "process_path" : "MyXXX",
      "process_pid" : 123,
      "process_uid" : 123,
      "process_cmdline" : "MyXXX"
    } ],
    "user_info" : [ {
      "user_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "user_name" : "MyXXX"
    } ],
    "file_info" : [ {
      "file_path" : "MyXXX",
      "file_content" : "MyXXX",
```



```
      "file_new_path" : "MyXXX",
      "file_hash" : "MyXXX",
      "file_md5" : "MyXXX",
      "file_sha256" : "MyXXX",
      "file_attr" : "MyXXX"
    } ],
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620"
  },
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "update_time" : "2021-01-30T23:00:00Z+0800",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
 }
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新一条事件，事件标题为MyXXX，URL链接为http://xxx，发生次数为4次，置信度为4。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class ChangeIncidentSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ChangeIncidentRequest request = new ChangeIncidentRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withIncidentId("{incident_id}");
        ChangeIncidentRequestBody body = new ChangeIncidentRequestBody();
        List<IncidentFileInfo> listDataObjectFileInfo = new ArrayList<>();
        listDataObjectFileInfo.add(
            new IncidentFileInfo()
```

```
        .withFilePath("MyXXX")
        .withFileContent("MyXXX")
        .withFileNewPath("MyXXX")
        .withFileHash("MyXXX")
        .withFileMd5("MyXXX")
        .withFileSha256("MyXXX")
        .withFileAttr("MyXXX")
);
List<IncidentUserInfo> listDataObjectUserInfo = new ArrayList<>();
listDataObjectUserInfo.add(
    new IncidentUserInfo()
        .withUserId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withUserName("MyXXX")
);
List<IncidentProcess> listDataObjectProcess = new ArrayList<>();
listDataObjectProcess.add(
    new IncidentProcess()
        .withProcessName("MyXXX")
        .withProcessPath("MyXXX")
        .withProcessPid(123)
        .withProcessUid(123)
        .withProcessCmdline("MyXXX")
);
IncidentMalware malwareDataObject = new IncidentMalware();
malwareDataObject.withMalwareFamily("family")
    .withMalwareClass("恶意占用内存");
IncidentRemediation remediationDataObject = new IncidentRemediation();
remediationDataObject.withRecommendation("MyXXX")
    .withUrl("MyXXX");
List<IncidentResourceList> listDataObjectResourceList = new ArrayList<>();
listDataObjectResourceList.add(
    new IncidentResourceList()
        .withId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withName("MyXXX")
        .withType("MyXXX")
        .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withEpId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
        .withEpName("MyXXX")
        .withTags("909494e3-558e-46b6-a9eb-07a8e18ca62f")
);
IncidentDestGeo destGeoNetworkList = new IncidentDestGeo();
destGeoNetworkList.withLatitude(java.math.BigDecimal.valueOf(90))
    .withLongitude(java.math.BigDecimal.valueOf(180));
IncidentSrcGeo srcGeoNetworkList = new IncidentSrcGeo();
srcGeoNetworkList.withLatitude(java.math.BigDecimal.valueOf(90))
    .withLongitude(java.math.BigDecimal.valueOf(180));
List<IncidentNetworkList> listDataObjectNetworkList = new ArrayList<>();
listDataObjectNetworkList.add(
    new IncidentNetworkList()
        .withDirection(IncidentNetworkList.DirectionEnum.fromValue("{}"))
        .withProtocol("TCP")
        .withSrcIp("192.168.0.1")
        .withSrcPort(1)
        .withSrcDomain("xxx")
        .withSrcGeo(srcGeoNetworkList)
        .withDestIp("192.168.0.1")
        .withDestPort("1")
        .withDestDomain("xxx")
        .withDestGeo(destGeoNetworkList)
);
IncidentDataSource dataSourceDataObject = new IncidentDataSource();
dataSourceDataObject.withSourceType(3)
    .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
    .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
    .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
IncidentEnvironment environmentDataObject = new IncidentEnvironment();
environmentDataObject.withVendorType("MyXXX")
```

```
            .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        Incident dataObjectbody = new Incident();
        dataObjectbody.withVersion("1.0")
            .withId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withWorkspaceId("909494e3-558e-46b6-a9eb-07a8e18ca620")
            .withEnvironment(environmentDataObject)
            .withDataSource(dataSourceDataObject)
            .withFirstObservedTime("2021-01-30T23:00:00Z+0800")
            .withLastObservedTime("2021-01-30T23:00:00Z+0800")
            .withCreateTime("2021-01-30T23:00:00Z+0800")
            .withArriveTime("2021-01-30T23:00:00Z+0800")
            .withTitle("MyXXX")
            .withDescription("This my XXXX")
            .withSourceUrl("http://xxx")
            .withCount(4)
            .withConfidence(4)
            .withSeverity(Incident.SeverityEnum.fromValue("TIPS"))
            .withCriticality(4)
            .withNetworkList(listDataObjectNetworkList)
            .withResourceList(listDataObjectResourceList)
            .withRemediation(remediationDataObject)
            .withVerificationState(Incident.VerificationStateEnum.fromValue("Unknown – 未知,True_Positive – 确
认,False_Positive – 误报。默认填写Unknown"))
            .withHandleStatus(Incident.HandleStatusEnum.fromValue("Open – 打开,Block – 阻塞,Closed – 关
闭。默认填写Open"))
            .withSla(60000)
            .withUpdateTime("2021-01-30T23:00:00Z+0800")
            .withCloseTime("2021-01-30T23:00:00Z+0800")
            .withIpdrrPhase(Incident.IpdrrPhaseEnum.fromValue("Prepartion|Detection and Analysis|
Containm,Eradication& Recovery| Post-Incident-Activity"))
            .withSimulation("false")
            .withActor("刘一博")
            .withOwner("MyXXX")
            .withCreator("MyXXX")
            .withCloseReason(Incident.CloseReasonEnum.fromValue("误检;已解决;重复;其他"))
            .withCloseComment("误检;已解决;重复;其他")
            .withMalware(malwareDataObject)
            .withSystemInfo(new Object())
            .withProcess(listDataObjectProcess)
            .withUserInfo(listDataObjectUserInfo)
            .withFileInfo(listDataObjectFileInfo);
        body.withDataObject(dataObjectbody);
        request.withBody(body);
        try {
            ChangeIncidentResponse response = client.changeIncident(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

更新一条事件，事件标题为MyXXX，URL链接为http://xxx，发生次数为4次，置信度为4。

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



```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ChangeIncidentRequest()
        request.workspace_id = "{workspace_id}"
        request.incident_id = "{incident_id}"
        listFileInfoDataObject = [
            IncidentFileInfo(
                file_path="MyXXX",
                file_content="MyXXX",
                file_new_path="MyXXX",
                file_hash="MyXXX",
                file_md5="MyXXX",
                file_sha256="MyXXX",
                file_attr="MyXXX"
            )
        ]
        listUserInfoDataObject = [
            IncidentUserInfo(
                user_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
                user_name="MyXXX"
            )
        ]
        listProcessDataObject = [
            IncidentProcess(
                process_name="MyXXX",
                process_path="MyXXX",
                process_pid=123,
                process_uid=123,
                process_cmdline="MyXXX"
            )
        ]
        malwareDataObject = IncidentMalware(
            malware_family="family",
            malware_class="恶意占用内存"
        )
        remediationDataObject = IncidentRemediation(
            recommendation="MyXXX",
            url="MyXXX"
        )
        listResourceListDataObject = [
            IncidentResourceList(
                id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
                name="MyXXX",
                type="MyXXX",
                region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
                domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
                project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
```

```
            ep_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            ep_name="MyXXX",
            tags="909494e3-558e-46b6-a9eb-07a8e18ca62f"
        )
    ]
    destGeoNetworkList = IncidentDestGeo(
        latitude=90,
        longitude=180
    )
    srcGeoNetworkList = IncidentSrcGeo(
        latitude=90,
        longitude=180
    )
    listNetworkListDataObject = [
        IncidentNetworkList(
            direction="{}",
            protocol="TCP",
            src_ip="192.168.0.1",
            src_port=1,
            src_domain="xxx",
            src_geo=srcGeoNetworkList,
            dest_ip="192.168.0.1",
            dest_port="1",
            dest_domain="xxx",
            dest_geo=destGeoNetworkList
        )
    ]
    dataSourceDataObject = IncidentDataSource(
        source_type=3,
        domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f"
    )
    environmentDataObject = IncidentEnvironment(
        vendor_type="MyXXX",
        domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f"
    )
    dataObjectbody = Incident(
        version="1.0",
        id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
        workspace_id="909494e3-558e-46b6-a9eb-07a8e18ca620",
        environment=environmentDataObject,
        data_source=dataSourceDataObject,
        first_observed_time="2021-01-30T23:00:00Z+0800",
        last_observed_time="2021-01-30T23:00:00Z+0800",
        create_time="2021-01-30T23:00:00Z+0800",
        arrive_time="2021-01-30T23:00:00Z+0800",
        title="MyXXX",
        description="This my XXXX",
        source_url="http://xxx",
        count=4,
        confidence=4,
        severity="TIPS",
        criticality=4,
        network_list=listNetworkListDataObject,
        resource_list=listResourceListDataObject,
        remediation=remediationDataObject,
        verification_state="Unknown – 未知,True_Positive – 确认,False_Positive – 误报。默认填写Unknown",
        handle_status="Open – 打开,Block – 阻塞,Closed – 关闭。默认填写Open",
        sla=60000,
        update_time="2021-01-30T23:00:00Z+0800",
        close_time="2021-01-30T23:00:00Z+0800",
        ipdrr_phase="Prepartion|Detection and Analysis|Containm,Eradication& Recovery| Post-Incident-
Activity",
        simulation="false",
        actor="刘一博",
        owner="MyXXX",
```

```
            creator="MyXXX",
            close_reason="误检;已解决;重复;其他",
            close_comment="误检;已解决;重复;其他",
            malware=malwareDataObject,
            system_info={},
            process=listProcessDataObject,
            user_info=listUserInfoDataObject,
            file_info=listFileInfoDataObject
        )
        request.body = ChangeIncidentRequestBody(
            data_object=dataObjectbody
        )
        response = client.change_incident(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

更新一条事件，事件标题为MyXXX，URL链接为http://xxx，发生次数为4次，置信度为4。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeIncidentRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.IncidentId = "{incident_id}"
    filePathFileInfo:= "MyXXX"
    fileContentFileInfo:= "MyXXX"
    fileNewPathFileInfo:= "MyXXX"
    fileHashFileInfo:= "MyXXX"
    fileMd5FileInfo:= "MyXXX"
    fileSha256FileInfo:= "MyXXX"
    fileAttrFileInfo:= "MyXXX"
    var listFileInfoDataObject = []model.IncidentFileInfo{
        {
```

```
        FilePath: &filePathFileInfo,
        FileContent: &fileContentFileInfo,
        FileNewPath: &fileNewPathFileInfo,
        FileHash: &fileHashFileInfo,
        FileMd5: &fileMd5FileInfo,
        FileSha256: &fileSha256FileInfo,
        FileAttr: &fileAttrFileInfo,
    },
}
userIdUserInfo:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
userNameUserInfo:= "MyXXX"
var listUserInfoDataObject = []model.IncidentUserInfo{
    {
        UserId: &userIdUserInfo,
        UserName: &userNameUserInfo,
    },
}
processNameProcess:= "MyXXX"
processPathProcess:= "MyXXX"
processPidProcess:= int32(123)
processUidProcess:= int32(123)
processCmdlineProcess:= "MyXXX"
var listProcessDataObject = []model.IncidentProcess{
    {
        ProcessName: &processNameProcess,
        ProcessPath: &processPathProcess,
        ProcessPid: &processPidProcess,
        ProcessUid: &processUidProcess,
        ProcessCmdline: &processCmdlineProcess,
    },
}
malwareFamilyMalware:= "family"
malwareClassMalware:= "恶意占用内存"
malwareDataObject := &model.IncidentMalware{
    MalwareFamily: &malwareFamilyMalware,
    MalwareClass: &malwareClassMalware,
}
recommendationRemediation:= "MyXXX"
urlRemediation:= "MyXXX"
remediationDataObject := &model.IncidentRemediation{
    Recommendation: &recommendationRemediation,
    Url: &urlRemediation,
}
idResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
nameResourceList:= "MyXXX"
typeResourceList:= "MyXXX"
regionIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
domainIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
projectIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
epIdResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
epNameResourceList:= "MyXXX"
tagsResourceList:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
var listResourceListDataObject = []model.IncidentResourceList{
    {
        Id: &idResourceList,
        Name: &nameResourceList,
        Type: &typeResourceList,
        RegionId: &regionIdResourceList,
        DomainId: &domainIdResourceList,
        ProjectId: &projectIdResourceList,
        EpId: &epIdResourceList,
        EpName: &epNameResourceList,
        Tags: &tagsResourceList,
    },
}
latitudeDestGeo:= float32(90)
longitudeDestGeo:= float32(180)
destGeoNetworkList := &model.IncidentDestGeo{
    Latitude: &latitudeDestGeo,
```

```
    Longitude: &longitudeDestGeo,
  }
  latitudeSrcGeo:= float32(90)
  longitudeSrcGeo:= float32(180)
  srcGeoNetworkList := &model.IncidentSrcGeo{
    Latitude: &latitudeSrcGeo,
    Longitude: &longitudeSrcGeo,
  }
  directionNetworkList:= model.GetIncidentNetworkListDirectionEnum().{}
  protocolNetworkList:= "TCP"
  srcIpNetworkList:= "192.168.0.1"
  srcPortNetworkList:= int32(1)
  srcDomainNetworkList:= "xxx"
  destIpNetworkList:= "192.168.0.1"
  destPortNetworkList:= "1"
  destDomainNetworkList:= "xxx"
  var listNetworkListDataObject = []model.IncidentNetworkList{
    {
      Direction: &directionNetworkList,
      Protocol: &protocolNetworkList,
      SrcIp: &srcIpNetworkList,
      SrcPort: &srcPortNetworkList,
      SrcDomain: &srcDomainNetworkList,
      SrcGeo: srcGeoNetworkList,
      DestIp: &destIpNetworkList,
      DestPort: &destPortNetworkList,
      DestDomain: &destDomainNetworkList,
      DestGeo: destGeoNetworkList,
    },
  }
  sourceTypeDataSource:= int32(3)
  domainIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  projectIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  regionIdDataSource:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  dataSourceDataObject := &model.IncidentDataSource{
    SourceType: &sourceTypeDataSource,
    DomainId: &domainIdDataSource,
    ProjectId: &projectIdDataSource,
    RegionId: &regionIdDataSource,
  }
  vendorTypeEnvironment:= "MyXXX"
  domainIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  regionIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  projectIdEnvironment:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  environmentDataObject := &model.IncidentEnvironment{
    VendorType: &vendorTypeEnvironment,
    DomainId: &domainIdEnvironment,
    RegionId: &regionIdEnvironment,
    ProjectId: &projectIdEnvironment,
  }
  versionDataObject:= "1.0"
  idDataObject:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  workspaceIdDataObject:= "909494e3-558e-46b6-a9eb-07a8e18ca620"
  firstObservedTimeDataObject:= "2021-01-30T23:00:00Z+0800"
  lastObservedTimeDataObject:= "2021-01-30T23:00:00Z+0800"
  createTimeDataObject:= "2021-01-30T23:00:00Z+0800"
  arriveTimeDataObject:= "2021-01-30T23:00:00Z+0800"
  titleDataObject:= "MyXXX"
  descriptionDataObject:= "This my XXXX"
  sourceUrlDataObject:= "http://xxx"
  countDataObject:= int32(4)
  confidenceDataObject:= int32(4)
  severityDataObject:= model.GetIncidentSeverityEnum().TIPS
  criticalityDataObject:= int32(4)
  verificationStateDataObject:= model.GetIncidentVerificationStateEnum().UNKNOWN_-_未
知,TRUE_POSITIVE_-_确认,FALSE_POSITIVE_-_误报。默认填写UNKNOWN
  handleStatusDataObject:= model.GetIncidentHandleStatusEnum().OPEN_-_打开,BLOCK_-_阻塞,CLOSED_-
_关闭。默认填写OPEN
  slaDataObject:= int32(60000)
```

```
    updateTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    closeTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    ipdrrPhaseDataObject:= model.GetIncidentIpdrrPhaseEnum().PREPARTION|DETECTION_AND_ANALYSIS|
CONTAINM,ERADICATION&_RECOVERY|_POST_INCIDENT_ACTIVITY
    simulationDataObject:= "false"
    actorDataObject:= "刘一博"
    ownerDataObject:= "MyXXX"
    creatorDataObject:= "MyXXX"
    closeReasonDataObject:= model.GetIncidentCloseReasonEnum().误检;已解决;重复;其他
    closeCommentDataObject:= "误检;已解决;重复;其他"
    var systemInfoDataObject interface{} = make(map[string]string)
    dataObjectbody := &model.Incident{
        Version: &versionDataObject,
        Id: &idDataObject,
        WorkspaceId: &workspaceIdDataObject,
        Environment: environmentDataObject,
        DataSource: dataSourceDataObject,
        FirstObservedTime: &firstObservedTimeDataObject,
        LastObservedTime: &lastObservedTimeDataObject,
        CreateTime: &createTimeDataObject,
        ArriveTime: &arriveTimeDataObject,
        Title: &titleDataObject,
        Description: &descriptionDataObject,
        SourceUrl: &sourceUrlDataObject,
        Count: &countDataObject,
        Confidence: &confidenceDataObject,
        Severity: &severityDataObject,
        Criticality: &criticalityDataObject,
        NetworkList: &listNetworkListDataObject,
        ResourceList: &listResourceListDataObject,
        Remediation: remediationDataObject,
        VerificationState: &verificationStateDataObject,
        HandleStatus: &handleStatusDataObject,
        Sla: &slaDataObject,
        UpdateTime: &updateTimeDataObject,
        CloseTime: &closeTimeDataObject,
        IpdrrPhase: &ipdrrPhaseDataObject,
        Simulation: &simulationDataObject,
        Actor: &actorDataObject,
        Owner: &ownerDataObject,
        Creator: &creatorDataObject,
        CloseReason: &closeReasonDataObject,
        CloseComment: &closeCommentDataObject,
        Malware: malwareDataObject,
        SystemInfo: &systemInfoDataObject,
        Process: &listProcessDataObject,
        UserInfo: &listUserInfoDataObject,
        FileInfo: &listFileInfoDataObject,
    }
    request.Body = &model.ChangeIncidentRequestBody{
        DataObject: dataObjectbody,
    }
    response, err := client.ChangeIncident(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	更新事件返回body体
400	更新事件错误返回body体

错误码

请参见**错误码**。

4.3 威胁情报管理

4.3.1 查询威胁情报列表

功能介绍

查询威胁情报列表

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/indicators/search

表 4-255 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

请求参数

表 4-256 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	租户的Token
content-type	是	String	application/json;charset=UTF-8

表 4-257 请求 Body 参数

参数	是否必选	参数类型	描述
ids	否	Array of strings	威胁情报ID列表
dataclass_id	否	String	数据类ID
condition	是	**condition** object	搜索条件表达式
offset	是	Integer	request offset, from 0
limit	是	Integer	request limit size
sort_by	否	String	sort by property, create_time.
from_date	否	String	查询起始时间，例如：2024-01-20T00:00:00.000Z +0800
to_date	否	String	查询截止时间，例如：2024-01-26T23:59:59.999Z +0800

表 4-258 condition

参数	是否必选	参数类型	描述
conditions	否	Array of **conditions** objects	表达式列表
logics	否	Array of strings	表达式名称列表

表 4-259 conditions

参数	是否必选	参数类型	描述
name	否	String	表达式名称
data	否	Array of strings	表达式内容列表

响应参数

状态码： 200

表 4-260 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为:request_uuid-timestamp-hostname.

表 4-261 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
total	Integer	总数
data	Array of **IndicatorDetail** objects	指标列表数据

表 4-262 IndicatorDetail

参数	参数类型	描述
id	String	威胁情报ID
name	String	威胁情报名称
data_object	**IndicatorDataObjectDetail** object	情报详情
workspace_id	String	工作空间ID
project_id	String	项目ID
dataclass_ref	**DataClassRefPojo** object	数据类对象信息
create_time	String	创建时间
update_time	String	更新时间

表 4-263 IndicatorDataObjectDetail

参数	参数类型	描述
indicator_type	**indicator_type** object	情报类型对象
value	String	值，如：ip url domain等

参数	参数类型	描述
update_time	String	更新时间
create_time	String	创建时间
environment	**environment** object	环境信息
data_source	**data_source** object	数据源信息
first_report_time	String	首次发生时间
is_deleted	Boolean	是否删除
last_report_time	String	最近发生时间
granular_marking	Integer	粒度（保密等级），由高到低：1（首次发现）、2（自产数据）、3（需购买）、4（外网直接查询）
name	String	名称
id	String	威胁情报ID
project_id	String	项目ID
revoked	Boolean	是否作废
status	String	状态， Open--打开，Closed--关闭, Revoked--作废
verdict	String	威胁度， Black--黑,White--白，Gray--灰
workspace_id	String	工作空间ID
confidence	Integer	置信度，取值范围是80-100

表 4-264 indicator_type

参数	参数类型	描述
indicator_type	String	情报类型
id	String	情报类型ID

表 4-265 environment

参数	参数类型	描述
vendor_type	String	环境供应商

参数	参数类型	描述
domain_id	String	租户ID
region_id	String	区域ID
project_id	String	项目ID

表 4-266 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下：1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	租户ID
project_id	String	项目ID
region_id	String	区域ID

表 4-267 DataClassRefPojo

参数	参数类型	描述
id	String	数据类ID
name	String	数据类名称

状态码： 400

表 4-268 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-269 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

查询id为id1、id2，名称为威胁情报名称，类型为DATA_SOURCE，数据类id为28f61af50fc9452aa0ed5ea25c3cc3d3的威胁情报列表，偏移量为0，查询上限10条，根据create_time排序

```
{
  "ids" : [ "id1", "id2" ],
  "dataclass_id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
  "condition" : {
    "conditions" : [ {
      "name" : "name",
      "data" : [ "name", "=", "威胁情报名称" ]
    } ],
    "logics" : [ "title" ]
  },
  "offset" : 0,
  "limit" : 10,
  "sort_by" : "create_time",
  "from_date" : "2024-01-20T00:00:00.000Z+0800",
  "to_date" : "2024-01-26T23:59:59.999Z+0800"
}
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : "00000000",
  "data" : [ {
    "create_time" : "2023-07-24T20:54:19Z+0800",
    "data_object" : {
      "indicator_type" : {
        "indicator_type" : "ipv6",
        "id" : "ac794b2dfab9fe8c0676587301a636d3"
      },
      "revoked" : false,
      "workspace_id" : "d5baeef8-3e75-4e91-9826-fb208ac58987",
      "update_time" : "2023-07-24T20:54:19.038Z+0800",
      "project_id" : "15645222e8744afa985c93dab6341da6",
      "first_report_time" : "2023-07-31T20:54:12.000Z+0800",
      "id" : "ff61d1f8-0de4-4077-9e9b-e312f6829c6d",
      "granular_marking" : 1,
      "value" : "{}",
      "create_time" : "2023-07-24T20:54:19.038Z+0800",
      "confidence" : 80,
      "last_report_time" : "2023-07-25T20:54:15.000Z+0800",
      "data_source" : {
        "domain_id" : "ac7438b990ef4a37b741004eb45e8bf4",
        "project_id" : "15645222e8744afa985c93dab6341da6",
        "region_id" : "cn-XXX-7",
        "source_type" : 1
      },
      "environment" : {
        "domain_id" : "ac7438b990ef4a37b741004eb45e8bf4",
        "project_id" : "15645222e8744afa985c93dab6341da6",
        "region_id" : "cn-xxx-7",
        "vendor_type" : "xxx"
      },
      "verdict" : "Black",
      "name" : "test",
      "status" : "Open"
    },
    "dataclass_ref" : {
      "id" : "97ccf890-7480-31f6-a961-cf8da1f2f040",
```

```
    "name" : "name"
  },
  "id" : "ff61d1f8-0de4-4077-9e9b-e312f6829c6d",
  "update_time" : "2023-07-24T20:54:19Z+0800"
 } ],
 "message" : "",
 "total" : 2
}
```

SDK 代码示例

SDK代码示例如下。

Java

查询id为id1、id2，名称为威胁情报名称，类型为DATA_SOURCE，数据类id为28f61af50fc9452aa0ed5ea25c3cc3d3的威胁情报列表，偏移量为0，查询上限10条，根据create_time排序

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class ListIndicatorsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListIndicatorsRequest request = new ListIndicatorsRequest();
        request.withWorkspaceId("{workspace_id}");
        IndicatorListSearchRequest body = new IndicatorListSearchRequest();
        List<String> listConditionLogics = new ArrayList<>();
        listConditionLogics.add("title");
        List<String> listConditionsData = new ArrayList<>();
        listConditionsData.add("name");
        listConditionsData.add("=");
        listConditionsData.add("威胁情报名称");
        List<IndicatorListSearchRequestConditionConditions> listConditionConditions = new ArrayList<>();
        listConditionConditions.add(
            new IndicatorListSearchRequestConditionConditions()
                .withName("name")
```

```
            .withData(listConditionsData)
        );
        IndicatorListSearchRequestCondition conditionbody = new IndicatorListSearchRequestCondition();
        conditionbody.withConditions(listConditionConditions)
            .withLogics(listConditionLogics);
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("id1");
        listbodyIds.add("id2");
        body.withToDate("2024-01-26T23:59:59.999Z+0800");
        body.withFromDate("2024-01-20T00:00:00.000Z+0800");
        body.withSortBy("create_time");
        body.withLimit(10);
        body.withOffset(0);
        body.withCondition(conditionbody);
        body.withDataclassId("28f61af50fc9452aa0ed5ea25c3cc3d3");
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            ListIndicatorsResponse response = client.listIndicators(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

查询id为id1、id2，名称为威胁情报名称，类型为DATA_SOURCE，数据类id为28f61af50fc9452aa0ed5ea25c3cc3d3的威胁情报列表，偏移量为0，查询上限10条，根据create_time排序

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListIndicatorsRequest()
        request.workspace_id = "{workspace_id}"
```

```
        listLogicsCondition = [
            "title"
        ]
        listDataConditions = [
            "name",
            "=",
            "威胁情报名称"
        ]
        listConditionsCondition = [
            IndicatorListSearchRequestConditionConditions(
                name="name",
                data=listDataConditions
            )
        ]
        conditionbody = IndicatorListSearchRequestCondition(
            conditions=listConditionsCondition,
            logics=listLogicsCondition
        )
        listIdsbody = [
            "id1",
            "id2"
        ]
        request.body = IndicatorListSearchRequest(
            to_date="2024-01-26T23:59:59.999Z+0800",
            from_date="2024-01-20T00:00:00.000Z+0800",
            sort_by="create_time",
            limit=10,
            offset=0,
            condition=conditionbody,
            dataclass_id="28f61af50fc9452aa0ed5ea25c3cc3d3",
            ids=listIdsbody
        )
        response = client.list_indicators(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

查询id为id1、id2，名称为威胁情报名称，类型为DATA_SOURCE，数据类id为28f61af50fc9452aa0ed5ea25c3cc3d3的威胁情报列表，偏移量为0，查询上限10条，根据create_time排序

```
package main

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

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

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
```

```
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListIndicatorsRequest{}
    request.WorkspaceId = "{workspace_id}"
    var listLogicsCondition = []string{
        "title",
    }
    var listDataConditions = []string{
        "name",
        "=",
        "威胁情报名称",
    }
    nameConditions:= "name"
    var listConditionsCondition = []model.IndicatorListSearchRequestConditionConditions{
        {
            Name: &nameConditions,
            Data: &listDataConditions,
        },
    }
    conditionbody := &model.IndicatorListSearchRequestCondition{
        Conditions: &listConditionsCondition,
        Logics: &listLogicsCondition,
    }
    var listIdsbody = []string{
        "id1",
        "id2",
    }
    toDateIndicatorListSearchRequest:= "2024-01-26T23:59:59.999Z+0800"
    fromDateIndicatorListSearchRequest:= "2024-01-20T00:00:00.000Z+0800"
    sortByIndicatorListSearchRequest:= "create_time"
    dataclassIdIndicatorListSearchRequest:= "28f61af50fc9452aa0ed5ea25c3cc3d3"
    request.Body = &model.IndicatorListSearchRequest{
        ToDate: &toDateIndicatorListSearchRequest,
        FromDate: &fromDateIndicatorListSearchRequest,
        SortBy: &sortByIndicatorListSearchRequest,
        Limit: int32(10),
        Offset: int32(0),
        Condition: conditionbody,
        DataclassId: &dataclassIdIndicatorListSearchRequest,
        Ids: &listIdsbody,
    }
    response, err := client.ListIndicators(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.3.2 创建威胁情报

功能介绍

创建威胁情报

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/indicators

表 4-270 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

请求参数

表 4-271 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	租户的Token
content-type	是	String	application/json;charset=UTF-8

表 4-272 请求 Body 参数

参数	是否必选	参数类型	描述
data_object	是	**CreateIndicatorDetail** object	情报详情信息

表 4-273 CreateIndicatorDetail

参数	是否必选	参数类型	描述
data_source	是	**data_source** object	数据源信息
verdict	是	String	威胁度
confidence	否	Integer	置信度
status	否	String	状态
labels	否	String	标签
value	是	String	值
granular_marking	是	String	粒度（保密等级），由高到低：1（首次发现）、2（自产数据）、3（需购买）、4（外网直接查询）
environment	是	**environment** object	环境信息
defanged	是	Boolean	是否失效
first_report_time	是	String	首次发生时间
last_report_time	否	String	最近发生时间
id	否	String	威胁情报ID
indicator_type	是	**indicator_type** object	威胁情报类型
name	是	String	威胁情报名称
dataclass_id	否	String	数据类ID
workspace_id	是	String	workspace id
project_id	否	String	Project id value
dataclass	否	**DataClassRefPojo** object	数据类对象信息

参数	是否必选	参数类型	描述
create_time	否	String	Create time
update_time	否	String	Update time

表 4-274 data_source

参数	是否必选	参数类型	描述
source_type	是	Integer	current page count
domain_id	是	String	Id value
project_id	是	String	Id value
region_id	是	String	Id value
product_name	是	String	Id value
product_feature	是	String	Id value

表 4-275 environment

参数	是否必选	参数类型	描述
vendor_type	是	String	环境供应商
domain_id	是	String	租户ID
region_id	是	String	区域ID
project_id	是	String	项目ID

表 4-276 indicator_type

参数	是否必选	参数类型	描述
indicator_type	是	String	威胁情报类型
id	是	String	情报类型ID

表 4-277 DataClassRefPojo

参数	是否必选	参数类型	描述
id	是	String	数据类ID

参数	是否必选	参数类型	描述
name	否	String	数据类名称

响应参数

状态码： 200

表 4-278 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-279 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**IndicatorDetail** object	情报详情信息

表 4-280 IndicatorDetail

参数	参数类型	描述
id	String	威胁情报ID
name	String	威胁情报名称
data_object	**IndicatorDataObjectDetail** object	情报详情
workspace_id	String	工作空间ID
project_id	String	项目ID
dataclass_ref	**DataClassRefPojo** object	数据类对象信息
create_time	String	创建时间
update_time	String	更新时间

表 4-281 IndicatorDataObjectDetail

参数	参数类型	描述
indicator_type	**indicator_type** object	情报类型对象
value	String	值，如：ip url domain等
update_time	String	更新时间
create_time	String	创建时间
environment	**environment** object	环境信息
data_source	**data_source** object	数据源信息
first_report_time	String	首次发生时间
is_deleted	Boolean	是否删除
last_report_time	String	最近发生时间
granular_marking	Integer	粒度（保密等级），由高到低：1（首次发现）、2（自产数据）、3（需购买）、4（外网直接查询）
name	String	名称
id	String	威胁情报ID
project_id	String	项目ID
revoked	Boolean	是否作废
status	String	状态， Open--打开，Closed--关闭, Revoked--作废
verdict	String	威胁度， Black--黑,White--白，Gray--灰
workspace_id	String	工作空间ID
confidence	Integer	置信度，取值范围是80-100

表 4-282 indicator_type

参数	参数类型	描述
indicator_type	String	情报类型
id	String	情报类型ID

表 4-283 environment

参数	参数类型	描述
vendor_type	String	环境供应商
domain_id	String	租户ID
region_id	String	区域ID
project_id	String	项目ID

表 4-284 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下：1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	租户ID
project_id	String	项目ID
region_id	String	区域ID

表 4-285 DataClassRefPojo

参数	参数类型	描述
id	String	数据类ID
name	String	数据类名称

状态码： 400

表 4-286 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-287 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

创建一条威胁情报，威胁情报名称为“威胁情报名称”，威胁情报版本为1，威胁情报类型为DATA_SOURCE，触发标志为否。

```
{
  "data_object" : {
    "data_source" : {
      "source_type" : 3,
      "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "product_name" : "test",
      "product_feature" : "test"
    },
    "verdict" : "BLACK",
    "confidence" : 4,
    "status" : "OPEN",
    "labels" : "OPEN",
    "value" : "123",
    "granular_marking" : "1",
    "environment" : {
      "vendor_type" : "MyXXX",
      "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    },
    "defanged" : false,
    "first_report_time" : "2021-01-30T23:00:00Z+0800",
    "last_report_time" : "2021-01-30T23:00:00Z+0800",
    "indicator_type" : {
      "id" : "909494e3-558e-xxxxxx-07a8e18ca6xxx",
      "indicator_type" : "ipv6"
    },
    "name" : "威胁情报名称",
    "dataclass_id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "dataclass" : {
      "id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
      "name" : "名称"
    },
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800"
  }
}
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
    "name" : "威胁情报名称",
    "data_object" : {
      "indicator_type" : {
        "indicator_type" : "ipv6",
        "id" : "ac794b2dfab9fe8c0676587301a636d3"
      },
      "value" : "ip",
      "data_source" : {
        "domain_id" : "ac7438b990ef4a37b741004eb45e8bf4",
        "project_id" : "5b8bb3c888db498f9eeaf1023f7ba597",
```

```
      "region_id" : "cn-xxx-7",
      "source_type" : 1
    },
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "granular_marking" : 1,
    "first_report_time" : "2023-07-04T16:47:01Z+0800",
    "status" : "Open"
  },
  "dataclass_ref" : {
    "id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
    "name" : "名称"
  },
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "update_time" : "2021-01-30T23:00:00Z+0800"
 }
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一条威胁情报，威胁情报名称为“威胁情报名称”，威胁情报版本为1，威胁情报类型为DATA_SOURCE，触发标志为否。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class CreateIndicatorSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateIndicatorRequest request = new CreateIndicatorRequest();
        request.withWorkspaceId("{workspace_id}");
        IndicatorCreateRequest body = new IndicatorCreateRequest();
        DataClassRefPojo dataclassDataObject = new DataClassRefPojo();
        dataclassDataObject.withId("28f61af50fc9452aa0ed5ea25c3cc3d3")
            .withName("名称");
        CreateIndicatorDetailIndicatorType indicatorTypeDataObject = new
```

```
CreateIndicatorDetailIndicatorType();
        indicatorTypeDataObject.withIndicatorType("ipv6")
            .withId("909494e3-558e-xxxxxx-07a8e18ca6xxx");
        CreateIndicatorDetailEnvironment environmentDataObject = new CreateIndicatorDetailEnvironment();
        environmentDataObject.withVendorType("MyXXX")
            .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        CreateIndicatorDetailDataSource dataSourceDataObject = new CreateIndicatorDetailDataSource();
        dataSourceDataObject.withSourceType(3)
            .withDomainId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withRegionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withProductName("test")
            .withProductFeature("test");
        CreateIndicatorDetail dataObjectbody = new CreateIndicatorDetail();
        dataObjectbody.withDataSource(dataSourceDataObject)
            .withVerdict("BLACK")
            .withConfidence(4)
            .withStatus("OPEN")
            .withLabels("OPEN")
            .withValue("123")
            .withGranularMarking("1")
            .withEnvironment(environmentDataObject)
            .withDefanged(false)
            .withFirstReportTime("2021-01-30T23:00:00Z+0800")
            .withLastReportTime("2021-01-30T23:00:00Z+0800")
            .withIndicatorType(indicatorTypeDataObject)
            .withName("威胁情报名称")
            .withDataclassId("28f61af50fc9452aa0ed5ea25c3cc3d3")
            .withWorkspaceId("909494e3-558e-46b6-a9eb-07a8e18ca620")
            .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withDataclass(dataclassDataObject)
            .withCreateTime("2021-01-30T23:00:00Z+0800")
            .withUpdateTime("2021-01-30T23:00:00Z+0800");
        body.withDataObject(dataObjectbody);
        request.withBody(body);
        try {
            CreateIndicatorResponse response = client.createIndicator(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

创建一条威胁情报，威胁情报名称为“威胁情报名称”，威胁情报版本为1，威胁情报类型为DATA_SOURCE，触发标志为否。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateIndicatorRequest()
        request.workspace_id = "{workspace_id}"
        dataclassDataObject = DataClassRefPojo(
            id="28f61af50fc9452aa0ed5ea25c3cc3d3",
            name="名称"
        )
        indicatorTypeDataObject = CreateIndicatorDetailIndicatorType(
            indicator_type="ipv6",
            id="909494e3-558e-xxxxxx-07a8e18ca6xxx"
        )
        environmentDataObject = CreateIndicatorDetailEnvironment(
            vendor_type="MyXXX",
            domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f"
        )
        dataSourceDataObject = CreateIndicatorDetailDataSource(
            source_type=3,
            domain_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            region_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            product_name="test",
            product_feature="test"
        )
        dataObjectbody = CreateIndicatorDetail(
            data_source=dataSourceDataObject,
            verdict="BLACK",
            confidence=4,
            status="OPEN",
            labels="OPEN",
            value="123",
            granular_marking="1",
            environment=environmentDataObject,
            defanged=False,
            first_report_time="2021-01-30T23:00:00Z+0800",
            last_report_time="2021-01-30T23:00:00Z+0800",
            indicator_type=indicatorTypeDataObject,
            name="威胁情报名称",
            dataclass_id="28f61af50fc9452aa0ed5ea25c3cc3d3",
            workspace_id="909494e3-558e-46b6-a9eb-07a8e18ca620",
            project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            dataclass=dataclassDataObject,
            create_time="2021-01-30T23:00:00Z+0800",
            update_time="2021-01-30T23:00:00Z+0800"
        )
        request.body = IndicatorCreateRequest(
            data_object=dataObjectbody
        )
        response = client.create_indicator(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

创建一条威胁情报，威胁情报名称为“威胁情报名称”，威胁情报版本为1，威胁情报类型为DATA_SOURCE，触发标志为否。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateIndicatorRequest{}
    request.WorkspaceId = "{workspace_id}"
    nameDataclass:= "名称"
    dataclassDataObject := &model.DataClassRefPojo{
        Id: "28f61af50fc9452aa0ed5ea25c3cc3d3",
        Name: &nameDataclass,
    }
    indicatorTypeDataObject := &model.CreateIndicatorDetailIndicatorType{
        IndicatorType: "ipv6",
        Id: "909494e3-558e-xxxxxx-07a8e18ca6xxx",
    }
    environmentDataObject := &model.CreateIndicatorDetailEnvironment{
        VendorType: "MyXXX",
        DomainId: "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        RegionId: "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        ProjectId: "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    }
    dataSourceDataObject := &model.CreateIndicatorDetailDataSource{
        SourceType: int32(3),
        DomainId: "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        ProjectId: "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        RegionId: "909494e3-558e-46b6-a9eb-07a8e18ca62f",
        ProductName: "test",
        ProductFeature: "test",
    }
    confidenceDataObject:= int32(4)
    statusDataObject:= "OPEN"
    labelsDataObject:= "OPEN"
    lastReportTimeDataObject:= "2021-01-30T23:00:00Z+0800"
```

```
    dataclassIdDataObject:= "28f61af50fc9452aa0ed5ea25c3cc3d3"
    projectIdDataObject:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    createTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    updateTimeDataObject:= "2021-01-30T23:00:00Z+0800"
    dataObjectbody := &model.CreateIndicatorDetail{
        DataSource: dataSourceDataObject,
        Verdict: "BLACK",
        Confidence: &confidenceDataObject,
        Status: &statusDataObject,
        Labels: &labelsDataObject,
        Value: "123",
        GranularMarking: "1",
        Environment: environmentDataObject,
        Defanged: false,
        FirstReportTime: "2021-01-30T23:00:00Z+0800",
        LastReportTime: &lastReportTimeDataObject,
        IndicatorType: indicatorTypeDataObject,
        Name: "威胁情报名称",
        DataclassId: &dataclassIdDataObject,
        WorkspaceId: "909494e3-558e-46b6-a9eb-07a8e18ca620",
        ProjectId: &projectIdDataObject,
        Dataclass: dataclassDataObject,
        CreateTime: &createTimeDataObject,
        UpdateTime: &updateTimeDataObject,
    }
    request.Body = &model.IndicatorCreateRequest{
        DataObject: dataObjectbody,
    }
    response, err := client.CreateIndicator(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.3.3 删除威胁情报

功能介绍

删除威胁情报

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/workspaces/{workspace_id}/soc/indicators

表 4-288 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

请求参数

表 4-289 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	租户的Token
content-type	是	String	application/json;charset=UTF-8

表 4-290 请求 Body 参数

参数	是否必选	参数类型	描述
batch_ids	否	Array of strings	威胁情报ID列表

响应参数

状态码： 200

表 4-291 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-292 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**IndicatorBatchOperateResponse** object	情报响应参数

表 4-293 IndicatorBatchOperateResponse

参数	参数类型	描述
success_ids	Array of strings	成功ID列表
error_ids	Array of strings	失败ID列表

状态码： 400

表 4-294 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-295 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

删除一条威胁情报，威胁情报批次ID为909494e3-558e-46b6-a9eb-07a8e18ca62f。

```
{
  "batch_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ]
}
```

响应示例

状态码： 200

请求成功响应

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "data" : {
    "success_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ],
    "error_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ]
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

删除一条威胁情报，威胁情报批次ID为909494e3-558e-46b6-a9eb-07a8e18ca62f。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class DeleteIndicatorSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteIndicatorRequest request = new DeleteIndicatorRequest();
        request.withWorkspaceId("{workspace_id}");
        DeleteIndicatorRequestBody body = new DeleteIndicatorRequestBody();
        List<String> listbodyBatchIds = new ArrayList<>();
        listbodyBatchIds.add("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        body.withBatchIds(listbodyBatchIds);
        request.withBody(body);
        try {
            DeleteIndicatorResponse response = client.deleteIndicator(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

删除一条威胁情报，威胁情报批次ID为909494e3-558e-46b6-a9eb-07a8e18ca62f。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteIndicatorRequest()
        request.workspace_id = "{workspace_id}"
        listBatchIdsbody = [
            "909494e3-558e-46b6-a9eb-07a8e18ca62f"
        ]
        request.body = DeleteIndicatorRequestBody(
            batch_ids=listBatchIdsbody
        )
        response = client.delete_indicator(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

删除一条威胁情报，威胁情报批次ID为909494e3-558e-46b6-a9eb-07a8e18ca62f。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteIndicatorRequest{}
    request.WorkspaceId = "{workspace_id}"
    var listBatchIdsbody = []string{
        "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    }
    request.Body = &model.DeleteIndicatorRequestBody{
        BatchIds: &listBatchIdsbody,
    }
    response, err := client.DeleteIndicator(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	请求成功响应
400	请求失败响应

错误码

请参见**错误码**。

4.3.4 查询威胁情报详情

功能介绍

查询威胁情报详情

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/indicators/{indicator_id}

表 4-296 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
indicator_id	是	String	威胁情报ID

请求参数

表 4-297 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	租户的Token
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-298 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-299 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息

参数	参数类型	描述
data	**IndicatorDetail** object	情报详情信息

表 4-300 IndicatorDetail

参数	参数类型	描述
id	String	威胁情报ID
name	String	威胁情报名称
data_object	**IndicatorDataObjectDetail** object	情报详情
workspace_id	String	工作空间ID
project_id	String	项目ID
dataclass_ref	**DataClassRefPojo** object	数据类对象信息
create_time	String	创建时间
update_time	String	更新时间

表 4-301 IndicatorDataObjectDetail

参数	参数类型	描述
indicator_type	**indicator_type** object	情报类型对象
value	String	值，如：ip url domain等
update_time	String	更新时间
create_time	String	创建时间
environment	**environment** object	环境信息
data_source	**data_source** object	数据源信息
first_report_time	String	首次发生时间
is_deleted	Boolean	是否删除
last_report_time	String	最近发生时间

参数	参数类型	描述
granular_marking	Integer	粒度（保密等级），由高到低：1（首次发现）、2（自产数据）、3（需购买）、4（外网直接查询）
name	String	名称
id	String	威胁情报ID
project_id	String	项目ID
revoked	Boolean	是否作废
status	String	状态， Open--打开，Closed--关闭, Revoked--作废
verdict	String	威胁度， Black--黑,White--白，Gray--灰
workspace_id	String	工作空间ID
confidence	Integer	置信度，取值范围是80-100

表 4-302 indicator_type

参数	参数类型	描述
indicator_type	String	情报类型
id	String	情报类型ID

表 4-303 environment

参数	参数类型	描述
vendor_type	String	环境供应商
domain_id	String	租户ID
region_id	String	区域ID
project_id	String	项目ID

表 4-304 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下：1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	租户ID

参数	参数类型	描述
project_id	String	项目ID
region_id	String	区域ID

表 4-305 DataClassRefPojo

参数	参数类型	描述
id	String	数据类ID
name	String	数据类名称

状态码： 400

表 4-306 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-307 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
    "name" : "指标名称",
    "data_object" : {
      "indicator_type" : {
        "indicator_type" : "ipv6",
```

```
      "id" : "ac794b2dfab9fe8c0676587301a636d3"
    },
    "value" : "ip",
    "data_source" : {
      "domain_id" : "ac7438b990ef4a37b741004eb45e8bf4",
      "project_id" : "5b8bb3c888db498f9eeaf1023f7ba597",
      "region_id" : "cn-xxx-7",
      "source_type" : 1
    },
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "granular_marking" : 1,
    "first_report_time" : "2023-07-04T16:47:01Z+0800",
    "status" : "Open"
  },
  "dataclass_ref" : {
    "id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
    "name" : "名称"
  },
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "update_time" : "2021-01-30T23:00:00Z+0800"
 }
}
```


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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowIndicatorDetailSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowIndicatorDetailRequest request = new ShowIndicatorDetailRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withIndicatorId("{indicator_id}");
        try {
            ShowIndicatorDetailResponse response = client.showIndicatorDetail(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowIndicatorDetailRequest()
        request.workspace_id = "{workspace_id}"
        request.indicator_id = "{indicator_id}"
        response = client.show_indicator_detail(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowIndicatorDetailRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.IndicatorId = "{indicator_id}"
    response, err := client.ShowIndicatorDetail(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.3.5 更新威胁情报

功能介绍

更新威胁情报

调用方法

请参见**如何调用API**。

URI

PUT /v1/{project_id}/workspaces/{workspace_id}/soc/indicators/{indicator_id}

表 4-308 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
indicator_id	是	String	威胁情报ID

请求参数

表 4-309 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	租户的Token
content-type	是	String	application/json;charset=UTF-8

表 4-310 请求 Body 参数

参数	是否必选	参数类型	描述
data_object	否	IndicatorDataObjectDetail object	情报详情

表 4-311 IndicatorDataObjectDetail

参数	是否必选	参数类型	描述
indicator_type	否	indicator_type object	情报类型对象
value	否	String	值，如：ip url domain等
update_time	否	String	更新时间
create_time	否	String	创建时间
environment	否	environment object	环境信息

参数	是否必选	参数类型	描述
data_source	否	data_source object	数据源信息
first_report_time	否	String	首次发生时间
is_deleted	否	Boolean	是否删除
last_report_time	否	String	最近发生时间
granular_marking	否	Integer	粒度（保密等级），由高到低：1（首次发现）、2（自产数据）、3（需购买）、4（外网直接查询）
name	否	String	名称
id	否	String	威胁情报ID
project_id	否	String	项目ID
revoked	否	Boolean	是否作废
status	否	String	状态， Open--打开，Closed--关闭, Revoked--作废
verdict	否	String	威胁度， Black--黑,White--白，Gray--灰
workspace_id	否	String	工作空间ID
confidence	否	Integer	置信度，取值范围是80-100

表 4-312 indicator_type

参数	是否必选	参数类型	描述
indicator_type	否	String	情报类型
id	否	String	情报类型ID

表 4-313 environment

参数	是否必选	参数类型	描述
vendor_type	否	String	环境供应商
domain_id	否	String	租户ID
region_id	否	String	区域ID

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

表 4-314 data_source

参数	是否必选	参数类型	描述
source_type	否	Integer	数据源类型，取值范围如下：1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	否	String	租户ID
project_id	否	String	项目ID
region_id	否	String	区域ID

响应参数

状态码： 200

表 4-315 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-316 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**IndicatorDetail** object	情报详情信息

表 4-317 IndicatorDetail

参数	参数类型	描述
id	String	威胁情报ID
name	String	威胁情报名称

参数	参数类型	描述
data_object	**IndicatorDataObjectDetail** object	情报详情
workspace_id	String	工作空间ID
project_id	String	项目ID
dataclass_ref	**DataClassRefPojo** object	数据类对象信息
create_time	String	创建时间
update_time	String	更新时间

表 4-318 IndicatorDataObjectDetail

参数	参数类型	描述
indicator_type	**indicator_type** object	情报类型对象
value	String	值，如：ip url domain等
update_time	String	更新时间
create_time	String	创建时间
environment	**environment** object	环境信息
data_source	**data_source** object	数据源信息
first_report_time	String	首次发生时间
is_deleted	Boolean	是否删除
last_report_time	String	最近发生时间
granular_marking	Integer	粒度（保密等级），由高到低：1（首次发现）、2（自产数据）、3（需购买）、4（外网直接查询）
name	String	名称
id	String	威胁情报ID
project_id	String	项目ID
revoked	Boolean	是否作废
status	String	状态， Open--打开，Closed--关闭, Revoked--作废

参数	参数类型	描述
verdict	String	威胁度， Black--黑,White--白，Gray--灰
workspace_id	String	工作空间ID
confidence	Integer	置信度，取值范围是80-100

表 4-319 indicator_type

参数	参数类型	描述
indicator_type	String	情报类型
id	String	情报类型ID

表 4-320 environment

参数	参数类型	描述
vendor_type	String	环境供应商
domain_id	String	租户ID
region_id	String	区域ID
project_id	String	项目ID

表 4-321 data_source

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下：1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	租户ID
project_id	String	项目ID
region_id	String	区域ID

表 4-322 DataClassRefPojo

参数	参数类型	描述
id	String	数据类ID
name	String	数据类名称

状态码： 400

表 4-323 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-324 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

更新一条威胁情报，威胁情报触发标志为否，值为ip。

```
{
  "data_object" : {
    "indicator_type" : {
      "indicator_type" : "ipv6",
      "id" : "ac794b2dfab9fe8c0676587301a636d3"
    },
    "value" : "ip",
    "data_source" : {
      "domain_id" : "ac7438b990ef4a37b741004eb45e8bf4",
      "project_id" : "5b8bb3c888db498f9eeaf1023f7ba597",
      "region_id" : "cn-xxx-7",
      "source_type" : 1
    },
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "granular_marking" : 1,
    "first_report_time" : "2023-07-04T16:47:01Z+0800",
    "status" : "Open"
  }
}
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "data" : {
    "id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
    "name" : "威胁情报名称",
    "data_object" : {
      "indicator_type" : {
        "indicator_type" : "ipv6",
        "id" : "ac794b2dfab9fe8c0676587301a636d3"
      },
```

```
    "value" : "ip",
    "data_source" : {
      "domain_id" : "ac7438b990ef4a37b741004eb45e8bf4",
      "project_id" : "5b8bb3c888db498f9eeaf1023f7ba597",
      "region_id" : "cn-xxx-7",
      "source_type" : 1
    },
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "granular_marking" : 1,
    "first_report_time" : "2023-07-04T16:47:01Z+0800",
    "status" : "Open"
  },
  "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca620",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "dataclass_ref" : {
    "id" : "28f61af50fc9452aa0ed5ea25c3cc3d3",
    "name" : "名称"
  },
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "update_time" : "2021-01-30T23:00:00Z+0800"
 }
}
```


SDK 代码示例

SDK代码示例如下。

Java

更新一条威胁情报，威胁情报触发标志为否，值为ip。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class UpdateIndicatorSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdateIndicatorRequest request = new UpdateIndicatorRequest();
        request.withWorkspaceId("{workspace_id}");
```

```
        request.withIndicatorId("{indicator_id}");
        UpdateIndicatorRequestBody body = new UpdateIndicatorRequestBody();
        IndicatorDataObjectDetailDataSource dataSourceDataObject = new
IndicatorDataObjectDetailDataSource();
        dataSourceDataObject.withSourceType(1)
            .withDomainId("ac7438b990ef4a37b741004eb45e8bf4")
            .withProjectId("5b8bb3c888db498f9eeaf1023f7ba597")
            .withRegionId("cn-xxx-7");
        IndicatorDataObjectDetailIndicatorType indicatorTypeDataObject = new
IndicatorDataObjectDetailIndicatorType();
        indicatorTypeDataObject.withIndicatorType("ipv6")
            .withId("ac794b2dfab9fe8c0676587301a636d3");
        IndicatorDataObjectDetail dataObjectbody = new IndicatorDataObjectDetail();
        dataObjectbody.withIndicatorType(indicatorTypeDataObject)
            .withValue("ip")
            .withDataSource(dataSourceDataObject)
            .withFirstReportTime("2023-07-04T16:47:01Z+0800")
            .withGranularMarking(1)
            .withProjectId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
            .withStatus("Open")
            .withWorkspaceId("909494e3-558e-46b6-a9eb-07a8e18ca620");
        body.withDataObject(dataObjectbody);
        request.withBody(body);
        try {
            UpdateIndicatorResponse response = client.updateIndicator(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

更新一条威胁情报，威胁情报触发标志为否，值为ip。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

```
    try:
        request = UpdateIndicatorRequest()
        request.workspace_id = "{workspace_id}"
        request.indicator_id = "{indicator_id}"
        dataSourceDataObject = IndicatorDataObjectDetailDataSource(
            source_type=1,
            domain_id="ac7438b990ef4a37b741004eb45e8bf4",
            project_id="5b8bb3c888db498f9eeaf1023f7ba597",
            region_id="cn-xxx-7"
        )
        indicatorTypeDataObject = IndicatorDataObjectDetailIndicatorType(
            indicator_type="ipv6",
            id="ac794b2dfab9fe8c0676587301a636d3"
        )
        dataObjectbody = IndicatorDataObjectDetail(
            indicator_type=indicatorTypeDataObject,
            value="ip",
            data_source=dataSourceDataObject,
            first_report_time="2023-07-04T16:47:01Z+0800",
            granular_marking=1,
            project_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            status="Open",
            workspace_id="909494e3-558e-46b6-a9eb-07a8e18ca620"
        )
        request.body = UpdateIndicatorRequestBody(
            data_object=dataObjectbody
        )
        response = client.update_indicator(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

更新一条威胁情报，威胁情报触发标志为否，值为ip。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
```

```
        Build())

    request := &model.UpdateIndicatorRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.IndicatorId = "{indicator_id}"
    sourceTypeDataSource:= int32(1)
    domainIdDataSource:= "ac7438b990ef4a37b741004eb45e8bf4"
    projectIdDataSource:= "5b8bb3c888db498f9eeaf1023f7ba597"
    regionIdDataSource:= "cn-xxx-7"
    dataSourceDataObject := &model.IndicatorDataObjectDetailDataSource{
        SourceType: &sourceTypeDataSource,
        DomainId: &domainIdDataSource,
        ProjectId: &projectIdDataSource,
        RegionId: &regionIdDataSource,
    }
    indicatorTypeIndicatorType:= "ipv6"
    idIndicatorType:= "ac794b2dfab9fe8c0676587301a636d3"
    indicatorTypeDataObject := &model.IndicatorDataObjectDetailIndicatorType{
        IndicatorType: &indicatorTypeIndicatorType,
        Id: &idIndicatorType,
    }
    valueDataObject:= "ip"
    firstReportTimeDataObject:= "2023-07-04T16:47:01Z+0800"
    granularMarkingDataObject:= int32(1)
    projectIdDataObject:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    statusDataObject:= "Open"
    workspaceIdDataObject:= "909494e3-558e-46b6-a9eb-07a8e18ca620"
    dataObjectbody := &model.IndicatorDataObjectDetail{
        IndicatorType: indicatorTypeDataObject,
        Value: &valueDataObject,
        DataSource: dataSourceDataObject,
        FirstReportTime: &firstReportTimeDataObject,
        GranularMarking: &granularMarkingDataObject,
        ProjectId: &projectIdDataObject,
        Status: &statusDataObject,
        WorkspaceId: &workspaceIdDataObject,
    }
    request.Body = &model.UpdateIndicatorRequestBody{
        DataObject: dataObjectbody,
    }
    response, err := client.UpdateIndicator(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	请求成功响应信息
400	请求错误响应信息

错误码

请参见**错误码**。

4.4 剧本管理

4.4.1 剧本运行监控

功能介绍

剧本运行监控

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/{playbook_id}/monitor

表 4-325 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
playbook_id	是	String	剧本ID

表 4-326 Query 参数

参数	是否必选	参数类型	描述
start_time	是	String	开始时间。格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。例如：2021-01-30T23:00:00Z+0800。时区信息为剧本实例产生的时区，无法解析时区的时间，默认时区填东八区。
version_query_type	是	String	统计剧本版本类型（ALL:全部，VALID:有效的，DELETED:已删除）

参数	是否必选	参数类型	描述
end_time	是	String	结束时间。格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。例如：2021-01-30T23:00:00Z+0800。时区信息为剧本实例产生的时区，无法解析时区的时间，默认时区填东八区。

请求参数

表 4-327 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-328 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-329 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**PlaybookInstanceMonitorDetail** object	剧本运行监控详情

表 4-330 PlaybookInstanceMonitorDetail

参数	参数类型	描述
total_instance_run_num	Integer	运行总次数
schedule_instance_run_num	Integer	定时触发执行次数
event_instance_run_num	Integer	时间触发执行次数
average_run_time	Number	平均运行时间
min_run_time_instance	**PlaybookInstanceRunStatistics** object	最短运行时间流程实例信息
max_run_time_instance	**PlaybookInstanceRunStatistics** object	最长运行时间流程实例信息
total_instance_num	Integer	剧本实例总数
success_instance_num	Integer	运行成功实例数量
fail_instance_num	Integer	运行失败实例数量
terminate_instance_num	Integer	运行终止实例数量
running_instance_num	Integer	运行中实例数量

表 4-331 PlaybookInstanceRunStatistics

参数	参数类型	描述
playbook_instance_id	String	剧本实例ID
playbook_instance_name	String	剧本实例名称
playbook_instance_run_time	Number	剧本实例运行时间

状态码： 400

表 4-332 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-333 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : "00000000",
  "message" : "",
  "data" : {
    "total_instance_run_num" : "Unknown Type: in",
    "schedule_instance_run_num" : 99999999,
    "event_instance_run_num" : 99999999,
    "average_run_time" : 9999999999,
    "min_run_time_instance" : {
      "playbook_instance_id" : "string",
      "playbook_instance_name" : "string",
      "playbook_instance_run_time" : 9999999999
    },
    "max_run_time_instance" : {
      "playbook_instance_id" : "string",
      "playbook_instance_name" : "string",
      "playbook_instance_run_time" : 9999999999
    },
    "total_instance_num" : 99999999,
    "success_instance_num" : 99999999,
    "fail_instance_num" : 99999999,
    "terminate_instance_num" : 99999999,
    "running_instance_num" : 99999999
  }
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowPlaybookMonitorsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowPlaybookMonitorsRequest request = new ShowPlaybookMonitorsRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withPlaybookId("{playbook_id}");
        try {
            ShowPlaybookMonitorsResponse response = client.showPlaybookMonitors(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

```
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowPlaybookMonitorsRequest()
        request.workspace_id = "{workspace_id}"
        request.playbook_id = "{playbook_id}"
        response = client.show_playbook_monitors(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPlaybookMonitorsRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.PlaybookId = "{playbook_id}"
    response, err := client.ShowPlaybookMonitors(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.4.2 剧本数据统计

功能介绍

剧本统计数据

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/statistics

表 4-334 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

请求参数

表 4-335 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）。

参数	是否必选	参数类型	描述
content-type	是	String	application/ json;charset=UTF-8

响应参数

状态码： 200

表 4-336 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-337 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**PlaybookStatisticDetail** object	剧本状态统计信息

表 4-338 PlaybookStatisticDetail

参数	参数类型	描述
unapproved_num	Integer	未审核剧本数量
disabled_num	Integer	未启用剧本数量
enabled_num	Integer	已启用剧本数量

状态码： 400

表 4-339 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-340 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "unapproved_num" : 99999999,
    "disabled_num" : 99999999,
    "enabled_num" : 99999999
  }
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowPlaybookStatisticsSolution {

    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);
```

```
        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowPlaybookStatisticsRequest request = new ShowPlaybookStatisticsRequest();
        request.withWorkspaceId("{workspace_id}");
        try {
            ShowPlaybookStatisticsResponse response = client.showPlaybookStatistics(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowPlaybookStatisticsRequest()
        request.workspace_id = "{workspace_id}"
        response = client.show_playbook_statistics(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"
```

```
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPlaybookStatisticsRequest{}
    request.WorkspaceId = "{workspace_id}"
    response, err := client.ShowPlaybookStatistics(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.4.3 查询剧本列表

功能介绍

查询剧本列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks

表 4-341 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

表 4-342 Query 参数

参数	是否必选	参数类型	描述
search_txt	否	String	搜索关键字
enabled	否	Boolean	是否启用
offset	是	Integer	分页查询参数。用于指定查询结果的起始位置，从0开始
limit	是	Integer	分页查询参数，用于指定一次查询最多的结果数，从1开始
description	否	String	剧本描述
dataclass_name	否	String	数据类名称
name	否	String	剧本名称

请求参数

表 4-343 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-344 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-345 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	响应消息信息
total	Integer	总条数
size	Integer	分页查询数据大小
page	Integer	当前页码
data	Array of **PlaybookInfo** objects	剧本列表信息

表 4-346 PlaybookInfo

参数	参数类型	描述
id	String	剧本ID
name	String	剧本名称
description	String	描述信息
create_time	String	剧本创建时间
update_time	String	剧本更新时间
project_id	String	项目ID
version_id	String	剧本版本ID
enabled	Boolean	是否启用
workspace_id	String	工作空间ID
approve_role	String	审核用户角色
user_role	String	用户角色

参数	参数类型	描述
edit_role	String	编辑用户角色
owner_id	String	所有者ID
version	String	版本号
dataclass_name	String	数据类名称
dataclass_id	String	数据类ID
unaudited_version_id	String	待审核剧本版本ID
reject_version_id	String	已驳回剧本版本ID

状态码： 400

表 4-347 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-348 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

剧本列表查询成功响应参数

```
{
  "code" : 0,
  "message" : null,
  "total" : 41,
  "page" : 10,
  "data" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "description" : "This my XXXX",
```

```
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "enabled" : true,
    "workspace_id" : "string",
    "approve_role" : "approve",
    "user_role" : "string",
    "edit_role" : "editor",
    "owner_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1",
    "dataclass_name" : "string",
    "dataclass_id" : "string",
    "unaudited_version_id" : "string",
    "reject_version_id" : "string"
  } ]
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListPlaybooksSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListPlaybooksRequest request = new ListPlaybooksRequest();
        request.withWorkspaceId("{workspace_id}");
        try {
            ListPlaybooksResponse response = client.listPlaybooks(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListPlaybooksRequest()
        request.workspace_id = "{workspace_id}"
        response = client.list_playbooks(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

    auth := basic.NewCredentialsBuilder().
```

```
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPlaybooksRequest{}
    request.WorkspaceId = "{workspace_id}"
    response, err := client.ListPlaybooks(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	剧本列表查询成功响应参数
400	查询失败响应参数

错误码

请参见**错误码**。

4.4.4 创建剧本

功能介绍

创建剧本

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks

表 4-349 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

请求参数

表 4-350 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-351 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	剧本名称
description	否	String	描述
workspace_id	是	String	工作空间ID

响应参数

状态码： 200

表 4-352 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-353 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**PlaybookInfo** object	剧本详情信息

表 4-354 PlaybookInfo

参数	参数类型	描述
id	String	剧本ID
name	String	剧本名称
description	String	描述信息
create_time	String	剧本创建时间
update_time	String	剧本更新时间
project_id	String	项目ID
version_id	String	剧本版本ID
enabled	Boolean	是否启用
workspace_id	String	工作空间ID
approve_role	String	审核用户角色
user_role	String	用户角色
edit_role	String	编辑用户角色
owner_id	String	所有者ID
version	String	版本号
dataclass_name	String	数据类名称
dataclass_id	String	数据类ID
unaudited_version_id	String	待审核剧本版本ID
reject_version_id	String	已驳回剧本版本ID

状态码： 400

表 4-355 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-356 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

创建一个剧本，剧本名称为MyXXX，工作空间ID为string，审批人为approve，启用状态为开启。

```
{
  "name" : "MyXXX",
  "description" : "This my XXXX",
  "workspace_id" : "string"
}
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "description" : "This my XXXX",
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "enabled" : true,
    "workspace_id" : "string",
    "approve_role" : "approve",
    "user_role" : "string",
    "edit_role" : "editor",
    "owner_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1",
    "dataclass_name" : "string",
    "dataclass_id" : "string",
    "unaudited_version_id" : "string",
    "reject_version_id" : "string"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一个剧本，剧本名称为MyXXX，工作空间ID为string，审批人为approve，启用状态为开启。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class CreatePlaybookSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreatePlaybookRequest request = new CreatePlaybookRequest();
        request.withWorkspaceId("{workspace_id}");
        CreatePlaybookInfo body = new CreatePlaybookInfo();
        body.withWorkspaceId("string");
        body.withDescription("This my XXXX");
        body.withName("MyXXX");
        request.withBody(body);
        try {
            CreatePlaybookResponse response = client.createPlaybook(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

创建一个剧本，剧本名称为MyXXX，工作空间ID为string，审批人为approve，启用状态为开启。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreatePlaybookRequest()
        request.workspace_id = "{workspace_id}"
        request.body = CreatePlaybookInfo(
            workspace_id="string",
            description="This my XXXX",
            name="MyXXX"
        )
        response = client.create_playbook(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

创建一个剧本，剧本名称为MyXXX，工作空间ID为string，审批人为approve，启用状态为开启。

```
package main

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

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

```
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePlaybookRequest{}
    request.WorkspaceId = "{workspace_id}"
    descriptionCreatePlaybookInfo:= "This my XXXX"
    request.Body = &model.CreatePlaybookInfo{
        WorkspaceId: "string",
        Description: &descriptionCreatePlaybookInfo,
        Name: "MyXXX",
    }
    response, err := client.CreatePlaybook(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.4.5 查询剧本详情

功能介绍

查询剧本详情

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/{playbook_id}

表 4-357 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
playbook_id	是	String	ID of playbook

请求参数

表 4-358 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-359 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-360 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**PlaybookInfo** object	剧本详情信息

表 4-361 PlaybookInfo

参数	参数类型	描述
id	String	剧本ID
name	String	剧本名称
description	String	描述信息
create_time	String	剧本创建时间
update_time	String	剧本更新时间
project_id	String	项目ID
version_id	String	剧本版本ID
enabled	Boolean	是否启用
workspace_id	String	工作空间ID
approve_role	String	审核用户角色
user_role	String	用户角色
edit_role	String	编辑用户角色
owner_id	String	所有者ID
version	String	版本号
dataclass_name	String	数据类名称
dataclass_id	String	数据类ID
unaudited_version_id	String	待审核剧本版本ID
reject_version_id	String	已驳回剧本版本ID

状态码： 400

表 4-362 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-363 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "description" : "This my XXXX",
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "enabled" : true,
    "workspace_id" : "string",
    "approve_role" : "approve",
    "user_role" : "string",
    "edit_role" : "editor",
    "owner_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1",
    "dataclass_name" : "string",
    "dataclass_id" : "string",
    "unaudited_version_id" : "string",
    "reject_version_id" : "string"
  }
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowPlaybookSolution {
```

```
    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowPlaybookRequest request = new ShowPlaybookRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withPlaybookId("{playbook_id}");
        try {
            ShowPlaybookResponse response = client.showPlaybook(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowPlaybookRequest()
```

```
        request.workspace_id = "{workspace_id}"
        request.playbook_id = "{playbook_id}"
        response = client.show_playbook(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPlaybookRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.PlaybookId = "{playbook_id}"
    response, err := client.ShowPlaybook(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	请求成功响应信息

状态码	描述
400	请求失败响应信息

错误码

请参见**错误码**。

4.4.6 删除剧本

功能介绍

删除剧本

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/{playbook_id}

表 4-364 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
playbook_id	是	String	ID of playbook

请求参数

表 4-365 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-366 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-367 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	PlaybookInfo object	剧本详情信息

表 4-368 PlaybookInfo

参数	参数类型	描述
id	String	剧本ID
name	String	剧本名称
description	String	描述信息
create_time	String	剧本创建时间
update_time	String	剧本更新时间
project_id	String	项目ID
version_id	String	剧本版本ID
enabled	Boolean	是否启用
workspace_id	String	工作空间ID
approve_role	String	审核用户角色
user_role	String	用户角色
edit_role	String	编辑用户角色
owner_id	String	所有者ID
version	String	版本号
dataclass_name	String	数据类名称

参数	参数类型	描述
dataclass_id	String	数据类ID
unaudited_version_id	String	待审核剧本版本ID
reject_version_id	String	已驳回剧本版本ID

状态码： 400

表 4-369 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-370 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "description" : "This my XXXX",
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "enabled" : true,
    "workspace_id" : "string",
    "approve_role" : "approve",
    "user_role" : "string",
    "edit_role" : "editor",
    "owner_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1",
```

```
    "dataclass_name" : "string",
    "dataclass_id" : "string",
    "unaudited_version_id" : "string",
    "reject_version_id" : "string"
  }
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class DeletePlaybookSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        DeletePlaybookRequest request = new DeletePlaybookRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withPlaybookId("{playbook_id}");
        try {
            DeletePlaybookResponse response = client.deletePlaybook(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeletePlaybookRequest()
        request.workspace_id = "{workspace_id}"
        request.playbook_id = "{playbook_id}"
        response = client.delete_playbook(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
```



```
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeletePlaybookRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.PlaybookId = "{playbook_id}"
    response, err := client.DeletePlaybook(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.4.7 修改剧本

功能介绍

修改剧本

调用方法

请参见**如何调用API**。

URI

PUT /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/{playbook_id}

表 4-371 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
playbook_id	是	String	剧本ID

请求参数

表 4-372 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-373 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	剧本名称
description	否	String	描述
enabled	否	Boolean	是否启用
active_version_id	否	String	启用的剧本版本ID

响应参数

状态码： 200

表 4-374 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-375 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息

参数	参数类型	描述
data	PlaybookInfo object	剧本详情信息

表 4-376 PlaybookInfo

参数	参数类型	描述
id	String	剧本ID
name	String	剧本名称
description	String	描述信息
create_time	String	剧本创建时间
update_time	String	剧本更新时间
project_id	String	项目ID
version_id	String	剧本版本ID
enabled	Boolean	是否启用
workspace_id	String	工作空间ID
approve_role	String	审核用户角色
user_role	String	用户角色
edit_role	String	编辑用户角色
owner_id	String	所有者ID
version	String	版本号
dataclass_name	String	数据类名称
dataclass_id	String	数据类ID
unaudited_version_id	String	待审核剧本版本ID
reject_version_id	String	已驳回剧本版本ID

状态码： 400

表 4-377 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-378 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

修改一个剧本，剧本名称为MyXXX，描述信息为This my XXXX，开启状态为已开启，启用剧本ID为active_version_id。

```
{
  "name" : "MyXXX",
  "description" : "This my XXXX",
  "enabled" : true,
  "active_version_id" : "active_version_id"
}
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "description" : "This my XXXX",
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "enabled" : true,
    "workspace_id" : "string",
    "approve_role" : "approve",
    "user_role" : "string",
    "edit_role" : "editor",
    "owner_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1",
    "dataclass_name" : "string",
    "dataclass_id" : "string",
    "unaudited_version_id" : "string",
    "reject_version_id" : "string"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改一个剧本，剧本名称为MyXXX，描述信息为This my XXXX，开启状态为已开启，启用剧本ID为active_version_id。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class UpdatePlaybookSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdatePlaybookRequest request = new UpdatePlaybookRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withPlaybookId("{playbook_id}");
        ModifyPlaybookInfo body = new ModifyPlaybookInfo();
        body.withActiveVersionId("active_version_id");
        body.withEnabled(true);
        body.withDescription("This my XXXX");
        body.withName("MyXXX");
        request.withBody(body);
        try {
            UpdatePlaybookResponse response = client.updatePlaybook(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

修改一个剧本，剧本名称为MyXXX，描述信息为This my XXXX，开启状态为已开启，启用剧本ID为active_version_id。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *
```

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdatePlaybookRequest()
        request.workspace_id = "{workspace_id}"
        request.playbook_id = "{playbook_id}"
        request.body = ModifyPlaybookInfo(
            active_version_id="active_version_id",
            enabled=True,
            description="This my XXXX",
            name="MyXXX"
        )
        response = client.update_playbook(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

修改一个剧本，剧本名称为MyXXX，描述信息为This my XXXX，开启状态为已开启，启用剧本ID为active_version_id。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
```

```
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

    request := &model.UpdatePlaybookRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.PlaybookId = "{playbook_id}"
    activeVersionIdModifyPlaybookInfo:= "active_version_id"
    enabledModifyPlaybookInfo:= true
    descriptionModifyPlaybookInfo:= "This my XXXX"
    nameModifyPlaybookInfo:= "MyXXX"
    request.Body = &model.ModifyPlaybookInfo{
        ActiveVersionId: &activeVersionIdModifyPlaybookInfo,
        Enabled: &enabledModifyPlaybookInfo,
        Description: &descriptionModifyPlaybookInfo,
        Name: &nameModifyPlaybookInfo,
    }
    response, err := client.UpdatePlaybook(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.5 告警规则管理

4.5.1 列出告警规则

功能介绍

List alert rules

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/siem/alert-rules

表 4-379 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目 ID。Project ID.
workspace_id	是	String	工作空间 ID。Workspace ID.

表 4-380 Query 参数

参数	是否必选	参数类型	描述
offset	是	Long	偏移量。Offset.
limit	是	Long	条数。Limit.
sort_key	否	String	排序字段。Sort key
sort_dir	否	String	排序顺序，顺序、逆序。Sort direction, asc, desc。
pipe_id	否	String	数据管道 ID。Pipe ID.
rule_name	否	String	告警规则名称。Alert rule name.
rule_id	否	String	告警规则 ID。Alert rule ID.
status	否	Array of strings	启用状态，启用、停用。Status, enabled, disabled.
severity	否	Array of strings	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL

请求参数

表 4-381 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。 IAM user token, fetch from IAM api.

响应参数

状态码： 200

表 4-382 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

表 4-383 响应 Body 参数

参数	参数类型	描述
count	Long	总数量。Total count.
records	Array of **AlertRule** objects	告警模型。Alert rules.

表 4-384 AlertRule

参数	参数类型	描述
rule_id	String	告警规则 ID。Alert rule ID.
pipe_id	String	数据管道 ID。Pipe ID.
pipe_name	String	数据管道名称。Pipe name.
create_by	String	创建人。Create by.
create_time	Long	创建时间。Create time.
update_by	String	更新人。Update by.
update_time	Long	更新时间。Update time.
delete_time	Long	删除时间。Delete time.
rule_name	String	告警规则名称。Alert rule name.
query	String	查询语句。Query.
query_type	String	查询语法，SQL。Query type. SQL.
status	String	启用状态，启用、停用。Status, enabled, disabled.
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL

参数	参数类型	描述
custom_properties	Map<String,String>	自定义扩展信息。Custom properties.
event_grouping	Boolean	告警分组。Event grouping.
schedule	**Schedule** object	调度规则。Schedule Rule.
triggers	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.

表 4-385 Schedule

参数	参数类型	描述
frequency_interval	Integer	调度间隔。Frequency interval.
frequency_unit	String	调度间隔单位，分钟、小时、天。Frequency unit. MINUTE, HOUR, DAY.
period_interval	Integer	时间窗口间隔。Period interval.
period_unit	String	时间窗口单位，分钟、小时、天。Period unit. MINUTE, HOUR, DAY.
delay_interval	Integer	延迟间隔。Delay interval
overtime_interval	Integer	超时间隔。Overtime interval

表 4-386 AlertRuleTrigger

参数	参数类型	描述
mode	String	模式，数量。Mode. COUNT.
operator	String	操作符，等于、不等于、大于、小于。operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	String	expression
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	Integer	accumulated_times

状态码： 400

表 4-387 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

请求示例

无

响应示例

状态码： 200

成功

```
{
  "count" : 9223372036854776000,
  "records" : [ {
    "rule_id" : "443a0117-1aa4-4595-ad4a-796fad4d4950",
    "pipe_id" : "772fb35b-83bc-46c9-a0b1-ebe31070a889",
    "create_by" : "582dd19dd99d4505a1d7929dc943b169",
    "create_time" : 1665221214,
    "update_by" : "582dd19dd99d4505a1d7929dc943b169",
    "update_time" : 1665221214,
    "delete_time" : 0,
    "rule_name" : "Alert rule",
    "query" : "* | select status, count(*) as count group by status",
    "query_type" : "SQL",
    "status" : "ENABLED",
    "severity" : "TIPS",
    "custom_properties" : {
      "references" : "https://localhost/references",
      "maintainer" : "isap"
    },
    "event_grouping" : true,
    "schedule" : {
      "frequency_interval" : 5,
      "frequency_unit" : "MINUTE",
      "period_interval" : 5,
      "period_unit" : "MINUTE",
      "delay_interval" : 2,
      "overtime_interval" : 10
    },
    "triggers" : [ {
      "mode" : "COUNT",
      "operator" : "GT",
      "expression" : 10,
      "severity" : "TIPS"
    } ]
  } ]
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListAlertRulesSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAlertRulesRequest request = new ListAlertRulesRequest();
        request.withWorkspaceId("{workspace_id}");
        try {
            ListAlertRulesResponse response = client.listAlertRules(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

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

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

try:
    request = ListAlertRulesRequest()
    request.workspace_id = "{workspace_id}"
    response = client.list_alert_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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAlertRulesRequest{}
    request.WorkspaceId = "{workspace_id}"
    response, err := client.ListAlertRules(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	成功
400	请求失败

错误码

请参见**错误码**。

4.5.2 创建告警规则

功能介绍

Create alert rule

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/siem/alert-rules

表 4-388 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目 ID。Project ID.
workspace_id	是	String	工作空间 ID。Workspace ID.

请求参数

表 4-389 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。 IAM user token, fetch from IAM api.

表 4-390 请求 Body 参数

参数	是否必选	参数类型	描述
pipe_id	是	String	数据管道 ID。Pipe ID.
rule_name	是	String	告警规则名称。Alert rule name.
description	否	String	描述。Description.
query	是	String	查询语句。Query.
query_type	否	String	查询语法，SQL。Query type. SQL.
status	否	String	启用状态，启用、停用。Status, enabled, disabled.
severity	否	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
custom_properties	否	Map<String,String>	自定义扩展信息。Custom properties.
alert_type	否	Map<String,String>	告警类型。Alert type.
event_grouping	否	Boolean	告警分组。Event grouping.
suspression	否	Boolean	告警抑制。Suspression.
simulation	否	Boolean	模拟告警。Simulation.
schedule	是	**Schedule** object	
triggers	是	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.
pipe_name	是	String	管道名称
alert_name	是	String	告警名称
alert_description	否	String	告警描述
alert_remediation	否	String	修复建议
accumulated_times	否	Integer	执行次数

表 4-391 Schedule

参数	是否必选	参数类型	描述
frequency_interval	是	Integer	调度间隔。Frequency interval.
frequency_unit	是	String	调度间隔单位，分钟、小时、天。Frequency unit. MINUTE, HOUR, DAY.
period_interval	是	Integer	时间窗口间隔。Period interval.
period_unit	是	String	时间窗口单位，分钟、小时、天。Period unit. MINUTE, HOUR, DAY.
delay_interval	否	Integer	延迟间隔。Delay interval
overtime_interval	否	Integer	超时间隔。Overtime interval

表 4-392 AlertRuleTrigger

参数	是否必选	参数类型	描述
mode	否	String	模式，数量。Mode. COUNT.
operator	否	String	操作符，等于、不等于、大于、小于。 operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	是	String	expression
severity	否	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	否	Integer	accumulated_times

响应参数

状态码： 200

表 4-393 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

表 4-394 响应 Body 参数

参数	参数类型	描述
rule_id	String	告警规则 ID。Alert rule ID.
pipe_id	String	数据管道 ID。Pipe ID.
pipe_name	String	数据管道名称。Pipe name.
create_by	String	创建人。Create by.
create_time	Long	创建时间。Create time.
update_by	String	更新人。Update by.
update_time	Long	更新时间。Update time.
delete_time	Long	删除时间。Delete time.
rule_name	String	告警规则名称。Alert rule name.
query	String	查询语句。Query.
query_type	String	查询语法，SQL。Query type. SQL.
status	String	启用状态，启用、停用。Status, enabled, disabled.
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
custom_properties	Map<String,String>	自定义扩展信息。Custom properties.
event_grouping	Boolean	告警分组。Event grouping.
schedule	**Schedule** object	调度规则。Schedule Rule.
triggers	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.

表 4-395 Schedule

参数	参数类型	描述
frequency_interval	Integer	调度间隔。Frequency interval.
frequency_unit	String	调度间隔单位，分钟、小时、天。Frequency unit. MINUTE, HOUR, DAY.
period_interval	Integer	时间窗口间隔。Period interval.
period_unit	String	时间窗口单位，分钟、小时、天。Period unit. MINUTE, HOUR, DAY.
delay_interval	Integer	延迟间隔。Delay interval
overtime_interval	Integer	超时间隔。Overtime interval

表 4-396 AlertRuleTrigger

参数	参数类型	描述
mode	String	模式，数量。Mode. COUNT.
operator	String	操作符，等于、不等于、大于、小于。operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	String	expression
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	Integer	accumulated_times

状态码： 400

表 4-397 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

请求示例

创建一条告警规则，告警规则所属的管道ID为772fb35b-83bc-46c9-a0b1-ebe31070a889，告警规则名称为Alert rule，查询类型为SQL，状态为启用。

```
{
  "pipe_id" : "772fb35b-83bc-46c9-a0b1-ebe31070a889",
  "pipe_name" : "sec-hss-alarm",
  "rule_name" : "Alert rule",
  "description" : "An alert rule",
  "query" : "* | select status, count(*) as count group by status",
  "query_type" : "SQL",
  "status" : "ENABLED",
  "severity" : "TIPS",
  "alert_name" : "test",
  "custom_properties" : {
    "references" : "https://localhost/references",
    "maintainer" : "isap"
  },
  "event_grouping" : false,
  "suspression" : false,
  "simulation" : false,
  "accumulated_times" : 1,
  "schedule" : {
    "frequency_interval" : 5,
    "frequency_unit" : "MINUTE",
    "period_interval" : 5,
    "period_unit" : "MINUTE",
    "delay_interval" : 2,
    "overtime_interval" : 10
  },
  "triggers" : [ {
    "mode" : "COUNT",
    "operator" : "GT",
    "expression" : 10,
    "severity" : "TIPS",
    "accumulated_times" : 1
  } ]
}
```

响应示例

状态码： 200

请求成功

```
{
  "rule_id" : "443a0117-1aa4-4595-ad4a-796fad4d4950",
  "pipe_id" : "772fb35b-83bc-46c9-a0b1-ebe31070a889",
  "create_by" : "582dd19dd99d4505a1d7929dc943b169",
  "create_time" : 1665221214,
  "update_by" : "582dd19dd99d4505a1d7929dc943b169",
  "update_time" : 1665221214,
  "delete_time" : 0,
  "rule_name" : "Alert rule",
  "query" : "* | select status, count(*) as count group by status",
  "query_type" : "SQL",
  "status" : "ENABLED",
  "severity" : "TIPS",
  "custom_properties" : {
    "references" : "https://localhost/references",
    "maintainer" : "isap"
  },
  "event_grouping" : true,
  "schedule" : {
    "frequency_interval" : 5,
    "frequency_unit" : "MINUTE",
    "period_interval" : 5,
    "period_unit" : "MINUTE",
    "delay_interval" : 2,
    "overtime_interval" : 10
  },
  "triggers" : [ {
    "mode" : "COUNT",
```

```
    "operator" : "GT",
    "expression" : 10,
    "severity" : "TIPS"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一条告警规则，告警规则所属的管道ID为772fb35b-83bc-46c9-a0b1-ebe31070a889，告警规则名称为Alert rule，查询类型为SQL，状态为启用。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class CreateAlertRuleSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateAlertRuleRequest request = new CreateAlertRuleRequest();
        request.withWorkspaceId("{workspace_id}");
        CreateAlertRuleRequestBody body = new CreateAlertRuleRequestBody();
        List<AlertRuleTrigger> listbodyTriggers = new ArrayList<>();
        listbodyTriggers.add(
            new AlertRuleTrigger()
                .withMode(AlertRuleTrigger.ModeEnum.fromValue("COUNT"))
                .withOperator(AlertRuleTrigger.OperatorEnum.fromValue("GT"))
                .withExpression("10")
                .withSeverity(AlertRuleTrigger.SeverityEnum.fromValue("TIPS"))
                .withAccumulatedTimes(1)
        );
        Schedule schedulebody = new Schedule();
        schedulebody.withFrequencyInterval(5)
            .withFrequencyUnit(Schedule.FrequencyUnitEnum.fromValue("MINUTE"))
```

```
            .withPeriodInterval(5)
            .withPeriodUnit(Schedule.PeriodUnitEnum.fromValue("MINUTE"))
            .withDelayInterval(2)
            .withOvertimeInterval(10);
        Map<String, String> listbodyCustomProperties = new HashMap<>();
        listbodyCustomProperties.put("references", "https://localhost/references");
        listbodyCustomProperties.put("maintainer", "isap");
        body.withAccumulatedTimes(1);
        body.withAlertName("test");
        body.withPipeName("sec-hss-alarm");
        body.withTriggers(listbodyTriggers);
        body.withSchedule(schedulebody);
        body.withSimulation(false);
        body.withSuspression(false);
        body.withEventGrouping(false);
        body.withCustomProperties(listbodyCustomProperties);
        body.withSeverity(CreateAlertRuleRequestBody.SeverityEnum.fromValue("TIPS"));
        body.withStatus(CreateAlertRuleRequestBody.StatusEnum.fromValue("ENABLED"));
        body.withQueryType(CreateAlertRuleRequestBody.QueryTypeEnum.fromValue("SQL"));
        body.withQuery("* | select status, count(*) as count group by status");
        body.withDescription("An alert rule");
        body.withRuleName("Alert rule");
        body.withPipeId("772fb35b-83bc-46c9-a0b1-ebe31070a889");
        request.withBody(body);
        try {
            CreateAlertRuleResponse response = client.createAlertRule(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

创建一条告警规则，告警规则所属的管道ID为772fb35b-83bc-46c9-a0b1-ebe31070a889，告警规则名称为Alert rule，查询类型为SQL，状态为启用。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

```
        .build()

    try:
        request = CreateAlertRuleRequest()
        request.workspace_id = "{workspace_id}"
        listTriggersbody = [
            AlertRuleTrigger(
                mode="COUNT",
                operator="GT",
                expression="10",
                severity="TIPS",
                accumulated_times=1
            )
        ]
        schedulebody = Schedule(
            frequency_interval=5,
            frequency_unit="MINUTE",
            period_interval=5,
            period_unit="MINUTE",
            delay_interval=2,
            overtime_interval=10
        )
        listCustomPropertiesbody = {
            "references": "https://localhost/references",
            "maintainer": "isap"
        }
        request.body = CreateAlertRuleRequestBody(
            accumulated_times=1,
            alert_name="test",
            pipe_name="sec-hss-alarm",
            triggers=listTriggersbody,
            schedule=schedulebody,
            simulation=False,
            suspression=False,
            event_grouping=False,
            custom_properties=listCustomPropertiesbody,
            severity="TIPS",
            status="ENABLED",
            query_type="SQL",
            query="* | select status, count(*) as count group by status",
            description="An alert rule",
            rule_name="Alert rule",
            pipe_id="772fb35b-83bc-46c9-a0b1-ebe31070a889"
        )
        response = client.create_alert_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

创建一条告警规则，告警规则所属的管道ID为772fb35b-83bc-46c9-a0b1-ebe31070a889，告警规则名称为Alert rule，查询类型为SQL，状态为启用。

```
package main

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

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

```
risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateAlertRuleRequest{}
    request.WorkspaceId = "{workspace_id}"
    modeTriggers:= model.GetAlertRuleTriggerModeEnum().COUNT
    operatorTriggers:= model.GetAlertRuleTriggerOperatorEnum().GT
    severityTriggers:= model.GetAlertRuleTriggerSeverityEnum().TIPS
    accumulatedTimesTriggers:= int32(1)
    var listTriggersbody = []model.AlertRuleTrigger{
        {
            Mode: &modeTriggers,
            Operator: &operatorTriggers,
            Expression: "10",
            Severity: &severityTriggers,
            AccumulatedTimes: &accumulatedTimesTriggers,
        },
    }
    delayIntervalSchedule:= int32(2)
    overtimeIntervalSchedule:= int32(10)
    schedulebody := &model.Schedule{
        FrequencyInterval: int32(5),
        FrequencyUnit: model.GetScheduleFrequencyUnitEnum().MINUTE,
        PeriodInterval: int32(5),
        PeriodUnit: model.GetSchedulePeriodUnitEnum().MINUTE,
        DelayInterval: &delayIntervalSchedule,
        OvertimeInterval: &overtimeIntervalSchedule,
    }
    var listCustomPropertiesbody = map[string]string{
        "references": "https://localhost/references",
        "maintainer": "isap",
    }
    accumulatedTimesCreateAlertRuleRequestBody:= int32(1)
    simulationCreateAlertRuleRequestBody:= false
    suspressionCreateAlertRuleRequestBody:= false
    eventGroupingCreateAlertRuleRequestBody:= false
    severityCreateAlertRuleRequestBody:= model.GetCreateAlertRuleRequestBodySeverityEnum().TIPS
    statusCreateAlertRuleRequestBody:= model.GetCreateAlertRuleRequestBodyStatusEnum().ENABLED
    queryTypeCreateAlertRuleRequestBody:= model.GetCreateAlertRuleRequestBodyQueryTypeEnum().SQL
    descriptionCreateAlertRuleRequestBody:= "An alert rule"
    request.Body = &model.CreateAlertRuleRequestBody{
        AccumulatedTimes: &accumulatedTimesCreateAlertRuleRequestBody,
        AlertName: "test",
        PipeName: "sec-hss-alarm",
        Triggers: listTriggersbody,
        Schedule: schedulebody,
        Simulation: &simulationCreateAlertRuleRequestBody,
        Suspression: &suspressionCreateAlertRuleRequestBody,
        EventGrouping: &eventGroupingCreateAlertRuleRequestBody,
        CustomProperties: listCustomPropertiesbody,
        Severity: &severityCreateAlertRuleRequestBody,
        Status: &statusCreateAlertRuleRequestBody,
```

```
        QueryType: &queryTypeCreateAlertRuleRequestBody,
        Query: "* | select status, count(*) as count group by status",
        Description: &descriptionCreateAlertRuleRequestBody,
        RuleName: "Alert rule",
        PipeId: "772fb35b-83bc-46c9-a0b1-ebe31070a889",
    }
    response, err := client.CreateAlertRule(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	请求失败

错误码

请参见**错误码**。

4.5.3 删除告警规则

功能介绍

Delete alert rule

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/workspaces/{workspace_id}/siem/alert-rules

表 4-398 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目 ID。Project ID.
workspace_id	是	String	工作空间 ID。Workspace ID.

请求参数

表 4-399 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。 IAM user token, fetch from IAM api.

表 4-400 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of strings	告警规则ID数组

响应参数

状态码： 200

表 4-401 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

表 4-402 响应 Body 参数

参数	参数类型	描述
deleted	Boolean	是否删除.
fail_list	Array of **AlertRule** objects	Alert rule ID.
success_list	Array of **AlertRule** objects	Alert rule ID.

表 4-403 AlertRule

参数	参数类型	描述
rule_id	String	告警规则 ID。Alert rule ID.

参数	参数类型	描述
pipe_id	String	数据管道 ID。Pipe ID.
pipe_name	String	数据管道名称。Pipe name.
create_by	String	创建人。Create by.
create_time	Long	创建时间。Create time.
update_by	String	更新人。Update by.
update_time	Long	更新时间。Update time.
delete_time	Long	删除时间。Delete time.
rule_name	String	告警规则名称。Alert rule name.
query	String	查询语句。Query.
query_type	String	查询语法，SQL。Query type. SQL.
status	String	启用状态，启用、停用。Status, enabled, disabled.
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
custom_properties	Map<String,String>	自定义扩展信息。Custom properties.
event_grouping	Boolean	告警分组。Event grouping.
schedule	**Schedule** object	调度规则。Schedule Rule.
triggers	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.

表 4-404 Schedule

参数	参数类型	描述
frequency_interval	Integer	调度间隔。Frequency interval.
frequency_unit	String	调度间隔单位，分钟、小时、天。Frequency unit. MINUTE, HOUR, DAY.
period_interval	Integer	时间窗口间隔。Period interval.
period_unit	String	时间窗口单位，分钟、小时、天。Period unit. MINUTE, HOUR, DAY.
delay_interval	Integer	延迟间隔。Delay interval
overtime_interval	Integer	超时间隔。Overtime interval

表 4-405 AlertRuleTrigger

参数	参数类型	描述
mode	String	模式，数量。Mode. COUNT.
operator	String	操作符，等于、不等于、大于、小于。operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	String	expression
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	Integer	accumulated_times

状态码： 400

表 4-406 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

请求示例

删除告警规则，告警规则请求体为告警规则ID数组。

```
[ "612b7f41-da89-495b-a6a1-fdf14e4cc794" ]
```

响应示例

状态码： 200

请求成功

```
{
  "deleted" : true,
  "fail_list" : [ ],
  "success_list" : [ ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

删除告警规则，告警规则请求体为告警规则ID数组。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class DeleteAlertRuleSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteAlertRuleRequest request = new DeleteAlertRuleRequest();
        request.withWorkspaceId("{workspace_id}");
        List<String> listbodyBody = new ArrayList<>();
        listbodyBody.add("612b7f41-da89-495b-a6a1-fdf14e4cc794");
        request.withBody(listbodyBody);
        try {
            DeleteAlertRuleResponse response = client.deleteAlertRule(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

删除告警规则，告警规则请求体为告警规则ID数组。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *
```

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeleteAlertRuleRequest()
        request.workspace_id = "{workspace_id}"
        listBodybody = [
            "612b7f41-da89-495b-a6a1-fdf14e4cc794"
        ]
        request.body = listBodybody
        response = client.delete_alert_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

删除告警规则，告警规则请求体为告警规则ID数组。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteAlertRuleRequest{}
```

```
    request.WorkspaceId = "{workspace_id}"
    var listBodybody = []string{
        "612b7f41-da89-495b-a6a1-fdf14e4cc794",
    }
    request.Body = &listBodybody
    response, err := client.DeleteAlertRule(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	请求失败

错误码

请参见**错误码**。

4.5.4 查看告警规则

功能介绍

查看告警规则 Get alert rule

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/siem/alert-rules/{rule_id}

表 4-407 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目 ID。Project ID.
workspace_id	是	String	工作空间 ID。Workspace ID.
rule_id	是	String	告警规则 ID。Alert rule ID.

请求参数

表 4-408 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。 IAM user token, fetch from IAM api.

响应参数

状态码： 200

表 4-409 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

表 4-410 响应 Body 参数

参数	参数类型	描述
rule_id	String	告警规则 ID。Alert rule ID.
pipe_id	String	数据管道 ID。Pipe ID.
pipe_name	String	数据管道名称。Pipe name.
create_by	String	创建人。Create by.
create_time	Long	创建时间。Create time.
update_by	String	更新人。Update by.
update_time	Long	更新时间。Update time.
delete_time	Long	删除时间。Delete time.
rule_name	String	告警规则名称。Alert rule name.
query	String	查询语句。Query.
query_type	String	查询语法，SQL。Query type. SQL.
status	String	启用状态，启用、停用。Status, enabled, disabled.

参数	参数类型	描述
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
custom_properties	Map<String,String>	自定义扩展信息。Custom properties.
event_grouping	Boolean	告警分组。Event grouping.
schedule	**Schedule** object	调度规则。Schedule Rule.
triggers	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.

表 4-411 Schedule

参数	参数类型	描述
frequency_interval	Integer	调度间隔。Frequency interval.
frequency_unit	String	调度间隔单位，分钟、小时、天。Frequency unit. MINUTE, HOUR, DAY.
period_interval	Integer	时间窗口间隔。Period interval.
period_unit	String	时间窗口单位，分钟、小时、天。Period unit. MINUTE, HOUR, DAY.
delay_interval	Integer	延迟间隔。Delay interval
overtime_interval	Integer	超时间隔。Overtime interval

表 4-412 AlertRuleTrigger

参数	参数类型	描述
mode	String	模式，数量。Mode. COUNT.
operator	String	操作符，等于、不等于、大于、小于。operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	String	expression
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	Integer	accumulated_times

状态码： 400

表 4-413 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

请求示例

无

响应示例

状态码： 200

请求成功

```
{
  "rule_id" : "443a0117-1aa4-4595-ad4a-796fad4d4950",
  "pipe_id" : "772fb35b-83bc-46c9-a0b1-ebe31070a889",
  "create_by" : "582dd19dd99d4505a1d7929dc943b169",
  "create_time" : 1665221214,
  "update_by" : "582dd19dd99d4505a1d7929dc943b169",
  "update_time" : 1665221214,
  "delete_time" : 0,
  "rule_name" : "Alert rule",
  "query" : "* | select status, count(*) as count group by status",
  "query_type" : "SQL",
  "status" : "ENABLED",
  "severity" : "TIPS",
  "custom_properties" : {
    "references" : "https://localhost/references",
    "maintainer" : "isap"
  },
  "event_grouping" : true,
  "schedule" : {
    "frequency_interval" : 5,
    "frequency_unit" : "MINUTE",
    "period_interval" : 5,
    "period_unit" : "MINUTE",
    "delay_interval" : 2,
    "overtime_interval" : 10
  },
  "triggers" : [ {
    "mode" : "COUNT",
    "operator" : "GT",
    "expression" : 10,
    "severity" : "TIPS"
  } ]
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowAlertRuleSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowAlertRuleRequest request = new ShowAlertRuleRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withRuleId("{rule_id}");
        try {
            ShowAlertRuleResponse response = client.showAlertRule(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

```
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowAlertRuleRequest()
        request.workspace_id = "{workspace_id}"
        request.rule_id = "{rule_id}"
        response = client.show_alert_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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAlertRuleRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.RuleId = "{rule_id}"
    response, err := client.ShowAlertRule(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	请求失败

错误码

请参见**错误码**。

4.5.5 更新告警规则

功能介绍

Update alert rule

调用方法

请参见**如何调用API**。

URI

PUT /v1/{project_id}/workspaces/{workspace_id}/siem/alert-rules/{rule_id}

表 4-414 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目 ID。Project ID.
workspace_id	是	String	工作空间 ID。Workspace ID.
rule_id	是	String	告警规则 ID。Alert rule ID.

请求参数

表 4-415 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。 IAM user token, fetch from IAM api.

表 4-416 请求 Body 参数

参数	是否必选	参数类型	描述
rule_name	否	String	告警规则名称。Alert rule name.
description	否	String	描述。Description.
query	否	String	查询语句。Query.
query_type	否	String	查询语法，SQL。Query type. SQL.
status	否	String	启用状态，启用、停用。Status, enabled, disabled.
severity	否	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
custom_properties	否	Map<String,String>	自定义扩展信息。Custom properties.
alert_type	否	Map<String,String>	告警类型。Alert type.
event_grouping	否	Boolean	告警分组。Event grouping.
suppression	否	Boolean	告警抑制。Suppression
simulation	否	Boolean	模拟告警。Simulation.
schedule	否	**Schedule** object	调度规则。Schedule Rule.
triggers	否	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.

表 4-417 Schedule

参数	是否必选	参数类型	描述
frequency_interval	是	Integer	调度间隔。Frequency interval.
frequency_unit	是	String	调度间隔单位，分钟、小时、天。Frequency unit. MINUTE, HOUR, DAY.
period_interval	是	Integer	时间窗口间隔。Period interval.
period_unit	是	String	时间窗口单位，分钟、小时、天。Period unit. MINUTE, HOUR, DAY.
delay_interval	否	Integer	延迟间隔。Delay interval
overtime_interval	否	Integer	超时间隔。Overtime interval

表 4-418 AlertRuleTrigger

参数	是否必选	参数类型	描述
mode	否	String	模式，数量。Mode. COUNT.
operator	否	String	操作符，等于、不等于、大于、小于。 operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	是	String	expression
severity	否	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	否	Integer	accumulated_times

响应参数

状态码： 200

表 4-419 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

表 4-420 响应 Body 参数

参数	参数类型	描述
rule_id	String	告警规则 ID。Alert rule ID.
pipe_id	String	数据管道 ID。Pipe ID.
pipe_name	String	数据管道名称。Pipe name.
create_by	String	创建人。Create by.
create_time	Long	创建时间。Create time.
update_by	String	更新人。Update by.
update_time	Long	更新时间。Update time.
delete_time	Long	删除时间。Delete time.
rule_name	String	告警规则名称。Alert rule name.
query	String	查询语句。Query.
query_type	String	查询语法，SQL。Query type. SQL.
status	String	启用状态，启用、停用。Status, enabled, disabled.
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
custom_properties	Map<String,String>	自定义扩展信息。Custom properties.
event_grouping	Boolean	告警分组。Event grouping.
schedule	**Schedule** object	调度规则。Schedule Rule.
triggers	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.

表 4-421 Schedule

参数	参数类型	描述
frequency_interval	Integer	调度间隔。Frequency interval.
frequency_unit	String	调度间隔单位，分钟、小时、天。Frequency unit. MINUTE, HOUR, DAY.
period_interval	Integer	时间窗口间隔。Period interval.
period_unit	String	时间窗口单位，分钟、小时、天。Period unit. MINUTE, HOUR, DAY.
delay_interval	Integer	延迟间隔。Delay interval
overtime_interval	Integer	超时间隔。Overtime interval

表 4-422 AlertRuleTrigger

参数	参数类型	描述
mode	String	模式，数量。Mode. COUNT.
operator	String	操作符，等于、不等于、大于、小于。operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	String	expression
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	Integer	accumulated_times

状态码： 400

表 4-423 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

请求示例

更新一条告警规则，告警规则名称为Alert rule，查询类型为SQL，状态为启用，严重程度为提示。

```
{
  "rule_name" : "Alert rule",
  "query" : "* | select status, count(*) as count group by status",
  "query_type" : "SQL",
  "status" : "ENABLED",
  "severity" : "TIPS",
  "custom_properties" : {
    "references" : "https://localhost/references",
    "maintainer" : "isap"
  },
  "event_grouping" : true,
  "schedule" : {
    "frequency_interval" : 5,
    "frequency_unit" : "MINUTE",
    "period_interval" : 5,
    "period_unit" : "MINUTE",
    "delay_interval" : 2,
    "overtime_interval" : 10
  },
  "triggers" : [ {
    "mode" : "COUNT",
    "operator" : "GT",
    "expression" : 10,
    "severity" : "TIPS"
  } ]
}
```

响应示例

状态码： 200

请求成功

```
{
  "rule_id" : "443a0117-1aa4-4595-ad4a-796fad4d4950",
  "pipe_id" : "772fb35b-83bc-46c9-a0b1-ebe31070a889",
  "create_by" : "582dd19dd99d4505a1d7929dc943b169",
  "create_time" : 1665221214,
  "update_by" : "582dd19dd99d4505a1d7929dc943b169",
  "update_time" : 1665221214,
  "delete_time" : 0,
  "rule_name" : "Alert rule",
  "query" : "* | select status, count(*) as count group by status",
  "query_type" : "SQL",
  "status" : "ENABLED",
  "severity" : "TIPS",
  "custom_properties" : {
    "references" : "https://localhost/references",
    "maintainer" : "isap"
  },
  "event_grouping" : true,
  "schedule" : {
    "frequency_interval" : 5,
    "frequency_unit" : "MINUTE",
    "period_interval" : 5,
    "period_unit" : "MINUTE",
    "delay_interval" : 2,
    "overtime_interval" : 10
  },
  "triggers" : [ {
    "mode" : "COUNT",
    "operator" : "GT",
    "expression" : 10,
    "severity" : "TIPS"
  } ]
}
```

状态码

状态码	描述
200	请求成功
400	请求失败

错误码

请参见**错误码**。

4.5.6 模拟告警规则

功能介绍

Simulate alert rule

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/siem/alert-rules/simulation

表 4-424 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目 ID。Project ID.
workspace_id	是	String	工作空间 ID。Workspace ID.

请求参数

表 4-425 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。 IAM user token, fetch from IAM api.

表 4-426 请求 Body 参数

参数	是否必选	参数类型	描述
pipe_id	是	String	数据管道 ID。Pipe ID.
query	是	String	查询语句。Query.
query_type	否	String	查询语法，SQL。Query type. SQL.
from	是	Long	开始时间。Start time.
to	是	Long	结束时间。End time.
event_grouping	否	Boolean	告警分组。Event grouping.
triggers	是	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.

表 4-427 AlertRuleTrigger

参数	是否必选	参数类型	描述
mode	否	String	模式，数量。Mode. COUNT.
operator	否	String	操作符，等于、不等于、大于、小于。 operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	是	String	expression
severity	否	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	否	Integer	accumulated_times

响应参数

状态码： 200

表 4-428 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

表 4-429 响应 Body 参数

参数	参数类型	描述
alert_count	Integer	告警数量。Alert count.
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL

状态码： 400

表 4-430 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

请求示例

模拟一条告警规则，告警规则所属的管道ID为ead2769b-afb0-45dd-b9fa-a2953e6ac82f，查询类型为SQL，严重程度为提示。

```
{
  "pipe_id" : "ead2769b-afb0-45dd-b9fa-a2953e6ac82f",
  "query" : "* | select status, count(*) as count group by status",
  "query_type" : "SQL",
  "event_grouping" : true,
  "from" : 1665221214000,
  "to" : 1665546370000,
  "triggers" : [ {
    "mode" : "COUNT",
    "operator" : "GT",
    "expression" : 10,
    "severity" : "TIPS"
  } ]
}
```

响应示例

状态码： 200

请求成功

```
{
  "alert_count" : 100,
  "severity" : "TIPS"
}
```

SDK 代码示例

SDK代码示例如下。

Java

模拟一条告警规则，告警规则所属的管道ID为ead2769b-afb0-45dd-b9fa-a2953e6ac82f，查询类型为SQL，严重程度为提示。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class CreateAlertRuleSimulationSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateAlertRuleSimulationRequest request = new CreateAlertRuleSimulationRequest();
        request.withWorkspaceId("{workspace_id}");
        CreateAlertRuleSimulationRequestBody body = new CreateAlertRuleSimulationRequestBody();
        List<AlertRuleTrigger> listbodyTriggers = new ArrayList<>();
        listbodyTriggers.add(
            new AlertRuleTrigger()
                .withMode(AlertRuleTrigger.ModeEnum.fromValue("COUNT"))
                .withOperator(AlertRuleTrigger.OperatorEnum.fromValue("GT"))
                .withExpression("10")
                .withSeverity(AlertRuleTrigger.SeverityEnum.fromValue("TIPS"))
        );
        body.withTriggers(listbodyTriggers);
        body.withEventGrouping(true);
        body.withTo(1665546370000L);
        body.withFrom(1665221214000L);
        body.withQueryType(CreateAlertRuleSimulationRequestBody.QueryTypeEnum.fromValue("SQL"));
        body.withQuery("* | select status, count(*) as count group by status");
        body.withPipeId("ead2769b-afb0-45dd-b9fa-a2953e6ac82f");
        request.withBody(body);
        try {
            CreateAlertRuleSimulationResponse response = client.createAlertRuleSimulation(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

模拟一条告警规则，告警规则所属的管道ID为ead2769b-afb0-45dd-b9fa-a2953e6ac82f，查询类型为SQL，严重程度为提示。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateAlertRuleSimulationRequest()
        request.workspace_id = "{workspace_id}"
        listTriggersbody = [
            AlertRuleTrigger(
                mode="COUNT",
                operator="GT",
                expression="10",
                severity="TIPS"
            )
        ]
        request.body = CreateAlertRuleSimulationRequestBody(
            triggers=listTriggersbody,
            event_grouping=True,
            to=1665546370000,
            _from=1665221214000,
            query_type="SQL",
            query="* | select status, count(*) as count group by status",
            pipe_id="ead2769b-afb0-45dd-b9fa-a2953e6ac82f"
        )
        response = client.create_alert_rule_simulation(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

模拟一条告警规则，告警规则所属的管道ID为ead2769b-afb0-45dd-b9fa-a2953e6ac82f，查询类型为SQL，严重程度为提示。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateAlertRuleSimulationRequest{}
    request.WorkspaceId = "{workspace_id}"
    modeTriggers:= model.GetAlertRuleTriggerModeEnum().COUNT
    operatorTriggers:= model.GetAlertRuleTriggerOperatorEnum().GT
    severityTriggers:= model.GetAlertRuleTriggerSeverityEnum().TIPS
    var listTriggersbody = []model.AlertRuleTrigger{
        {
            Mode: &modeTriggers,
            Operator: &operatorTriggers,
            Expression: "10",
            Severity: &severityTriggers,
        },
    }
    eventGroupingCreateAlertRuleSimulationRequestBody:= true
    queryTypeCreateAlertRuleSimulationRequestBody:= model.GetCreateAlertRuleSimulationRequestBodyQueryTypeEnum().SQL
    request.Body = &model.CreateAlertRuleSimulationRequestBody{
        Triggers: listTriggersbody,
        EventGrouping: &eventGroupingCreateAlertRuleSimulationRequestBody,
        To: int64(1665546370000),
        From: int64(1665221214000),
        QueryType: &queryTypeCreateAlertRuleSimulationRequestBody,
        Query: "* | select status, count(*) as count group by status",
        PipeId: "ead2769b-afb0-45dd-b9fa-a2953e6ac82f",
    }
    response, err := client.CreateAlertRuleSimulation(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	请求失败

错误码

请参见**错误码**。

4.5.7 告警规则总览

功能介绍

List alert rule metrics

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/siem/alert-rules/metrics

表 4-431 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目 ID。Project ID.
workspace_id	是	String	工作空间 ID。Workspace ID.

请求参数

表 4-432 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。 IAM user token, fetch from IAM api.

响应参数

状态码： 200

表 4-433 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

表 4-434 响应 Body 参数

参数	参数类型	描述
category	String	指标类型，分组数量。Metric category. GROUP_COUNT.
metric	Map<String,Number>	指标值。Metric value.

状态码： 400

表 4-435 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

请求示例

无

响应示例

状态码： 200

请求成功

- 示例 1

```
{
  "category" : {
    "GROUP_COUNT" : null
  },
  "metric" : null
}
```

- 示例 2

```
{
  "category" : "GROUP_COUNT",
  "metric" : {
    "ENABLED" : 8,
    "DISABLED" : 2
  }
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListAlertRuleMetricsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAlertRuleMetricsRequest request = new ListAlertRuleMetricsRequest();
        request.withWorkspaceId("{workspace_id}");
        try {
            ListAlertRuleMetricsResponse response = client.listAlertRuleMetrics(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListAlertRuleMetricsRequest()
        request.workspace_id = "{workspace_id}"
        response = client.list_alert_rule_metrics(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

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

    request := &model.ListAlertRuleMetricsRequest{}
    request.WorkspaceId = "{workspace_id}"
    response, err := client.ListAlertRuleMetrics(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	请求失败

错误码

请参见**错误码**。

4.5.8 启用告警规则

功能介绍

Enable alert rule

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/siem/alert-rules/enable

表 4-436 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目 ID。Project ID.
workspace_id	是	String	工作空间 ID。Workspace ID.

请求参数

表 4-437 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。 IAM user token, fetch from IAM api.

表 4-438 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of strings	EnableAlertRuleRequestBody

响应参数

状态码： 200

表 4-439 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

表 4-440 响应 Body 参数

参数	参数类型	描述
fail_list	Array of **AlertRule** objects	Alert rule ID.
success_list	Array of **AlertRule** objects	Alert rule ID.

表 4-441 AlertRule

参数	参数类型	描述
rule_id	String	告警规则 ID。Alert rule ID.
pipe_id	String	数据管道 ID。Pipe ID.

参数	参数类型	描述
pipe_name	String	数据管道名称。Pipe name.
create_by	String	创建人。Create by.
create_time	Long	创建时间。Create time.
update_by	String	更新人。Update by.
update_time	Long	更新时间。Update time.
delete_time	Long	删除时间。Delete time.
rule_name	String	告警规则名称。Alert rule name.
query	String	查询语句。Query.
query_type	String	查询语法，SQL。Query type. SQL.
status	String	启用状态，启用、停用。Status, enabled, disabled.
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
custom_properties	Map<String,String>	自定义扩展信息。Custom properties.
event_grouping	Boolean	告警分组。Event grouping.
schedule	**Schedule** object	调度规则。Schedule Rule.
triggers	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.

表 4-442 Schedule

参数	参数类型	描述
frequency_interval	Integer	调度间隔。Frequency interval.
frequency_unit	String	调度间隔单位，分钟、小时、天。Frequency unit. MINUTE, HOUR, DAY.
period_interval	Integer	时间窗口间隔。Period interval.
period_unit	String	时间窗口单位，分钟、小时、天。Period unit. MINUTE, HOUR, DAY.
delay_interval	Integer	延迟间隔。Delay interval
overtime_interval	Integer	超时间隔。Overtime interval

表 4-443 AlertRuleTrigger

参数	参数类型	描述
mode	String	模式，数量。Mode. COUNT.
operator	String	操作符，等于、不等于、大于、小于。operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	String	expression
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	Integer	accumulated_times

状态码： 400

表 4-444 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

请求示例

启用告警规则,名称为123123

```
[ "123123" ]
```

响应示例

状态码： 200

请求成功

```
{
  "fail_list" : [ ],
  "success_list" : [ ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

启用告警规则,名称为123123

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class EnableAlertRuleSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        EnableAlertRuleRequest request = new EnableAlertRuleRequest();
        request.withWorkspaceId("{workspace_id}");
        List<String> listbodyBody = new ArrayList<>();
        listbodyBody.add("123123");
        request.withBody(listbodyBody);
        try {
            EnableAlertRuleResponse response = client.enableAlertRule(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

启用告警规则,名称为123123

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *
```

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = EnableAlertRuleRequest()
        request.workspace_id = "{workspace_id}"
        listBodybody = [
            "123123"
        ]
        request.body = listBodybody
        response = client.enable_alert_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

启用告警规则,名称为123123

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.EnableAlertRuleRequest{}
```

```
    request.WorkspaceId = "{workspace_id}"
    var listBodybody = []string{
        "123123",
    }
    request.Body = &listBodybody
    response, err := client.EnableAlertRule(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	请求失败

错误码

请参见**错误码**。

4.5.9 停用告警规则

功能介绍

Disable alert rule

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/siem/alert-rules/disable

表 4-445 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目 ID。Project ID.
workspace_id	是	String	工作空间 ID。Workspace ID.

请求参数

表 4-446 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。 IAM user token, fetch from IAM api.

表 4-447 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of strings	DisableAlertRuleRequestBody

响应参数

状态码： 200

表 4-448 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

表 4-449 响应 Body 参数

参数	参数类型	描述
fail_list	Array of **AlertRule** objects	Alert rule ID.
success_list	Array of **AlertRule** objects	Alert rule ID.

表 4-450 AlertRule

参数	参数类型	描述
rule_id	String	告警规则 ID。Alert rule ID.
pipe_id	String	数据管道 ID。Pipe ID.

参数	参数类型	描述
pipe_name	String	数据管道名称。Pipe name.
create_by	String	创建人。Create by.
create_time	Long	创建时间。Create time.
update_by	String	更新人。Update by.
update_time	Long	更新时间。Update time.
delete_time	Long	删除时间。Delete time.
rule_name	String	告警规则名称。Alert rule name.
query	String	查询语句。Query.
query_type	String	查询语法，SQL。Query type. SQL.
status	String	启用状态，启用、停用。Status, enabled, disabled.
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
custom_properties	Map<String,String>	自定义扩展信息。Custom properties.
event_grouping	Boolean	告警分组。Event grouping.
schedule	**Schedule** object	调度规则。Schedule Rule.
triggers	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.

表 4-451 Schedule

参数	参数类型	描述
frequency_interval	Integer	调度间隔。Frequency interval.
frequency_unit	String	调度间隔单位，分钟、小时、天。Frequency unit. MINUTE, HOUR, DAY.
period_interval	Integer	时间窗口间隔。Period interval.
period_unit	String	时间窗口单位，分钟、小时、天。Period unit. MINUTE, HOUR, DAY.
delay_interval	Integer	延迟间隔。Delay interval
overtime_interval	Integer	超时间隔。Overtime interval

表 4-452 AlertRuleTrigger

参数	参数类型	描述
mode	String	模式，数量。Mode. COUNT.
operator	String	操作符，等于、不等于、大于、小于。operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	String	expression
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	Integer	accumulated_times

状态码： 400

表 4-453 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

请求示例

停用告警规则,名称为123123

```
[ "123123" ]
```

响应示例

状态码： 200

请求成功

```
{
  "fail_list" : [ ],
  "success_list" : [ ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

停用告警规则,名称为123123

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class DisableAlertRuleSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        DisableAlertRuleRequest request = new DisableAlertRuleRequest();
        request.withWorkspaceId("{workspace_id}");
        List<String> listbodyBody = new ArrayList<>();
        listbodyBody.add("123123");
        request.withBody(listbodyBody);
        try {
            DisableAlertRuleResponse response = client.disableAlertRule(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

停用告警规则,名称为123123

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *
```

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DisableAlertRuleRequest()
        request.workspace_id = "{workspace_id}"
        listBodybody = [
            "123123"
        ]
        request.body = listBodybody
        response = client.disable_alert_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

停用告警规则,名称为123123

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DisableAlertRuleRequest{}
```

```
    request.WorkspaceId = "{workspace_id}"
    var listBodybody = []string{
        "123123",
    }
    request.Body = &listBodybody
    response, err := client.DisableAlertRule(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	请求失败

错误码

请参见**错误码**。

4.5.10 列出告警规则模板

功能介绍

List alert rule templates

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/siem/alert-rules/templates

表 4-454 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目 ID。Project ID.
workspace_id	是	String	工作空间 ID。Workspace ID.

表 4-455 Query 参数

参数	是否必选	参数类型	描述
offset	是	Long	偏移量。Offset.
limit	是	Long	条数。Limit.
sort_key	否	String	排序字段。Sort key
sort_dir	否	String	排序顺序，顺序、逆序。Sort direction, asc, desc。
severity	否	Array of strings	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL

请求参数

表 4-456 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。 IAM user token, fetch from IAM api.

响应参数

状态码： 200

表 4-457 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

表 4-458 响应 Body 参数

参数	参数类型	描述
count	Long	总数量。Total count.
records	Array of **AlertRuleTemplate** objects	告警规则模板。Alert rule templates.

表 4-459 AlertRuleTemplate

参数	参数类型	描述
template_id	String	告警规则模板 ID。Alert rule template ID.
update_time	Long	更新时间。Update time.
template_name	String	告警规则模板名称。Alert rule template name.
data_source	String	数据源。Data source.
version	String	版本。Version
query	String	查询语句。Query.
query_type	String	查询语法，SQL。Query type. SQL.
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
custom_properties	Map<String,String>	自定义扩展信息。Custom properties.
event_grouping	Boolean	告警分组。Event grouping.
schedule	**Schedule** object	调度规则。Schedule Rule.
triggers	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.

表 4-460 Schedule

参数	参数类型	描述
frequency_interval	Integer	调度间隔。Frequency interval.
frequency_unit	String	调度间隔单位，分钟、小时、天。Frequency unit. MINUTE, HOUR, DAY.
period_interval	Integer	时间窗口间隔。Period interval.
period_unit	String	时间窗口单位，分钟、小时、天。Period unit. MINUTE, HOUR, DAY.
delay_interval	Integer	延迟间隔。Delay interval
overtime_interval	Integer	超时间隔。Overtime interval

表 4-461 AlertRuleTrigger

参数	参数类型	描述
mode	String	模式，数量。Mode. COUNT.
operator	String	操作符，等于、不等于、大于、小于。operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	String	expression
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	Integer	accumulated_times

状态码： 400

表 4-462 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

请求示例

无

响应示例

状态码： 200

请求成功

```
{
  "count" : 9223372036854776000,
  "records" : [ {
    "template_id" : "443a0117-1aa4-4595-ad4a-796fad4d4950",
    "update_time" : 1665221214,
    "template_name" : "Alert rule template",
    "data_source" : "sec_hss_vul",
    "version" : "1.0.0",
    "query" : "* | select status, count(*) as count group by status",
    "query_type" : "SQL",
    "severity" : "TIPS",
    "custom_properties" : {
      "references" : "https://localhost/references",
      "maintainer" : "isap"
    },
    "event_grouping" : true,
    "schedule" : {
```

```
    "frequency_interval" : 5,
    "frequency_unit" : "MINUTE",
    "period_interval" : 5,
    "period_unit" : "MINUTE",
    "delay_interval" : 2,
    "overtime_interval" : 10
   },
   "triggers" : [ {
    "mode" : "COUNT",
    "operator" : "GT",
    "expression" : 10,
    "severity" : "TIPS"
   } ]
  } ]
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListAlertRuleTemplatesSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListAlertRuleTemplatesRequest request = new ListAlertRuleTemplatesRequest();
        request.withWorkspaceId("{workspace_id}");
        try {
            ListAlertRuleTemplatesResponse response = client.listAlertRuleTemplates(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListAlertRuleTemplatesRequest()
        request.workspace_id = "{workspace_id}"
        response = client.list_alert_rule_templates(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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



```
        WithProjectId(projectId).
        Build()

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAlertRuleTemplatesRequest{}
    request.WorkspaceId = "{workspace_id}"
    response, err := client.ListAlertRuleTemplates(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	请求失败

错误码

请参见**错误码**。

4.5.11 查看告警规则模板

功能介绍

List alert rule templates

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/siem/alert-rules/templates/{template_id}

表 4-463 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目 ID。Project ID.
workspace_id	是	String	工作空间 ID。Workspace ID.
template_id	是	String	告警规则模板 ID。Alert rule template ID.

请求参数

表 4-464 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。 IAM user token, fetch from IAM api.

响应参数

状态码： 200

表 4-465 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

表 4-466 响应 Body 参数

参数	参数类型	描述
template_id	String	告警规则模板 ID。Alert rule template ID.
update_time	Long	更新时间。Update time.
template_name	String	告警规则模板名称。Alert rule template name.
data_source	String	数据源。Data source.
version	String	版本。Version

参数	参数类型	描述
query	String	查询语句。Query.
query_type	String	查询语法，SQL。Query type. SQL.
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
custom_properties	Map<String,String>	自定义扩展信息。Custom properties.
event_grouping	Boolean	告警分组。Event grouping.
schedule	**Schedule** object	调度规则。Schedule Rule.
triggers	Array of **AlertRuleTrigger** objects	告警触发规则。Alert triggers.

表 4-467 Schedule

参数	参数类型	描述
frequency_interval	Integer	调度间隔。Frequency interval.
frequency_unit	String	调度间隔单位，分钟、小时、天。Frequency unit. MINUTE, HOUR, DAY.
period_interval	Integer	时间窗口间隔。Period interval.
period_unit	String	时间窗口单位，分钟、小时、天。Period unit. MINUTE, HOUR, DAY.
delay_interval	Integer	延迟间隔。Delay interval
overtime_interval	Integer	超时间隔。Overtime interval

表 4-468 AlertRuleTrigger

参数	参数类型	描述
mode	String	模式，数量。Mode. COUNT.
operator	String	操作符，等于、不等于、大于、小于。operator. EQ equal, NE not equal, GT greater than, LT less than.
expression	String	expression

参数	参数类型	描述
severity	String	严重程度，提示、低危、中危、高危、致命。Severity. TIPS, LOW, MEDIUM, HIGH, FATAL
accumulated_times	Integer	accumulated_times

状态码： 400

表 4-469 响应 Header 参数

参数	参数类型	描述
X-request-id	String	This field is the request ID number for task tracking. Format is request_uuid-timestamp-hostname.

请求示例

无

响应示例

状态码： 200

请求成功

```
{
  "template_id" : "443a0117-1aa4-4595-ad4a-796fad4d4950",
  "update_time" : 1665221214,
  "template_name" : "Alert rule template",
  "data_source" : "sec_hss_vul",
  "version" : "1.0.0",
  "query" : "* | select status, count(*) as count group by status",
  "query_type" : "SQL",
  "severity" : "TIPS",
  "custom_properties" : {
    "references" : "https://localhost/references",
    "maintainer" : "isap"
  },
  "event_grouping" : true,
  "schedule" : {
    "frequency_interval" : 5,
    "frequency_unit" : "MINUTE",
    "period_interval" : 5,
    "period_unit" : "MINUTE",
    "delay_interval" : 2,
    "overtime_interval" : 10
  },
  "triggers" : [ {
    "mode" : "COUNT",
    "operator" : "GT",
    "expression" : 10,
    "severity" : "TIPS"
```

```
  } ]
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowAlertRuleTemplateSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowAlertRuleTemplateRequest request = new ShowAlertRuleTemplateRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withTemplateId("{template_id}");
        try {
            ShowAlertRuleTemplateResponse response = client.showAlertRuleTemplate(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowAlertRuleTemplateRequest()
        request.workspace_id = "{workspace_id}"
        request.template_id = "{template_id}"
        response = client.show_alert_rule_template(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAlertRuleTemplateRequest{}
```

```
    request.WorkspaceId = "{workspace_id}"
    request.TemplateId = "{template_id}"
    response, err := client.ShowAlertRuleTemplate(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	请求失败

错误码

请参见**错误码**。

4.6 剧本版本管理

4.6.1 克隆剧本及版本

功能介绍

克隆剧本及版本

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}/clone

表 4-470 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

参数	是否必选	参数类型	描述
version_id	是	String	剧本版本ID

请求参数

表 4-471 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-472 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	名称

响应参数

状态码： 200

表 4-473 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-474 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	Error message
data	**PlaybookVersionInfo** object	剧本版本详情信息

表 4-475 PlaybookVersionInfo

参数	参数类型	描述
id	String	剧本版本ID
description	String	描述
create_time	String	创建时间
update_time	String	更新时间
project_id	String	项目ID
creator_id	String	创建者ID
modifier_id	String	修改者ID
playbook_id	String	剧本ID
version	String	版本号
enabled	Boolean	是否启用。（ true--已启用，false-未启用 ）
status	String	剧本版本状态，编辑中：EDITING 审核中：APPROVING 不通过：UNPASSED 已发布：PUBLISHED
action_strategy	String	执行策略. 目前仅支持异步并发执行，对应值为ASYNC
actions	Array of **ActionInfo** objects	剧本关联流程列表信息
rule_enable	Boolean	是否启用触发条件过滤器
rules	**RuleInfo** object	剧本触发规格信息
dataclass_id	String	数据类ID
trigger_type	String	剧本触发方式(EVENT--事件触发，TIMER--定时触发)
dataobject_create	Boolean	标识数据对象是否创建时触发剧本
dataobject_update	Boolean	标识数据对象是否更新时触发剧本
dataobject_delete	Boolean	标识数据对象是否删除时触发剧本
version_type	Integer	版本类型（ 0--草稿版本，1--正式版本)
rule_id	String	过滤规则ID
dataclass_name	String	数据类名称
approve_name	String	审核者

表 4-476 ActionInfo

参数	参数类型	描述
id	String	剧本流程动作ID
name	String	流程动作名称
description	String	描述
action_type	String	流程动作类型
action_id	String	流程ID
playbook_id	String	剧本ID
playbook_version_id	String	剧本版本ID
project_id	String	项目ID

表 4-477 RuleInfo

参数	参数类型	描述
id	String	规则ID
project_id	String	项目ID
rule	String	触发规则

状态码： 400

表 4-478 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-479 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

克隆一个剧本及其版本，剧本名称为name。

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

响应示例

状态码： 200

请求成功响应参数

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "description" : "This my XXXX",
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "creator_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "modifier_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "playbook_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1",
    "enabled" : true,
    "status" : "editing",
    "action_strategy" : "sync",
    "actions" : [ {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "MyXXX",
      "description" : "This my XXXX",
      "action_type" : "Workflow",
      "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "playbook_id" : "string",
      "playbook_version_id" : "string",
      "project_id" : "string"
    } ],
    "rule_enable" : true,
    "rules" : {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "rule" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    },
    "dataclass_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "trigger_type" : "event",
    "dataobject_create" : true,
    "dataobject_update" : true,
    "dataobject_delete" : true,
    "version_type" : 1,
    "rule_id" : "string",
    "dataclass_name" : "string",
    "approve_name" : "string"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

克隆一个剧本及其版本，剧本名称为name。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class CopyPlaybookVersionSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CopyPlaybookVersionRequest request = new CopyPlaybookVersionRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        CopyPlaybookInfo body = new CopyPlaybookInfo();
        body.withName("name");
        request.withBody(body);
        try {
            CopyPlaybookVersionResponse response = client.copyPlaybookVersion(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

克隆一个剧本及其版本，剧本名称为name。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

if __name__ == "__main__":
```

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CopyPlaybookVersionRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        request.body = CopyPlaybookInfo(
            name="name"
        )
        response = client.copy_playbook_version(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

克隆一个剧本及其版本，剧本名称为name。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CopyPlaybookVersionRequest{}
    request.WorkspaceId = "{workspace_id}"
```

```
    request.VersionId = "{version_id}"
    nameCopyPlaybookInfo:= "name"
    request.Body = &model.CopyPlaybookInfo{
        Name: &nameCopyPlaybookInfo,
    }
    response, err := client.CopyPlaybookVersion(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	请求成功响应参数
400	请求失败响应参数

错误码

请参见**错误码**。

4.6.2 查询剧本版本列表

功能介绍

查询剧本版本列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/{playbook_id}/versions

表 4-480 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
playbook_id	是	String	剧本ID

表 4-481 Query 参数

参数	是否必选	参数类型	描述
status	否	String	剧本版本状态，编辑中：EDITING 审核中：APPROVING 不通过：UNPASSED 已发布：PUBLISHED
enabled	否	Integer	启用/禁用
version_type	否	Integer	版本类型， 草稿版本：0 正式版本：1
offset	否	Integer	分页查询参数。用于指定查询结果的起始位置，从0开始
limit	否	Integer	分页查询参数，用于指定一次查询最多的结果数，从1开始

请求参数

表 4-482 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-483 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-484 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
size	Integer	分页查询数据大小
page	Integer	当前页码
total	Integer	总数
data	Array of **PlaybookVersionListEntity** objects	剧本版本列表信息

表 4-485 PlaybookVersionListEntity

参数	参数类型	描述
id	String	剧本版本ID
description	String	描述
create_time	String	创建时间
update_time	String	更新时间
project_id	String	项目ID
creator_id	String	创建者ID
modifier_id	String	修改者ID
playbook_id	String	剧本ID
version	String	版本号
enabled	Boolean	是否激活
status	String	状态. (EDITING--编辑中, APPROVING--审核中, UNPASSED--审核不通过, PUBLISHED--审核通过)
action_strategy	String	执行策略. 目前仅支持异步并发执行，对应值为ASYNC
rule_enable	Boolean	过滤规则是否启用
dataclass_id	String	数据类ID
trigger_type	String	触发方式. EVENT--事件触发, TIMER--定时触发
dataobject_create	Boolean	标识数据对象是否创建时触发剧本

参数	参数类型	描述
dataobject_update	Boolean	标识数据对象是否更新时触发剧本
dataobject_delete	Boolean	标识数据对象是否删除时触发剧本
version_type	Integer	版本类型
rule_id	String	过滤规则ID
dataclass_name	String	数据类名称
approve_name	String	审核者

状态码： 400

表 4-486 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-487 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "size" : 3,
  "page" : 10,
  "total" : 41,
  "data" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "description" : "This my XXXX",
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
```

```
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "creator_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "modifier_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "playbook_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1",
    "enabled" : true,
    "status" : "editing",
    "action_strategy" : "sync",
    "rule_enable" : true,
    "dataclass_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "trigger_type" : "event",
    "dataobject_create" : true,
    "dataobject_update" : true,
    "dataobject_delete" : true,
    "version_type" : 1,
    "rule_id" : "string",
    "dataclass_name" : "string",
    "approve_name" : "string"
  } ]
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListPlaybookVersionsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListPlaybookVersionsRequest request = new ListPlaybookVersionsRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withPlaybookId("{playbook_id}");
        try {
            ListPlaybookVersionsResponse response = client.listPlaybookVersions(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListPlaybookVersionsRequest()
        request.workspace_id = "{workspace_id}"
        request.playbook_id = "{playbook_id}"
        response = client.list_playbook_versions(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

```
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPlaybookVersionsRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.PlaybookId = "{playbook_id}"
    response, err := client.ListPlaybookVersions(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.6.3 创建剧本版本

功能介绍

创建剧本版本

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/{playbook_id}/versions

表 4-488 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
playbook_id	是	String	剧本ID

请求参数

表 4-489 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-490 请求 Body 参数

参数	是否必选	参数类型	描述
description	否	String	描述
workspace_id	否	String	工作空间ID
playbook_id	否	String	剧本ID
actions	否	Array of **ActionInfo** objects	关联流程列表
dataclass_id	否	String	数据类ID
rule_enable	否	Boolean	过滤规则是否启用
rule_id	否	String	过滤规则ID
trigger_type	否	String	触发方式. EVENT--事件触发, TIMER--定时触发
dataobject_create	否	Boolean	标识数据对象是否创建时触发剧本
dataobject_update	否	Boolean	标识数据对象是否更新时触发剧本

参数	是否必选	参数类型	描述
dataobject_delete	否	Boolean	标识数据对象是否删除时触发剧本
action_strategy	否	String	执行策略. 目前仅支持异步并发执行，对应值为ASYNC

表 4-491 ActionInfo

参数	是否必选	参数类型	描述
id	否	String	剧本流程动作ID
name	否	String	流程动作名称
description	否	String	描述
action_type	否	String	流程动作类型
action_id	否	String	流程ID
playbook_id	否	String	剧本ID
playbook_version_id	否	String	剧本版本ID
project_id	否	String	项目ID

响应参数

状态码： 200

表 4-492 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-493 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	Error message
data	**PlaybookVersionInfo** object	剧本版本详情信息

表 4-494 PlaybookVersionInfo

参数	参数类型	描述
id	String	剧本版本ID
description	String	描述
create_time	String	创建时间
update_time	String	更新时间
project_id	String	项目ID
creator_id	String	创建者ID
modifier_id	String	修改者ID
playbook_id	String	剧本ID
version	String	版本号
enabled	Boolean	是否启用。（ true--已启用，false-未启用 ）
status	String	剧本版本状态，编辑中：EDITING 审核中：APPROVING 不通过：UNPASSED 已发布：PUBLISHED
action_strategy	String	执行策略. 目前仅支持异步并发执行，对应值为ASYNC
actions	Array of **ActionInfo** objects	剧本关联流程列表信息
rule_enable	Boolean	是否启用触发条件过滤器
rules	**RuleInfo** object	剧本触发规格信息
dataclass_id	String	数据类ID
trigger_type	String	剧本触发方式(EVENT--事件触发，TIMER--定时触发)
dataobject_create	Boolean	标识数据对象是否创建时触发剧本
dataobject_update	Boolean	标识数据对象是否更新时触发剧本
dataobject_delete	Boolean	标识数据对象是否删除时触发剧本
version_type	Integer	版本类型（ 0--草稿版本，1--正式版本)
rule_id	String	过滤规则ID
dataclass_name	String	数据类名称
approve_name	String	审核者

表 4-495 ActionInfo

参数	参数类型	描述
id	String	剧本流程动作ID
name	String	流程动作名称
description	String	描述
action_type	String	流程动作类型
action_id	String	流程ID
playbook_id	String	剧本ID
playbook_version_id	String	剧本版本ID
project_id	String	项目ID

表 4-496 RuleInfo

参数	参数类型	描述
id	String	规则ID
project_id	String	项目ID
rule	String	触发规则

状态码： 400

表 4-497 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-498 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

创建一个剧本版本，剧本版本所属工作空间ID为string，剧本ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，数据类ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，规则为启用。

```
{
  "description" : "This my XXXX",
  "workspace_id" : "string",
  "playbook_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "actions" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "description" : "This my XXXX",
    "action_type" : "Workflow",
    "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "playbook_id" : "string",
    "playbook_version_id" : "string",
    "project_id" : "string"
  } ],
  "dataclass_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "rule_enable" : true,
  "rule_id" : "4185bbd2-9d18-4362-92cb-46df0b24fe4e",
  "trigger_type" : "event",
  "dataobject_create" : true,
  "dataobject_update" : true,
  "dataobject_delete" : true,
  "action_strategy" : "sync"
}
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "description" : "This my XXXX",
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "creator_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "modifier_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "playbook_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1",
    "enabled" : true,
    "status" : "editing",
    "action_strategy" : "sync",
    "actions" : [ {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "MyXXX",
      "description" : "This my XXXX",
      "action_type" : "Workflow",
      "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "playbook_id" : "string",
      "playbook_version_id" : "string",
      "project_id" : "string"
    } ],
    "rule_enable" : true,
    "rules" : {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "rule" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
```

```
  },
  "dataclass_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "trigger_type" : "event",
  "dataobject_create" : true,
  "dataobject_update" : true,
  "dataobject_delete" : true,
  "version_type" : 1,
  "rule_id" : "string",
  "dataclass_name" : "string",
  "approve_name" : "string"
 }
}
```

状态码

状态码	描述
200	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.6.4 查询剧本版本详情

功能介绍

Show playbook version version

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}

表 4-499 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
version_id	是	String	剧本版本ID

请求参数

表 4-500 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-501 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-502 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	Error message
data	**PlaybookVersionInfo** object	剧本版本详情信息

表 4-503 PlaybookVersionInfo

参数	参数类型	描述
id	String	剧本版本ID
description	String	描述
create_time	String	创建时间
update_time	String	更新时间
project_id	String	项目ID

参数	参数类型	描述
creator_id	String	创建者ID
modifier_id	String	修改者ID
playbook_id	String	剧本ID
version	String	版本号
enabled	Boolean	是否启用。（true--已启用，false-未启用）
status	String	剧本版本状态，编辑中：EDITING 审核中：APPROVING 不通过：UNPASSED 已发布：PUBLISHED
action_strategy	String	执行策略. 目前仅支持异步并发执行，对应值为ASYNC
actions	Array of **ActionInfo** objects	剧本关联流程列表信息
rule_enable	Boolean	是否启用触发条件过滤器
rules	**RuleInfo** object	剧本触发规格信息
dataclass_id	String	数据类ID
trigger_type	String	剧本触发方式(EVENT--事件触发，TIMER--定时触发)
dataobject_create	Boolean	标识数据对象是否创建时触发剧本
dataobject_update	Boolean	标识数据对象是否更新时触发剧本
dataobject_delete	Boolean	标识数据对象是否删除时触发剧本
version_type	Integer	版本类型（0--草稿版本，1--正式版本)
rule_id	String	过滤规则ID
dataclass_name	String	数据类名称
approve_name	String	审核者

表 4-504 ActionInfo

参数	参数类型	描述
id	String	剧本流程动作ID
name	String	流程动作名称
description	String	描述

参数	参数类型	描述
action_type	String	流程动作类型
action_id	String	流程ID
playbook_id	String	剧本ID
playbook_version_id	String	剧本版本ID
project_id	String	项目ID

表 4-505 RuleInfo

参数	参数类型	描述
id	String	规则ID
project_id	String	项目ID
rule	String	触发规则

状态码： 400

表 4-506 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-507 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "description" : "This my XXXX",
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "creator_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "modifier_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "playbook_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1",
    "enabled" : true,
    "status" : "editing",
    "action_strategy" : "sync",
    "actions" : [ {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "MyXXX",
      "description" : "This my XXXX",
      "action_type" : "Workflow",
      "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "playbook_id" : "string",
      "playbook_version_id" : "string",
      "project_id" : "string"
    } ],
    "rule_enable" : true,
    "rules" : {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "rule" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    },
    "dataclass_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "trigger_type" : "event",
    "dataobject_create" : true,
    "dataobject_update" : true,
    "dataobject_delete" : true,
    "version_type" : 1,
    "rule_id" : "string",
    "dataclass_name" : "string",
    "approve_name" : "string"
  }
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowPlaybookVersionSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowPlaybookVersionRequest request = new ShowPlaybookVersionRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        try {
            ShowPlaybookVersionResponse response = client.showPlaybookVersion(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowPlaybookVersionRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        response = client.show_playbook_version(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPlaybookVersionRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    response, err := client.ShowPlaybookVersion(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.6.5 删除剧本版本

功能介绍

删除剧本版本

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}

表 4-508 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
version_id	是	String	剧本版本ID

请求参数

表 4-509 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-510 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-511 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	响应消息

状态码： 400

表 4-512 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-513 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message"
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class DeletePlaybookVersionSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        DeletePlaybookVersionRequest request = new DeletePlaybookVersionRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        try {
            DeletePlaybookVersionResponse response = client.deletePlaybookVersion(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *
```

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeletePlaybookVersionRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        response = client.delete_playbook_version(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeletePlaybookVersionRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    response, err := client.DeletePlaybookVersion(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.6.6 更新剧本版本

功能介绍

更新剧本版本

调用方法

请参见**如何调用API**。

URI

PUT /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}

表 4-514 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
version_id	是	String	剧本版本ID

请求参数

表 4-515 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-516 请求 Body 参数

参数	是否必选	参数类型	描述
description	否	String	描述
workspace_id	否	String	工作空间ID
playbook_id	否	String	剧本ID
dataclass_id	否	String	数据类ID
rule_enable	否	Boolean	是否启用触发条件过滤器
enabled	否	Boolean	是否激活。(false:未激活, true:已激活)
status	否	String	状态(APPROVING:审核中，EDITING-编辑中，UNPASSED-审核未通过，PUBLISHED-已发布)
rule_id	否	String	规则ID
trigger_type	否	String	触发方式. EVENT--事件触发, TIMER--定时触发
dataobject_create	否	Boolean	数据对象是否创建时触发剧本
dataobject_update	否	Boolean	数据对象是否更新时触发剧本
dataobject_delete	否	Boolean	数据对象是否删除时触发剧本
action_strategy	否	String	执行策略. 目前仅支持异步并发执行，对应值为ASYNC

响应参数

状态码： 200

表 4-517 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-518 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	Error message
data	**PlaybookVersionInfo** object	剧本版本详情信息

表 4-519 PlaybookVersionInfo

参数	参数类型	描述
id	String	剧本版本ID
description	String	描述
create_time	String	创建时间
update_time	String	更新时间
project_id	String	项目ID
creator_id	String	创建者ID
modifier_id	String	修改者ID
playbook_id	String	剧本ID
version	String	版本号
enabled	Boolean	是否启用。（true--已启用，false-未启用）
status	String	剧本版本状态，编辑中：EDITING 审核中：APPROVING 不通过：UNPASSED 已发布：PUBLISHED
action_strategy	String	执行策略. 目前仅支持异步并发执行，对应值为ASYNC

参数	参数类型	描述
actions	Array of **ActionInfo** objects	剧本关联流程列表信息
rule_enable	Boolean	是否启用触发条件过滤器
rules	**RuleInfo** object	剧本触发规格信息
dataclass_id	String	数据类ID
trigger_type	String	剧本触发方式(EVENT--事件触发，TIMER--定时触发)
dataobject_create	Boolean	标识数据对象是否创建时触发剧本
dataobject_update	Boolean	标识数据对象是否更新时触发剧本
dataobject_delete	Boolean	标识数据对象是否删除时触发剧本
version_type	Integer	版本类型（0--草稿版本，1--正式版本)
rule_id	String	过滤规则ID
dataclass_name	String	数据类名称
approve_name	String	审核者

表 4-520 ActionInfo

参数	参数类型	描述
id	String	剧本流程动作ID
name	String	流程动作名称
description	String	描述
action_type	String	流程动作类型
action_id	String	流程ID
playbook_id	String	剧本ID
playbook_version_id	String	剧本版本ID
project_id	String	项目ID

表 4-521 RuleInfo

参数	参数类型	描述
id	String	规则ID
project_id	String	项目ID
rule	String	触发规则

状态码： 400

表 4-522 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-523 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

更新一个剧本版本，剧本版本所属工作空间ID为string，剧本ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，数据类ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，剧本状态为已启用。

```
{
  "description" : "This my XXXX",
  "workspace_id" : "string",
  "playbook_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "dataclass_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "rule_enable" : true,
  "enabled" : true,
  "status" : "UNPASSED",
  "rule_id" : "4185bbd2-9d18-4362-92cb-46df0b24fe4e",
  "trigger_type" : "event",
  "dataobject_create" : true,
  "dataobject_update" : true,
  "dataobject_delete" : true,
  "action_strategy" : "sync"
}
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "description" : "This my XXXX",
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "creator_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "modifier_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "playbook_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1",
    "enabled" : true,
    "status" : "editing",
    "action_strategy" : "sync",
    "actions" : [ {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "MyXXX",
      "description" : "This my XXXX",
      "action_type" : "Workflow",
      "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "playbook_id" : "string",
      "playbook_version_id" : "string",
      "project_id" : "string"
    } ],
    "rule_enable" : true,
    "rules" : {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "rule" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    },
    "dataclass_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "trigger_type" : "event",
    "dataobject_create" : true,
    "dataobject_update" : true,
    "dataobject_delete" : true,
    "version_type" : 1,
    "rule_id" : "string",
    "dataclass_name" : "string",
    "approve_name" : "string"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新一个剧本版本，剧本版本所属工作空间ID为string，剧本ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，数据类ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，剧本状态为已启用。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class UpdatePlaybookVersionSolution {
```

```
    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdatePlaybookVersionRequest request = new UpdatePlaybookVersionRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        ModifyPlaybookVersionInfo body = new ModifyPlaybookVersionInfo();
        body.withActionStrategy("sync");
        body.withDataobjectDelete(true);
        body.withDataobjectUpdate(true);
        body.withDataobjectCreate(true);
        body.withTriggerType("event");
        body.withRuleId("4185bbd2-9d18-4362-92cb-46df0b24fe4e");
        body.withStatus("UNPASSED");
        body.withEnabled(true);
        body.withRuleEnable(true);
        body.withDataclassId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        body.withPlaybookId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        body.withWorkspaceId("string");
        body.withDescription("This my XXXX");
        request.withBody(body);
        try {
            UpdatePlaybookVersionResponse response = client.updatePlaybookVersion(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

更新一个剧本版本，剧本版本所属工作空间ID为string，剧本ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，数据类ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，剧本状态为已启用。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *
```

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdatePlaybookVersionRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        request.body = ModifyPlaybookVersionInfo(
            action_strategy="sync",
            dataobject_delete=True,
            dataobject_update=True,
            dataobject_create=True,
            trigger_type="event",
            rule_id="4185bbd2-9d18-4362-92cb-46df0b24fe4e",
            status="UNPASSED",
            enabled=True,
            rule_enable=True,
            dataclass_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            playbook_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            workspace_id="string",
            description="This my XXXX"
        )
        response = client.update_playbook_version(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

更新一个剧本版本，剧本版本所属工作空间ID为string，剧本ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，数据类ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，剧本状态为已启用。

```
package main

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

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

```
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdatePlaybookVersionRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    actionStrategyModifyPlaybookVersionInfo:= "sync"
    dataobjectDeleteModifyPlaybookVersionInfo:= true
    dataobjectUpdateModifyPlaybookVersionInfo:= true
    dataobjectCreateModifyPlaybookVersionInfo:= true
    triggerTypeModifyPlaybookVersionInfo:= "event"
    ruleIdModifyPlaybookVersionInfo:= "4185bbd2-9d18-4362-92cb-46df0b24fe4e"
    statusModifyPlaybookVersionInfo:= "UNPASSED"
    enabledModifyPlaybookVersionInfo:= true
    ruleEnableModifyPlaybookVersionInfo:= true
    dataclassIdModifyPlaybookVersionInfo:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    playbookIdModifyPlaybookVersionInfo:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    workspaceIdModifyPlaybookVersionInfo:= "string"
    descriptionModifyPlaybookVersionInfo:= "This my XXXX"
    request.Body = &model.ModifyPlaybookVersionInfo{
        ActionStrategy: &actionStrategyModifyPlaybookVersionInfo,
        DataobjectDelete: &dataobjectDeleteModifyPlaybookVersionInfo,
        DataobjectUpdate: &dataobjectUpdateModifyPlaybookVersionInfo,
        DataobjectCreate: &dataobjectCreateModifyPlaybookVersionInfo,
        TriggerType: &triggerTypeModifyPlaybookVersionInfo,
        RuleId: &ruleIdModifyPlaybookVersionInfo,
        Status: &statusModifyPlaybookVersionInfo,
        Enabled: &enabledModifyPlaybookVersionInfo,
        RuleEnable: &ruleEnableModifyPlaybookVersionInfo,
        DataclassId: &dataclassIdModifyPlaybookVersionInfo,
        PlaybookId: &playbookIdModifyPlaybookVersionInfo,
        WorkspaceId: &workspaceIdModifyPlaybookVersionInfo,
        Description: &descriptionModifyPlaybookVersionInfo,
    }
    response, err := client.UpdatePlaybookVersion(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.7 剧本规则管理

4.7.1 查询剧本规则详情

功能介绍

查询剧本规则详情

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}/rules/{rule_id}

表 4-524 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
version_id	是	String	version Id value
rule_id	是	String	version Id value

请求参数

表 4-525 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-526 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-527 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**RuleInfo** object	剧本触发规格信息

表 4-528 RuleInfo

参数	参数类型	描述
id	String	规则ID
project_id	String	项目ID
rule	String	触发规则

状态码： 400

表 4-529 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-530 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "rule" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  }
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowPlaybookRuleSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowPlaybookRuleRequest request = new ShowPlaybookRuleRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        request.withRuleId("{rule_id}");
        try {
            ShowPlaybookRuleResponse response = client.showPlaybookRule(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowPlaybookRuleRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        request.rule_id = "{rule_id}"
        response = client.show_playbook_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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
```

```
variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPlaybookRuleRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    request.RuleId = "{rule_id}"
    response, err := client.ShowPlaybookRule(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.7.2 删除剧本规则

功能介绍

删除剧本规则

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}/rules/{rule_id}

表 4-531 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
version_id	是	String	剧本版本ID
rule_id	是	String	规则ID

请求参数

表 4-532 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-533 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-534 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	响应消息

状态码： 400

表 4-535 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-536 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message"
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class DeletePlaybookRuleSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        DeletePlaybookRuleRequest request = new DeletePlaybookRuleRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        request.withRuleId("{rule_id}");
        try {
            DeletePlaybookRuleResponse response = client.deletePlaybookRule(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeletePlaybookRuleRequest()
        request.workspace_id = "{workspace_id}"
```



```
        request.version_id = "{version_id}"
        request.rule_id = "{rule_id}"
        response = client.delete_playbook_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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeletePlaybookRuleRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    request.RuleId = "{rule_id}"
    response, err := client.DeletePlaybookRule(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.7.3 创建剧本规则

功能介绍

创建剧本规则

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}/rules

表 4-537 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
version_id	是	String	剧本版本ID

请求参数

表 4-538 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）。

参数	是否必选	参数类型	描述
content-type	是	String	application/json;charset=UTF-8

表 4-539 请求 Body 参数

参数	是否必选	参数类型	描述
rule	是	**ConditionInfo** object	剧本触发规则详情

表 4-540 ConditionInfo

参数	是否必选	参数类型	描述
expression_type	否	String	表达式类型。默认为common,事件触发剧本必填
conditions	否	Array of **ConditionItem** objects	触发条件。事件触发剧本必填
logics	否	Array of strings	条件逻辑组合。事件触发剧本必填
cron	否	String	Cron 表达式（定时任务）。定时触发剧本必填
schedule_type	否	String	定时重复类型(second--秒, hour--小时,day--天，week-周)。定时触发剧本必填
start_type	否	String	剧本开始执行类型，IMMEDIATELY--创建完成立即执行, CUSTOM--自定义执行时间。定时触发剧本必填
end_type	否	String	剧本结束执行类型，FOREVER--一直执行, CUSTOM--自定义结束时间。定时触发剧本必填
end_time	否	String	定时结束时间。定时触发剧本必填
repeat_range	否	String	执行时间段2021-01-30T23:00:00Z+0800。定时触发剧本必填
only_once	否	Boolean	是否只执行一次。定时触发剧本必填

参数	是否必选	参数类型	描述
execution_type	否	String	执行队列类型（PARALLEL-新任务与之前任务并行）。定时触发剧本必填

表 4-541 ConditionItem

参数	是否必选	参数类型	描述
name	否	String	条件名称
detail	否	String	条件详情
data	否	Array of strings	条件表达式数据

响应参数

状态码： 200

表 4-542 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-543 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**RuleInfo** object	剧本触发规格信息

表 4-544 RuleInfo

参数	参数类型	描述
id	String	规则ID
project_id	String	项目ID
rule	String	触发规则

状态码： 400

表 4-545 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-546 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

创建一条剧本规则，名称为condition_0，表达式类型为所有，

```
{
  "rule" : {
    "expression_type" : "common",
    "conditions" : [ {
      "name" : "condition_0",
      "detail" : "123",
      "data" : [ "handle_status, ==, Open" ]
    } ],
    "logics" : [ "condition_0" ]
  }
}
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "rule" : "{\"expression_type\":\"common\",\"conditions\":[{\"name\":\"condition_0\",\"data\":
[\"ref_order_id\",\"==\",\"123\"],\"detail\":\"123\"}],\"logics\":[\"condition_0\"]}"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一条剧本规则，名称为condition_0，表达式类型为所有，

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class CreatePlaybookRuleSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreatePlaybookRuleRequest request = new CreatePlaybookRuleRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        CreateRuleInfo body = new CreateRuleInfo();
        List<String> listRuleLogics = new ArrayList<>();
        listRuleLogics.add("condition_0");
        List<String> listConditionsData = new ArrayList<>();
        listConditionsData.add("handle_status, ==, Open");
        List<ConditionItem> listRuleConditions = new ArrayList<>();
        listRuleConditions.add(
            new ConditionItem()
                .withName("condition_0")
                .withDetail("123")
                .withData(listConditionsData)
        );
        ConditionInfo rulebody = new ConditionInfo();
        rulebody.withExpressionType("common")
            .withConditions(listRuleConditions)
            .withLogics(listRuleLogics);
        body.withRule(rulebody);
        request.withBody(body);
        try {
            CreatePlaybookRuleResponse response = client.createPlaybookRule(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

创建一条剧本规则，名称为condition_0，表达式类型为所有，

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreatePlaybookRuleRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        listLogicsRule = [
            "condition_0"
        ]
        listDataConditions = [
            "handle_status, ==, Open"
        ]
        listConditionsRule = [
            ConditionItem(
                name="condition_0",
                detail="123",
                data=listDataConditions
            )
        ]
        rulebody = ConditionInfo(
            expression_type="common",
            conditions=listConditionsRule,
            logics=listLogicsRule
        )
        request.body = CreateRuleInfo(
            rule=rulebody
        )
        response = client.create_playbook_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

创建一条剧本规则，名称为condition_0，表达式类型为所有，

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePlaybookRuleRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    var listLogicsRule = []string{
        "condition_0",
    }
    var listDataConditions = []string{
        "handle_status, ==, Open",
    }
    nameConditions:= "condition_0"
    detailConditions:= "123"
    var listConditionsRule = []model.ConditionItem{
        {
            Name: &nameConditions,
            Detail: &detailConditions,
            Data: &listDataConditions,
        },
    }
    expressionTypeRule:= "common"
    rulebody := &model.ConditionInfo{
        ExpressionType: &expressionTypeRule,
        Conditions: &listConditionsRule,
        Logics: &listLogicsRule,
    }
    request.Body = &model.CreateRuleInfo{
        Rule: rulebody,
    }
    response, err := client.CreatePlaybookRule(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.7.4 更新剧本规则

功能介绍

更新剧本规则

调用方法

请参见**如何调用API**。

URI

PUT /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}/rules/{rule_id}

表 4-547 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
version_id	是	String	剧本版本ID
rule_id	是	String	剧本规则ID

请求参数

表 4-548 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-549 请求 Body 参数

参数	是否必选	参数类型	描述
rule	否	**ConditionInfo** object	剧本触发规则详情

表 4-550 ConditionInfo

参数	是否必选	参数类型	描述
expression_type	否	String	表达式类型。默认为common,事件触发剧本必填
conditions	否	Array of **ConditionItem** objects	触发条件。事件触发剧本必填
logics	否	Array of strings	条件逻辑组合。事件触发剧本必填
cron	否	String	Cron 表达式（定时任务）。定时触发剧本必填
schedule_type	否	String	定时重复类型(second--秒, hour--小时,day--天，week-周)。定时触发剧本必填
start_type	否	String	剧本开始执行类型，IMMEDIATELY--创建完成立即执行, CUSTOM--自定义执行时间。定时触发剧本必填
end_type	否	String	剧本结束执行类型，FOREVER--一直执行, CUSTOM--自定义结束时间。定时触发剧本必填

参数	是否必选	参数类型	描述
end_time	否	String	定时结束时间。定时触发剧本必填
repeat_range	否	String	执行时间段 2021-01-30T23:00:00Z+0800。定时触发剧本必填
only_once	否	Boolean	是否只执行一次。定时触发剧本必填
execution_type	否	String	执行队列类型（PARALLEL-新任务与之前任务并行）。定时触发剧本必填

表 4-551 ConditionItem

参数	是否必选	参数类型	描述
name	否	String	条件名称
detail	否	String	条件详情
data	否	Array of strings	条件表达式数据

响应参数

状态码： 200

表 4-552 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-553 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**RuleInfo** object	剧本触发规格信息

表 4-554 RuleInfo

参数	参数类型	描述
id	String	规则ID
project_id	String	项目ID
rule	String	触发规则

状态码： 400

表 4-555 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-556 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

更新一条剧本规则，名称为condition_0，表达式类型为所有，

```
{
  "rule" : {
    "expression_type" : "common",
    "conditions" : [ {
      "name" : "condition_0",
      "detail" : "123",
      "data" : [ "handle_status, ==, Open" ]
    } ],
    "logics" : [ "condition_0" ]
  }
}
```

响应示例

状态码： 200

请求成功响应参数

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
```

```
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "rule" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新一条剧本规则，名称为condition_0，表达式类型为所有，

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class UpdatePlaybookRuleSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdatePlaybookRuleRequest request = new UpdatePlaybookRuleRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        request.withRuleId("{rule_id}");
        ModifyRuleInfo body = new ModifyRuleInfo();
        List<String> listRuleLogics = new ArrayList<>();
        listRuleLogics.add("condition_0");
        List<String> listConditionsData = new ArrayList<>();
        listConditionsData.add("handle_status, ==, Open");
        List<ConditionItem> listRuleConditions = new ArrayList<>();
        listRuleConditions.add(
            new ConditionItem()
                .withName("condition_0")
                .withDetail("123")
                .withData(listConditionsData)
        );
        ConditionInfo rulebody = new ConditionInfo();
        rulebody.withExpressionType("common")
            .withConditions(listRuleConditions)
```

```
            .withLogics(listRuleLogics);
        body.withRule(rulebody);
        request.withBody(body);
        try {
            UpdatePlaybookRuleResponse response = client.updatePlaybookRule(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

更新一条剧本规则，名称为condition_0，表达式类型为所有，

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = UpdatePlaybookRuleRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        request.rule_id = "{rule_id}"
        listLogicsRule = [
            "condition_0"
        ]
        listDataConditions = [
            "handle_status, ==, Open"
        ]
        listConditionsRule = [
            ConditionItem(
                name="condition_0",
                detail="123",
                data=listDataConditions
            )
        ]
        rulebody = ConditionInfo(
            expression_type="common",
```

```
            conditions=listConditionsRule,
            logics=listLogicsRule
        )
        request.body = ModifyRuleInfo(
            rule=rulebody
        )
        response = client.update_playbook_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

更新一条剧本规则，名称为condition_0，表达式类型为所有，

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdatePlaybookRuleRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    request.RuleId = "{rule_id}"
    var listLogicsRule = []string{
        "condition_0",
    }
    var listDataConditions = []string{
        "handle_status, ==, Open",
    }
    nameConditions:= "condition_0"
    detailConditions:= "123"
    var listConditionsRule = []model.ConditionItem{
        {
            Name: &nameConditions,
            Detail: &detailConditions,
            Data: &listDataConditions,
        },
    }
```

```
    expressionTypeRule:= "common"
    rulebody := &model.ConditionInfo{
        ExpressionType: &expressionTypeRule,
        Conditions: &listConditionsRule,
        Logics: &listLogicsRule,
    }
    request.Body = &model.ModifyRuleInfo{
        Rule: rulebody,
    }
    response, err := client.UpdatePlaybookRule(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	请求成功响应参数
400	请求失败响应参数

错误码

请参见**错误码**。

4.8 剧本实例管理

4.8.1 查询剧本实例列表

功能介绍

查询剧本实例列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/instances

表 4-557 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

表 4-558 Query 参数

参数	是否必选	参数类型	描述
status	否	String	剧本实例状态. (RUNNING--运行中、FINISHED--成功、FAILED--失败、RETRYING--重试中、TERMINATING--终止中、TERMINATED--已终止)
name	否	String	实例名称
playbook_name	否	String	剧本名称
dataclass_name	否	String	数据类名称
dataobject_name	否	String	数据对象名称
trigger_type	否	String	触发类型. TIMER--定时触发, EVENT--事件触发
from_date	否	String	查询起始时间
to_date	否	String	查询结束时间
limit	是	Integer	分页查询参数，用于指定一次查询最多的结果数，从1开始
offset	是	Integer	分页查询参数。用于指定查询结果的起始位置，从0开始

请求参数

表 4-559 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）。

参数	是否必选	参数类型	描述
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-560 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-561 响应 Body 参数

参数	参数类型	描述
count	Integer	总数
instances	Array of **PlaybookInstanceInfo** objects	剧本实例列表信息

表 4-562 PlaybookInstanceInfo

参数	参数类型	描述
id	String	剧本实例ID
name	String	剧本实例名称
project_id	String	项目ID
playbook	**PlaybookInfoRef** object	剧本信息
dataclass	**DataclassInfoRef** object	数据类信息
dataobject	**DataobjectInfo** object	数据对象详情
status	String	剧本实例状态. (RUNNING--运行中、FINISHED--成功、FAILED--失败、RETRYING--重试中、TERMINATING--终止中、TERMINATED--已终止)

参数	参数类型	描述
trigger_type	String	触发类型. TIMER--定时触发, EVENT--事件触发
start_time	String	创建时间
end_time	String	更新时间

表 4-563 PlaybookInfoRef

参数	参数类型	描述
id	String	剧本ID
version_id	String	剧本版本ID
name	String	名称
version	String	版本

表 4-564 DataclassInfoRef

参数	参数类型	描述
id	String	数据类ID
name	String	数据类名称

表 4-565 DataobjectInfo

参数	参数类型	描述
id	String	ID值
create_time	String	创建时间
update_time	String	更新时间
project_id	String	项目ID
dataclass_id	String	数据类ID
name	String	名称
content	String	数据内容

状态码： 400

表 4-566 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-567 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "count" : 41,
  "instances" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "playbook" : {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "version_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "version" : "v1.1.1"
    },
    "dataclass" : {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    },
    "dataobject" : {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    },
    "status" : "TERMINATED",
    "trigger_type" : "string",
    "start_time" : "2021-01-30T23:00:00Z+0800",
    "end_time" : "2021-01-30T23:00:00Z+0800"
  } ]
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListPlaybookInstancesSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListPlaybookInstancesRequest request = new ListPlaybookInstancesRequest();
        request.withWorkspaceId("{workspace_id}");
        try {
            ListPlaybookInstancesResponse response = client.listPlaybookInstances(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

```
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListPlaybookInstancesRequest()
        request.workspace_id = "{workspace_id}"
        response = client.list_playbook_instances(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPlaybookInstancesRequest{}
    request.WorkspaceId = "{workspace_id}"
    response, err := client.ListPlaybookInstances(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.8.2 查询剧本实例详情

功能介绍

Show playbook instance

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/instances/{instance_id}

表 4-568 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
instance_id	是	String	instance _id

请求参数

表 4-569 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-570 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-571 响应 Body 参数

参数	参数类型	描述
id	String	剧本实例ID
name	String	剧本实例名称
project_id	String	项目ID
playbook	**PlaybookInfoRef** object	剧本信息
dataclass	**DataclassInfoRef** object	数据类信息
dataobject	**DataobjectInfo** object	数据对象详情
status	String	剧本实例状态. (RUNNING--运行中、FINISHED--成功、FAILED--失败、RETRYING--重试中、TERMINATING--终止中、TERMINATED--已终止)
trigger_type	String	触发类型. TIMER--定时触发, EVENT--事件触发

参数	参数类型	描述
start_time	String	创建时间
end_time	String	更新时间

表 4-572 PlaybookInfoRef

参数	参数类型	描述
id	String	剧本ID
version_id	String	剧本版本ID
name	String	名称
version	String	版本

表 4-573 DataclassInfoRef

参数	参数类型	描述
id	String	数据类ID
name	String	数据类名称

表 4-574 DataobjectInfo

参数	参数类型	描述
id	String	ID值
create_time	String	创建时间
update_time	String	更新时间
project_id	String	项目ID
dataclass_id	String	数据类ID
name	String	名称
content	String	数据内容

状态码： 400

表 4-575 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-576 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

Instance Informations

```
{
  "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "name" : "MyXXX",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "playbook" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1"
  },
  "dataclass" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  },
  "dataobject" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  },
  "status" : "TERMINATED",
  "trigger_type" : "string",
  "start_time" : "2021-01-30T23:00:00Z+0800",
  "end_time" : "2021-01-30T23:00:00Z+0800"
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowPlaybookInstanceSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowPlaybookInstanceRequest request = new ShowPlaybookInstanceRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withInstanceId("{instance_id}");
        try {
            ShowPlaybookInstanceResponse response = client.showPlaybookInstance(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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



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

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

    try:
        request = ShowPlaybookInstanceRequest()
        request.workspace_id = "{workspace_id}"
        request.instance_id = "{instance_id}"
        response = client.show_playbook_instance(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPlaybookInstanceRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.InstanceId = "{instance_id}"
    response, err := client.ShowPlaybookInstance(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	Instance Informations
400	Error response

错误码

请参见**错误码**。

4.8.3 操作剧本实例

功能介绍

操作剧本实例

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/instances/{instance_id}/operation

表 4-577 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
instance_id	是	String	剧本实例ID

请求参数

表 4-578 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-579 请求 Body 参数

参数	是否必选	参数类型	描述
operation	否	String	操作类型。重试： RETRY 终止： TERMINATE

响应参数

状态码： 200

表 4-580 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-581 响应 Body 参数

参数	参数类型	描述
id	String	剧本实例ID
name	String	剧本实例名称
project_id	String	项目ID
playbook	**PlaybookInfoRef** object	剧本信息
dataclass	**DataclassInfoRef** object	数据类信息

参数	参数类型	描述
dataobject	DataobjectInfo object	数据对象详情
status	String	剧本实例状态. (RUNNING--运行中、FINISHED--成功、FAILED--失败、RETRYING--重试中、TERMINATING--终止中、TERMINATED--已终止)
trigger_type	String	触发类型. TIMER--定时触发, EVENT--事件触发
start_time	String	创建时间
end_time	String	更新时间

表 4-582 PlaybookInfoRef

参数	参数类型	描述
id	String	剧本ID
version_id	String	剧本版本ID
name	String	名称
version	String	版本

表 4-583 DataclassInfoRef

参数	参数类型	描述
id	String	数据类ID
name	String	数据类名称

表 4-584 DataobjectInfo

参数	参数类型	描述
id	String	ID值
create_time	String	创建时间
update_time	String	更新时间
project_id	String	项目ID
dataclass_id	String	数据类ID
name	String	名称

参数	参数类型	描述
content	String	数据内容

状态码： 400

表 4-585 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-586 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

重试所有操作剧本实例。

```
{
  "operation" : "RETRY"
}
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "name" : "MyXXX",
  "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "playbook" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version" : "v1.1.1"
  },
  "dataclass" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  },
  "dataobject" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  },
  "status" : "TERMINATED",
  "trigger_type" : "string",
```

```
  "start_time" : "2021-01-30T23:00:00Z+0800",
  "end_time" : "2021-01-30T23:00:00Z+0800"
}
```

SDK 代码示例

SDK代码示例如下。

Java

重试所有操作剧本实例。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ChangePlaybookInstanceSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ChangePlaybookInstanceRequest request = new ChangePlaybookInstanceRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withInstanceId("{instance_id}");
        OperationPlaybookInfo body = new OperationPlaybookInfo();
        body.withOperation("RETRY");
        request.withBody(body);
        try {
            ChangePlaybookInstanceResponse response = client.changePlaybookInstance(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

重试所有操作剧本实例。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ChangePlaybookInstanceRequest()
        request.workspace_id = "{workspace_id}"
        request.instance_id = "{instance_id}"
        request.body = OperationPlaybookInfo(
            operation="RETRY"
        )
        response = client.change_playbook_instance(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

重试所有操作剧本实例。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangePlaybookInstanceRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.InstanceId = "{instance_id}"
    operationOperationPlaybookInfo:= "RETRY"
    request.Body = &model.OperationPlaybookInfo{
        Operation: &operationOperationPlaybookInfo,
    }
    response, err := client.ChangePlaybookInstance(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.8.4 查询剧本拓扑关系

功能介绍

查询剧本拓扑关系

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/instances/{instance_id}/topology

表 4-587 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
instance_id	是	String	剧本实例ID

请求参数

表 4-588 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-589 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-590 响应 Body 参数

参数	参数类型	描述
count	Integer	总数
action_instances	Array of **ActionInstanceInfo** objects	流程实例列表

表 4-591 ActionInstanceInfo

参数	参数类型	描述
action	**ActionInfo** object	剧本流程动作信息
instance_log	**AuditLogInfo** object	剧本实例审计日志信息

表 4-592 ActionInfo

参数	参数类型	描述
id	String	剧本流程动作ID
name	String	流程动作名称
description	String	描述
action_type	String	流程动作类型
action_id	String	流程ID
playbook_id	String	剧本ID
playbook_version_id	String	剧本版本ID
project_id	String	项目ID

表 4-593 AuditLogInfo

参数	参数类型	描述
instance_type	String	实例类型（AOP_WORKFLOW--流程, SCRIPT--脚本, PLAYBOOK--剧本）
action_id	String	流程ID
action_name	String	流程名称
instance_id	String	实例ID
parent_instance_id	String	父节点实例ID
log_level	String	日志级别
input	String	输入
output	String	输出
error_msg	String	错误信息
start_time	String	开始时间

参数	参数类型	描述
end_time	String	结束时间
status	String	状态。(RUNNING--运行中、FINISHED--成功、FAILED--失败、RETRYING--重试中、TERMINATING--终止中、TERMINATED--已终止)
trigger_type	String	触发类型. TIMER--定时触发, EVENT--事件触发

状态码： 400

表 4-594 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-595 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "count" : 41,
  "action_instances" : [ {
    "action" : {
      "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "name" : "MyXXX",
      "description" : "This my XXXX",
      "action_type" : "Workflow",
      "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
      "playbook_id" : "string",
      "playbook_version_id" : "string",
      "project_id" : "string"
    },
    "instance_log" : {
```

```
    "instance_type" : "APP, AOP_WORKFLOW, SCRIPT, PLAYBOOK, TASK, DEBUG",
    "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "action_name" : "DisabledIp",
    "instance_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "parent_instance_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "log_level" : "DEBUG INFO WARN",
    "input" : "input",
    "output" : "output",
    "error_msg" : "error_msg",
    "start_time" : "2021-01-30T23:00:00Z",
    "end_time" : "2021-01-31T23:00:00Z",
    "status" : "CREATED, RUNNING, FINISHED, RETRYING, TERMINATING, TERMINATED, FAILED",
    "trigger_type" : "DEBUG, TIMER, EVENT, MANUAL"
  }
 } ]
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ShowPlaybookTopologySolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowPlaybookTopologyRequest request = new ShowPlaybookTopologyRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withInstanceId("{instance_id}");
        try {
            ShowPlaybookTopologyResponse response = client.showPlaybookTopology(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ShowPlaybookTopologyRequest()
        request.workspace_id = "{workspace_id}"
        request.instance_id = "{instance_id}"
        response = client.show_playbook_topology(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowPlaybookTopologyRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.InstanceId = "{instance_id}"
    response, err := client.ShowPlaybookTopology(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.8.5 查询剧本实例审计日志

功能介绍

查询剧本实例审计日志

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/instances/auditlogs

表 4-596 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

表 4-597 Query 参数

参数	是否必选	参数类型	描述
offset	是	Long	offset
limit	是	Long	limit
sort_key	否	String	sort_key
sort_dir	否	String	sort_dir. asc, desc

请求参数

表 4-598 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-599 请求 Body 参数

参数	是否必选	参数类型	描述
instance_type	否	String	实例类型（AOP_WORKFLOW--流程, SCRIPT--脚本, PLAYBOOK--剧本）
action_id	否	String	流程ID
action_name	否	String	流程名称
instance_id	否	String	实例ID
parent_instance_id	否	String	父节点实例ID

参数	是否必选	参数类型	描述
log_level	否	String	日志级别
input	否	String	输入
output	否	String	输出
error_msg	否	String	错误信息
start_time	否	String	开始时间
end_time	否	String	结束时间
status	否	String	状态。(RUNNING--运行中、FINISHED--成功、FAILED--失败、RETRYING--重试中、TERMINATING--终止中、TERMINATED--已终止)
trigger_type	否	String	触发类型. TIMER--定时触发, EVENT--事件触发

响应参数

状态码： 200

表 4-600 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-601 响应 Body 参数

参数	参数类型	描述
count	Integer	总条数
audit_logs	Array of **AuditLogInfo** objects	审计日志列表信息

表 4-602 AuditLogInfo

参数	参数类型	描述
instance_type	String	实例类型（AOP_WORKFLOW--流程, SCRIPT--脚本, PLAYBOOK--剧本）

参数	参数类型	描述
action_id	String	流程ID
action_name	String	流程名称
instance_id	String	实例ID
parent_instance_id	String	父节点实例ID
log_level	String	日志级别
input	String	输入
output	String	输出
error_msg	String	错误信息
start_time	String	开始时间
end_time	String	结束时间
status	String	状态。(RUNNING--运行中、FINISHED--成功、FAILED--失败、RETRYING--重试中、TERMINATING--终止中、TERMINATED--已终止)
trigger_type	String	触发类型. TIMER--定时触发, EVENT--事件触发

状态码： 400

表 4-603 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-604 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

查询实例类型为APP, AOP_WORKFLOW, SCRIPT, PLAYBOOK, TASK, DEBUG，动作id为909494e3-558e-46b6-a9eb-07a8e18ca62f，动作名称为DisabledIp，实例id为

909494e3-558e-46b6-a9eb-07a8e18ca62f，父实例id为909494e3-558e-46b6-a9eb-07a8e18ca62f，日志等级为DEBUG INFO WARN，输入为input，输出为output，错误信息为error_msg，开始时间2021-01-30 23：00：00，结束时间2021-01-31 23：00：00，状态为CREATED, RUNNING, FINISHED, RETRYING, TERMINATING, TERMINATED, FAILED，触发类型为DEBUG, TIMER, EVENT, MANUAL的剧本实例审计日志

```
{
  "instance_type" : "APP, AOP_WORKFLOW, SCRIPT, PLAYBOOK, TASK, DEBUG",
  "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "action_name" : "DisabledIp",
  "instance_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "parent_instance_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "log_level" : "DEBUG INFO WARN",
  "input" : "input",
  "output" : "output",
  "error_msg" : "error_msg",
  "start_time" : "2021-01-30T23:00:00Z",
  "end_time" : "2021-01-31T23:00:00Z",
  "status" : "CREATED, RUNNING, FINISHED, RETRYING, TERMINATING, TERMINATED, FAILED",
  "trigger_type" : "DEBUG, TIMER, EVENT, MANUAL"
}
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "count" : 41,
  "audit_logs" : [ {
    "instance_type" : "APP, AOP_WORKFLOW, SCRIPT, PLAYBOOK, TASK, DEBUG",
    "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "action_name" : "DisabledIp",
    "instance_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "parent_instance_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "log_level" : "DEBUG INFO WARN",
    "input" : "input",
    "output" : "output",
    "error_msg" : "error_msg",
    "start_time" : "2021-01-30T23:00:00Z",
    "end_time" : "2021-01-31T23:00:00Z",
    "status" : "CREATED, RUNNING, FINISHED, RETRYING, TERMINATING, TERMINATED, FAILED",
    "trigger_type" : "DEBUG, TIMER, EVENT, MANUAL"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

查询实例类型为APP, AOP_WORKFLOW, SCRIPT, PLAYBOOK, TASK, DEBUG，动作id为909494e3-558e-46b6-a9eb-07a8e18ca62f，动作名称为DisabledIp，实例id为909494e3-558e-46b6-a9eb-07a8e18ca62f，父实例id为909494e3-558e-46b6-a9eb-07a8e18ca62f，日志等级为DEBUG INFO WARN，输入为input，输出为output，错误信息为error_msg，开始时间2021-01-30 23：00：00，结束时间2021-01-31 23：00：00，状态为CREATED, RUNNING, FINISHED, RETRYING, TERMINATING, TERMINATED, FAILED，触发类型为DEBUG, TIMER, EVENT, MANUAL的剧本实例审计日志

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListPlaybookAuditLogsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListPlaybookAuditLogsRequest request = new ListPlaybookAuditLogsRequest();
        request.withWorkspaceId("{workspace_id}");
        AuditLogInfo body = new AuditLogInfo();
        body.withTriggerType("DEBUG, TIMER, EVENT, MANUAL");
        body.withStatus("CREATED, RUNNING, FINISHED, RETRYING, TERMINATING, TERMINATED, FAILED");
        body.withEndTime("2021-01-31T23:00:00Z");
        body.withStartTime("2021-01-30T23:00:00Z");
        body.withErrorMsg("error_msg");
        body.withOutput("output");
        body.withInput("input");
        body.withLogLevel("DEBUG INFO WARN");
        body.withParentInstanceId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        body.withInstanceId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        body.withActionName("DisabledIp");
        body.withActionId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        body.withInstanceType("APP, AOP_WORKFLOW, SCRIPT, PLAYBOOK, TASK, DEBUG");
        request.withBody(body);
        try {
            ListPlaybookAuditLogsResponse response = client.listPlaybookAuditLogs(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

查询实例类型为APP, AOP_WORKFLOW, SCRIPT, PLAYBOOK, TASK, DEBUG，动作id为909494e3-558e-46b6-a9eb-07a8e18ca62f，动作名称为DisabledIp，实例id为909494e3-558e-46b6-a9eb-07a8e18ca62f，父实例id为909494e3-558e-46b6-a9eb-07a8e18ca62f，日志等级为DEBUG INFO WARN，输入为input，输出为output，错误信息为error_msg，开始时间2021-01-30 23：00：00，结束时间2021-01-31 23：00：00，状态为CREATED, RUNNING, FINISHED, RETRYING, TERMINATING, TERMINATED, FAILED，触发类型为DEBUG, TIMER, EVENT, MANUAL的剧本实例审计日志

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListPlaybookAuditLogsRequest()
        request.workspace_id = "{workspace_id}"
        request.body = AuditLogInfo(
            trigger_type="DEBUG, TIMER, EVENT, MANUAL",
            status="CREATED, RUNNING, FINISHED, RETRYING, TERMINATING, TERMINATED, FAILED",
            end_time="2021-01-31T23:00:00Z",
            start_time="2021-01-30T23:00:00Z",
            error_msg="error_msg",
            output="output",
            input="input",
            log_level="DEBUG INFO WARN",
            parent_instance_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            instance_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            action_name="DisabledIp",
            action_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            instance_type="APP, AOP_WORKFLOW, SCRIPT, PLAYBOOK, TASK, DEBUG"
        )
        response = client.list_playbook_audit_logs(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

查询实例类型为APP, AOP_WORKFLOW, SCRIPT, PLAYBOOK, TASK, DEBUG，动作id为909494e3-558e-46b6-a9eb-07a8e18ca62f，动作名称为DisabledIp，实例id为

909494e3-558e-46b6-a9eb-07a8e18ca62f，父实例id为909494e3-558e-46b6-a9eb-07a8e18ca62f，日志等级为DEBUG INFO WARN，输入为input，输出为output，错误信息为error_msg，开始时间2021-01-30 23：00：00，结束时间2021-01-31 23：00：00，状态为CREATED, RUNNING, FINISHED, RETRYING, TERMINATING, TERMINATED, FAILED，触发类型为DEBUG, TIMER, EVENT, MANUAL的剧本实例审计日志

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPlaybookAuditLogsRequest{}
    request.WorkspaceId = "{workspace_id}"
    triggerTypeAuditLogInfo:= "DEBUG, TIMER, EVENT, MANUAL"
    statusAuditLogInfo:= "CREATED, RUNNING, FINISHED, RETRYING, TERMINATING, TERMINATED, FAILED"
    endTimeAuditLogInfo:= "2021-01-31T23:00:00Z"
    startTimeAuditLogInfo:= "2021-01-30T23:00:00Z"
    errorMsgAuditLogInfo:= "error_msg"
    outputAuditLogInfo:= "output"
    inputAuditLogInfo:= "input"
    logLevelAuditLogInfo:= "DEBUG INFO WARN"
    parentInstanceIdAuditLogInfo:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    instanceIdAuditLogInfo:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    actionNameAuditLogInfo:= "DisabledIp"
    actionIdAuditLogInfo:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    instanceTypeAuditLogInfo:= "APP, AOP_WORKFLOW, SCRIPT, PLAYBOOK, TASK, DEBUG"
    request.Body = &model.AuditLogInfo{
        TriggerType: &triggerTypeAuditLogInfo,
        Status: &statusAuditLogInfo,
        EndTime: &endTimeAuditLogInfo,
        StartTime: &startTimeAuditLogInfo,
        ErrorMsg: &errorMsgAuditLogInfo,
        Output: &outputAuditLogInfo,
        Input: &inputAuditLogInfo,
        LogLevel: &logLevelAuditLogInfo,
        ParentInstanceId: &parentInstanceIdAuditLogInfo,
        InstanceId: &instanceIdAuditLogInfo,
        ActionName: &actionNameAuditLogInfo,
        ActionId: &actionIdAuditLogInfo,
        InstanceType: &instanceTypeAuditLogInfo,
```

```
    }
    response, err := client.ListPlaybookAuditLogs(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.9 剧本审核管理

4.9.1 审核剧本

功能介绍

审核剧本

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}/approval

表 4-605 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

参数	是否必选	参数类型	描述
version_id	是	String	版本ID

请求参数

表 4-606 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-607 请求 Body 参数

参数	是否必选	参数类型	描述
result	否	String	审核结果 通过：PASS 不通过：UN_PASS
content	否	String	审核意见

响应参数

状态码： 200

表 4-608 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-609 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	响应消息

参数	参数类型	描述
data	ApproveOpinionDetail object	审核详情信息

表 4-610 ApproveOpinionDetail

参数	参数类型	描述
result	String	审核结果
content	String	审核内容

状态码： 400

表 4-611 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-612 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

审核一个剧本，审核结果为PASS，审核意见为xxxxx。

```
{
  "result" : "PASS",
  "content" : "xxxxx"
}
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "result" : "PASS",
```

```
  "content" : "need modify"
 }
}
```

SDK 代码示例

SDK代码示例如下。

Java

审核一个剧本，审核结果为PASS，审核意见为xxxxx。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class CreatePlaybookApproveSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreatePlaybookApproveRequest request = new CreatePlaybookApproveRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        ApprovePlaybookInfo body = new ApprovePlaybookInfo();
        body.withContent("xxxxx");
        body.withResult("PASS");
        request.withBody(body);
        try {
            CreatePlaybookApproveResponse response = client.createPlaybookApprove(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

审核一个剧本，审核结果为PASS，审核意见为xxxxx。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreatePlaybookApproveRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        request.body = ApprovePlaybookInfo(
            content="xxxxx",
            result="PASS"
        )
        response = client.create_playbook_approve(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

审核一个剧本，审核结果为PASS，审核意见为xxxxx。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePlaybookApproveRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    contentApprovePlaybookInfo:= "xxxxx"
    resultApprovePlaybookInfo:= "PASS"
    request.Body = &model.ApprovePlaybookInfo{
        Content: &contentApprovePlaybookInfo,
        Result: &resultApprovePlaybookInfo,
    }
    response, err := client.CreatePlaybookApprove(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.9.2 查询剧本审核结果

功能介绍

查询剧本审核结果

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/approval

表 4-613 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

表 4-614 Query 参数

参数	是否必选	参数类型	描述
resource_id	否	String	资源ID
approve_type	否	String	审核类型。（ PLAYBOOK-剧本, AOP_WORKFLOW--流程)

请求参数

表 4-615 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-616 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-617 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	响应消息
data	Array of **ApproveOpinionDetail** objects	剧本审核详情

表 4-618 ApproveOpinionDetail

参数	参数类型	描述
result	String	审核结果
content	String	审核内容

状态码： 400

表 4-619 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-620 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
```

```
  "message" : "Error message",
  "data" : [ {
    "result" : "PASS",
    "content" : "need modify"
  } ]
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListPlaybookApprovesSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListPlaybookApprovesRequest request = new ListPlaybookApprovesRequest();
        request.withWorkspaceId("{workspace_id}");
        try {
            ListPlaybookApprovesResponse response = client.listPlaybookApproves(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListPlaybookApprovesRequest()
        request.workspace_id = "{workspace_id}"
        response = client.list_playbook_approves(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

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

    request := &model.ListPlaybookApprovesRequest{}
    request.WorkspaceId = "{workspace_id}"
    response, err := client.ListPlaybookApproves(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	请求成功响应信息
400	请求失败响应信息

错误码

请参见**错误码**。

4.10 剧本动作管理

4.10.1 查询剧本动作

功能介绍

查询剧本动作列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}/actions

表 4-621 路径参数

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

参数	是否必选	参数类型	描述
workspace_id	是	String	工作空间ID
version_id	是	String	剧本版本ID

表 4-622 Query 参数

参数	是否必选	参数类型	描述
limit	是	Integer	分页查询参数，用于指定一次查询最多的结果数，从1开始
offset	是	Integer	分页查询参数。用于指定查询结果的起始位置，从0开始

请求参数

表 4-623 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-624 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-625 响应 Body 参数

参数	参数类型	描述
code	String	错误码

参数	参数类型	描述
message	String	错误信息
total	Integer	总数
size	Integer	分页大小
page	Integer	当前页数
data	Array of **ActionInfo** objects	剧本动作列表信息

表 4-626 ActionInfo

参数	参数类型	描述
id	String	剧本流程动作ID
name	String	流程动作名称
description	String	描述
action_type	String	流程动作类型
action_id	String	流程ID
playbook_id	String	剧本ID
playbook_version_id	String	剧本版本ID
project_id	String	项目ID

状态码： 400

表 4-627 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-628 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应参数

```
{
  "code" : 0,
  "message" : "Error message",
  "total" : 41,
  "size" : 3,
  "page" : 10,
  "data" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "description" : "This my XXXX",
    "action_type" : "Workflow",
    "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "playbook_id" : "string",
    "playbook_version_id" : "string",
    "project_id" : "string"
  } ]
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListPlaybookActionsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
```

```
            .build();
        ListPlaybookActionsRequest request = new ListPlaybookActionsRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        try {
            ListPlaybookActionsResponse response = client.listPlaybookActions(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListPlaybookActionsRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        response = client.list_playbook_actions(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
```

```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPlaybookActionsRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    response, err := client.ListPlaybookActions(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	请求成功响应参数
400	请求失败请求参数

错误码

请参见**错误码**。

4.10.2 创建剧本动作

功能介绍

创建剧本动作

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}/actions

表 4-629 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
version_id	是	String	剧本版本ID

请求参数

表 4-630 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-631 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of **CreateAction** objects	Create actions

表 4-632 CreateAction

参数	是否必选	参数类型	描述
name	否	String	名称
description	否	String	描述

参数	是否必选	参数类型	描述
action_type	是	String	类型，默认AOP_WORKFLOW.
action_id	是	String	剧本动作ID
sort_order	否	String	排序方式

响应参数

状态码： 200

表 4-633 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-634 响应 Body 参数

参数	参数类型	描述
code	String	Error code
message	String	Error message
data	Array of **ActionInfo** objects	list of informations of playbook action

表 4-635 ActionInfo

参数	参数类型	描述
id	String	剧本流程动作ID
name	String	流程动作名称
description	String	描述
action_type	String	流程动作类型
action_id	String	流程ID
playbook_id	String	剧本ID
playbook_version_id	String	剧本版本ID
project_id	String	项目ID

状态码： 400

表 4-636 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-637 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

创建一个剧本动作，剧本名称为MyXXX，描述为This my XXXX，剧本动作类型为aopworkflow，剧本动作ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，排序顺序为string。

```
[ {
  "name" : "MyXXX",
  "description" : "This my XXXX",
  "action_type" : "aopworkflow",
  "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "sort_order" : "string"
} ]
```

响应示例

状态码： 200

请求成功响应信息

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "description" : "This my XXXX",
    "action_type" : "Workflow",
    "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "playbook_id" : "string",
    "playbook_version_id" : "string",
    "project_id" : "string"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一个剧本动作，剧本名称为MyXXX，描述为This my XXXX，剧本动作类型为aopworkflow，剧本动作ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，排序顺序为string。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class CreatePlaybookActionSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreatePlaybookActionRequest request = new CreatePlaybookActionRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        List<CreateAction> listbodyCreateActionInfo = new ArrayList<>();
        listbodyCreateActionInfo.add(
            new CreateAction()
                .withName("MyXXX")
                .withDescription("This my XXXX")
                .withActionType("aopworkflow")
                .withActionId("909494e3-558e-46b6-a9eb-07a8e18ca62f")
                .withSortOrder("string")
        );
        request.withBody(listbodyCreateActionInfo);
        try {
            CreatePlaybookActionResponse response = client.createPlaybookAction(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

创建一个剧本动作，剧本名称为MyXXX，描述为This my XXXX，剧本动作类型为aopworkflow，剧本动作ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，排序顺序为string。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreatePlaybookActionRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        listCreateActionInfobody = [
            CreateAction(
                name="MyXXX",
                description="This my XXXX",
                action_type="aopworkflow",
                action_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
                sort_order="string"
            )
        ]
        request.body = listCreateActionInfobody
        response = client.create_playbook_action(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

创建一个剧本动作，剧本名称为MyXXX，描述为This my XXXX，剧本动作类型为aopworkflow，剧本动作ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，排序顺序为string。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePlaybookActionRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    nameCreateActionInfo:= "MyXXX"
    descriptionCreateActionInfo:= "This my XXXX"
    sortOrderCreateActionInfo:= "string"
    var listCreateActionInfobody = []model.CreateAction{
        {
            Name: &nameCreateActionInfo,
            Description: &descriptionCreateActionInfo,
            ActionType: "aopworkflow",
            ActionId: "909494e3-558e-46b6-a9eb-07a8e18ca62f",
            SortOrder: &sortOrderCreateActionInfo,
        },
    }
    request.Body = &listCreateActionInfobody
    response, err := client.CreatePlaybookAction(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	请求成功响应信息

状态码	描述
400	请求失败响应信息

错误码

请参见**错误码**。

4.10.3 删除剧本动作

功能介绍

删除剧本动作

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}/actions/{action_id}

表 4-638 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
version_id	是	String	剧本版本ID
action_id	是	String	剧本动作ID

请求参数

表 4-639 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-640 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-641 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	响应消息

状态码： 400

表 4-642 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-643 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

无

响应示例

状态码： 200

请求成功响应参数

```
{
  "code" : 0,
```

```
  "message" : "Error message"
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class DeletePlaybookActionSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        DeletePlaybookActionRequest request = new DeletePlaybookActionRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        request.withActionId("{action_id}");
        try {
            DeletePlaybookActionResponse response = client.deletePlaybookAction(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = DeletePlaybookActionRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        request.action_id = "{action_id}"
        response = client.delete_playbook_action(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
    request := &model.DeletePlaybookActionRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    request.ActionId = "{action_id}"
    response, err := client.DeletePlaybookAction(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	请求成功响应参数
400	请求失败响应参数

错误码

请参见**错误码**。

4.10.4 更新剧本动作

功能介绍

更新剧本动作

调用方法

请参见**如何调用API**。

URI

PUT /v1/{project_id}/workspaces/{workspace_id}/soc/playbooks/versions/{version_id}/actions/{action_id}

表 4-644 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID
version_id	是	String	剧本版本ID

参数	是否必选	参数类型	描述
action_id	是	String	剧本动作ID

请求参数

表 4-645 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-646 请求 Body 参数

参数	是否必选	参数类型	描述
name	否	String	名称
description	否	String	描述
action_type	否	String	类型，默认AOP_WORKFLOW.
action_id	否	String	剧本动作ID
sort_order	否	String	排序方式

响应参数

状态码： 200

表 4-647 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-648 响应 Body 参数

参数	参数类型	描述
code	String	Error code
message	String	Error message
data	**ActionInfo** object	剧本流程动作信息

表 4-649 ActionInfo

参数	参数类型	描述
id	String	剧本流程动作ID
name	String	流程动作名称
description	String	描述
action_type	String	流程动作类型
action_id	String	流程ID
playbook_id	String	剧本ID
playbook_version_id	String	剧本版本ID
project_id	String	项目ID

状态码： 400

表 4-650 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-651 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

更新一个剧本动作，剧本名称为MyXXX，描述为This my XXXX，剧本动作类型为aopworkflow，剧本动作ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，排序顺序为string。

```
{
  "name" : "MyXXX",
  "description" : "This my XXXX",
  "action_type" : "aopworkflow",
  "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "sort_order" : "string"
}
```

响应示例

状态码： 200

请求成功响应参数

```
{
  "code" : 0,
  "message" : "Error message",
  "data" : {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "MyXXX",
    "description" : "This my XXXX",
    "action_type" : "Workflow",
    "action_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "playbook_id" : "string",
    "playbook_version_id" : "string",
    "project_id" : "string"
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新一个剧本动作，剧本名称为MyXXX，描述为This my XXXX，剧本动作类型为aopworkflow，剧本动作ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，排序顺序为string。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class UpdatePlaybookActionSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        UpdatePlaybookActionRequest request = new UpdatePlaybookActionRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withVersionId("{version_id}");
        request.withActionId("{action_id}");
        ModifyActionInfo body = new ModifyActionInfo();
        body.withSortOrder("string");
        body.withActionId("909494e3-558e-46b6-a9eb-07a8e18ca62f");
        body.withActionType("aopworkflow");
        body.withDescription("This my XXXX");
        body.withName("MyXXX");
        request.withBody(body);
        try {
            UpdatePlaybookActionResponse response = client.updatePlaybookAction(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

更新一个剧本动作，剧本名称为MyXXX，描述为This my XXXX，剧本动作类型为aopworkflow，剧本动作ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，排序顺序为string。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

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

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

    try:
        request = UpdatePlaybookActionRequest()
        request.workspace_id = "{workspace_id}"
        request.version_id = "{version_id}"
        request.action_id = "{action_id}"
        request.body = ModifyActionInfo(
            sort_order="string",
            action_id="909494e3-558e-46b6-a9eb-07a8e18ca62f",
            action_type="aopworkflow",
            description="This my XXXX",
            name="MyXXX"
        )
        response = client.update_playbook_action(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

更新一个剧本动作，剧本名称为MyXXX，描述为This my XXXX，剧本动作类型为aopworkflow，剧本动作ID为909494e3-558e-46b6-a9eb-07a8e18ca62f，排序顺序为string。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdatePlaybookActionRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.VersionId = "{version_id}"
    request.ActionId = "{action_id}"
    sortOrderModifyActionInfo:= "string"
```

```
    actionIdModifyActionInfo:= "909494e3-558e-46b6-a9eb-07a8e18ca62f"
    actionTypeModifyActionInfo:= "aopworkflow"
    descriptionModifyActionInfo:= "This my XXXX"
    nameModifyActionInfo:= "MyXXX"
    request.Body = &model.ModifyActionInfo{
        SortOrder: &sortOrderModifyActionInfo,
        ActionId: &actionIdModifyActionInfo,
        ActionType: &actionTypeModifyActionInfo,
        Description: &descriptionModifyActionInfo,
        Name: &nameModifyActionInfo,
    }
    response, err := client.UpdatePlaybookAction(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	请求成功响应参数
400	请求失败响应参数

错误码

请参见**错误码**。

4.11 事件关系管理

4.11.1 查询关联 Dataobject 列表

功能介绍

查询关联Dataobject列表

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/{dataclass_type}/{data_object_id}/{related_dataclass_type}/search

表 4-652 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id
dataclass_type	是	String	关联主体dataobject所属数据类，小写复数，如告警为alerts，事件为incidents
data_object_id	是	String	关联主体dataobject的id
related_dataclass_type	是	String	被关联的dataobject所属数据类，小写复数，如告警为alerts，事件为incidents

请求参数

表 4-653 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）
content-type	是	String	内容类型

表 4-654 请求 Body 参数

参数	是否必选	参数类型	描述
limit	否	Integer	分页大小
offset	否	Integer	偏移量
sort_by	否	String	排序字段：create_time | update_time
order	否	String	排序方式：DESC | ASC
from_date	否	String	搜索开始时间，例如：2023-02-20T00:00:00.000Z
to_date	否	String	搜索结束时间，例如：2023-02-27T23:59:59.999Z

参数	是否必选	参数类型	描述
condition	否	**condition** object	搜索条件表达式

表 4-655 condition

参数	是否必选	参数类型	描述
conditions	否	Array of **conditions** objects	表达式列表
logics	否	Array of strings	表达式名称列表

表 4-656 conditions

参数	是否必选	参数类型	描述
name	否	String	表达式名称
data	否	Array of strings	表达式内容列表

响应参数

状态码： 200

表 4-657 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-658 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
total	Integer	告警总数
limit	Integer	分页大小

参数	参数类型	描述
offset	Integer	偏移量
success	Boolean	是否成功
data	Array of DataObjectDetail objects	告警列表

表 4-659 DataObjectDetail

参数	参数类型	描述
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
data_object	DataObject object	告警实体信息
dataclass_ref	dataclass_ref object	数据类对象
format_version	Integer	格式版本
id	String	事件唯一标识，UUID格式，最大36个字符
project_id	String	当前项目的id
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
version	Integer	版本
workspace_id	String	当前的工作空间id

表 4-660 DataObject

参数	参数类型	描述
version	String	告警对象的版本，该字段的值必须为云SSA服务确定的官方发布版本之一
id	String	事件唯一标识，UUID格式，最大36个字符
domain_id	String	数据投递后，被委托用户的domain_id

参数	参数类型	描述
region_id	String	数据投递后，被委托用户的region_id
workspace_id	String	当前的工作空间id
environment	**environment** object	告警产生的环境坐标信息
datasource	**datasource** object	首次上报数据源
first_observed_time	String	首次发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
last_observed_time	String	最近发现时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
create_time	String	记录时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
arrive_time	String	接收时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
title	String	告警标题
description	String	告警描述信息
source_url	String	告警URL链接，指向数据源产品中有关当前事件说明的页面
count	Integer	事件发生次数
confidence	Integer	事件的置信度。置信度的定义旨在说明识别的行为或问题的可能性。 取值范围：0-100，0表示置信度为0%，100表示置信度为100%

参数	参数类型	描述
severity	String	严重性等级，取值范围：Tips | Low | Medium | High | Fatal 说明： 0: Tips – 未发现任何问题。 1: Low – 无需针对问题执行任何操作。 2: Medium – 问题需要处理，但不紧急。 3: High – 问题必须优先处理。 4: Fatal – 问题必须立即处理，以防止产生进一步的损害
criticality	Integer	关键性，是指事件涉及的资源的重要性级别。 取值范围：0-100，0表示资源不关键，100表示最关键资源
alert_type	**alert_type** object	告警分类，详细定义参考《告警类型定义》
network_list	Array of **network_list** objects	网络信息
resource_list	Array of **resource_list** objects	受影响资源
remediation	**remediation** object	补救措施
verification_state	String	验证状态，标识事件的准确性。可选类型如下： Unknown – 未知 True_Positive – 确认 False_Positive – 误报 默认填写Unknown
handle_status	String	事件处理状态，可选类型如下： Open – 打开，默认 Block – 阻塞 Closed – 关闭 默认填写Open
sla	Integer	约束闭环时间：设置风险接受持续时间。单位：小时

参数	参数类型	描述
update_time	String	更新时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
close_time	String	关闭时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
ipdrr_phase	String	周期/处置阶段编号 Prepartion|Detection and Analysis|Containm，Eradication& Recovery|Post-Incident-Activity
simulation	String	调试字段
actor	String	告警调查员
owner	String	责任人、服务责任人
creator	String	创建人
close_reason	String	关闭原因: 误检 - False detection 已解决 - Resolved 重复 - Repeated 其他 - Other
close_comment	String	关闭评论
malware	**malware** object	恶意软件
system_info	Object	系统信息
process	Array of **process** objects	进程信息
user_info	Array of **user_info** objects	用户信息
file_info	Array of **file_info** objects	文件信息

表 4-661 environment

参数	参数类型	描述
vendor_type	String	环境供应商
domain_id	String	租户id

参数	参数类型	描述
region_id	String	区域id，全局服务global
cross_workspace_id	String	数据投递前的源工作空间id，在源空间下值为null，投递后为被委托用户的id
project_id	String	项目id， 全局服务默认null

表 4-662 datasource

参数	参数类型	描述
source_type	Integer	数据源类型，取值范围如下： 1 - 云上产品 2 - 第三方产品 3 - 租户私有产品
domain_id	String	数据源产品所属账号的id
project_id	String	数据源产品所属项目的id
region_id	String	数据源产品所在区域，具体取值范围查看云地区和终端节点定义，例如cn-north-1
company_name	String	数据源产品所属公司的名称
product_name	String	数据源产品的名称
product_feature	String	产品功能特性名称，用来指明检测到当前事件的产品的功能特性
product_module	String	检测模块列表

表 4-663 alert_type

参数	参数类型	描述
category	String	类别
alert_type	String	告警类型

表 4-664 network_list

参数	参数类型	描述
direction	String	方向，取值范围：IN | OUT

参数	参数类型	描述
protocol	String	协议，包含7层和4层的协议 参考：IANA registered name https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml
src_ip	String	源IP地址
src_port	Integer	源端口，0–65535
src_domain	String	源域名
src_geo	**src_geo** object	源IP的地理位置信息
dest_ip	String	目的IP地址
dest_port	String	目的端口，0–65535
dest_domain	String	目的域名
dest_geo	**dest_geo** object	目标IP的地理位置信息

表 4-665 src_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-666 dest_geo

参数	参数类型	描述
latitude	Number	纬度
longitude	Number	经度
city_code	String	城市编码，Beijing | Shanghai
country_code	String	国家简码，参考ISO 3166-1 alpha-2，例如：CN | US | DE | IT | SG

表 4-667 resource_list

参数	参数类型	描述
id	String	云服务资源id
name	String	资源名称
type	String	资源类型；引用云RMS type字段
provider	String	云服务名称；引用云RMS provider字段
region_id	String	区域；按照云regionId填写，如cn-north-1等
domain_id	String	资源所属账号ID，UUID格式
project_id	String	资源所属项目ID，UUID格式
ep_id	String	企业项目id
ep_name	String	企业项目名称
tags	String	资源标签 1、最多50个key/values对 2、values：最大255字符，取值范围：字母数字,空格,+, -, =, ., _, :, /,@

表 4-668 remediation

参数	参数类型	描述
recommendation	String	推荐处理方法
url	String	链接，指向该事件的一般修复信息。该URL必须可以从公网访问，不需要提供凭证

表 4-669 malware

参数	参数类型	描述
malware_family	String	恶意家族
malware_class	String	恶意软件分类

表 4-670 process

参数	参数类型	描述
process_name	String	进程名

参数	参数类型	描述
process_path	String	进程执行文件路径
process_pid	Integer	进程id
process_uid	Integer	进程用户id
process_cmdline	String	进程命令行
process_parent_name	String	父进程名称
process_parent_path	String	父进程执行文件路径
process_parent_pid	Integer	父进程id
process_parent_uid	Integer	父进程用户id
process_parent_cmdline	String	父进程命令行
process_child_name	String	子进程名称
process_child_path	String	子进程执行文件路径
process_child_pid	Integer	子进程id
process_child_uid	Integer	子进程用户id
process_child_cmdline	String	子进程命令行
process_launche_time	String	进程启动时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
process_terminate_time	String	进程结束时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区

表 4-671 user_info

参数	参数类型	描述
user_id	String	用户uid
user_name	String	用户名称

表 4-672 file_info

参数	参数类型	描述
file_path	String	文件路径/名称
file_content	String	文件内容
file_new_path	String	文件新路径/名称
file_hash	String	文件hash
file_md5	String	文件md5
file_sha256	String	文件sha256
file_attr	String	文件属性

表 4-673 dataclass_ref

参数	参数类型	描述
id	String	数据类唯一标识，UUID格式，最大36个字符
name	String	数据类名称

状态码： 400

表 4-674 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-675 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

查询数据对象的关系列表，偏移量为10，查询3条

```
{
  "limit" : 3,
```

```
  "offset" : 10
}
```

响应示例

状态码： 200

查询关联Dataobject列表返回body体

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "total" : 41,
  "limit" : 3,
  "offset" : 10,
  "data" : null
}
```

SDK 代码示例

SDK代码示例如下。

Java

查询数据对象的关系列表，偏移量为10，查询3条

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListDataobjectRelationsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListDataobjectRelationsRequest request = new ListDataobjectRelationsRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withDataclassType("{dataclass_type}");
        request.withDataObjectId("{data_object_id}");
        request.withRelatedDataclassType("{related_dataclass_type}");
        DataobjectSearch body = new DataobjectSearch();
        body.withOffset(10);
```

```
        body.withLimit(3);
        request.withBody(body);
        try {
            ListDataobjectRelationsResponse response = client.listDataobjectRelations(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

查询数据对象的关系列表，偏移量为10，查询3条

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListDataobjectRelationsRequest()
        request.workspace_id = "{workspace_id}"
        request.dataclass_type = "{dataclass_type}"
        request.data_object_id = "{data_object_id}"
        request.related_dataclass_type = "{related_dataclass_type}"
        request.body = DataobjectSearch(
            offset=10,
            limit=3
        )
        response = client.list_dataobject_relations(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

查询数据对象的关系列表，偏移量为10，查询3条

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDataobjectRelationsRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.DataclassType = "{dataclass_type}"
    request.DataObjectId = "{data_object_id}"
    request.RelatedDataclassType = "{related_dataclass_type}"
    offsetDataobjectSearch:= int32(10)
    limitDataobjectSearch:= int32(3)
    request.Body = &model.DataobjectSearch{
        Offset: &offsetDataobjectSearch,
        Limit: &limitDataobjectSearch,
    }
    response, err := client.ListDataobjectRelations(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	查询关联Dataobject列表返回body体

状态码	描述
400	查询关联Dataobject列表错误返回body体

错误码

请参见**错误码**。

4.11.2 关联 Dataobject

功能介绍

关联Dataobject

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/soc/{dataclass_type}/{data_object_id}/{related_dataclass_type}

表 4-676 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id
dataclass_type	是	String	关联主体dataobject所属数据类，小写复数，如告警为alerts，事件为incidents
data_object_id	是	String	关联主体dataobject的id
related_dataclass_type	是	String	被关联的dataobject所属数据类，小写复数，如告警为alerts，事件为incidents

请求参数

表 4-677 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）
content-type	是	String	内容类型

表 4-678 请求 Body 参数

参数	是否必选	参数类型	描述
ids	否	Array of strings	关联dataobject的ID列表

响应参数

状态码： 200

表 4-679 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-680 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
request_id	String	请求ID
total	Integer	总数
limit	Integer	分页大小
offset	Integer	偏移量
success	Boolean	是否成功

参数	参数类型	描述
data	BatchOperateDataobjectResult object	批量操作告警返回对象

表 4-681 BatchOperateDataobjectResult

参数	参数类型	描述
error_ids	Array of strings	失败id
success_ids	Array of strings	成功id

状态码： 400

表 4-682 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-683 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

创建一条ID为f60bf0e7-73b8-4832-8fc4-8c2a12830552的事件关系。

```
{
  "ids" : [ "f60bf0e7-73b8-4832-8fc4-8c2a12830552" ]
}
```

响应示例

状态码： 200

关联Dataobject返回body体

```
{
  "code" : 0,
  "message" : "Error message",
  "request_id" : "Error message",
  "success" : false,
```

```
  "total" : 41,
  "limit" : 3,
  "offset" : 10,
  "data" : {
    "success_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ],
    "error_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ]
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建一条ID为f60bf0e7-73b8-4832-8fc4-8c2a12830552的事件关系。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class CreateDataobjectRelationsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateDataobjectRelationsRequest request = new CreateDataobjectRelationsRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withDataclassType("{dataclass_type}");
        request.withDataObjectId("{data_object_id}");
        request.withRelatedDataclassType("{related_dataclass_type}");
        CreateDataobjectRelationsRequestBody body = new CreateDataobjectRelationsRequestBody();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("f60bf0e7-73b8-4832-8fc4-8c2a12830552");
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            CreateDataobjectRelationsResponse response = client.createDataobjectRelations(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

创建一条ID为f60bf0e7-73b8-4832-8fc4-8c2a12830552的事件关系。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateDataobjectRelationsRequest()
        request.workspace_id = "{workspace_id}"
        request.dataclass_type = "{dataclass_type}"
        request.data_object_id = "{data_object_id}"
        request.related_dataclass_type = "{related_dataclass_type}"
        listIdsbody = [
            "f60bf0e7-73b8-4832-8fc4-8c2a12830552"
        ]
        request.body = CreateDataobjectRelationsRequestBody(
            ids=listIdsbody
        )
        response = client.create_dataobject_relations(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

创建一条ID为f60bf0e7-73b8-4832-8fc4-8c2a12830552的事件关系。

```
package main

import (
```

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDataobjectRelationsRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.DataclassType = "{dataclass_type}"
    request.DataObjectId = "{data_object_id}"
    request.RelatedDataclassType = "{related_dataclass_type}"
    var listIdsbody = []string{
        "f60bf0e7-73b8-4832-8fc4-8c2a12830552",
    }
    request.Body = &model.CreateDataobjectRelationsRequestBody{
        Ids: &listIdsbody,
    }
    response, err := client.CreateDataobjectRelations(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	关联Dataobject返回body体
400	关联Dataobject错误返回body体

错误码

请参见**错误码**。

4.11.3 取消关联 Dataobject

功能介绍

取消关联Dataobject

调用方法

请参见**如何调用API**。

URI

DELETE /v1/{project_id}/workspaces/{workspace_id}/soc/{dataclass_type}/{data_object_id}/{related_dataclass_type}

表 4-684 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id
dataclass_type	是	String	关联主体dataobject所属数据类，小写复数，如告警为alerts，事件为incidents
data_object_id	是	String	关联主体dataobject的id
related_dataclass_type	是	String	被关联的dataobject所属数据类，小写复数，如告警为alerts，事件为incidents

请求参数

表 4-685 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）
content-type	是	String	内容类型

表 4-686 请求 Body 参数

参数	是否必选	参数类型	描述
ids	否	Array of strings	关联dataobject的ID列表

响应参数

状态码： 200

表 4-687 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-688 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误信息
data	**BatchOperateDataobjectResult** object	批量操作告警返回对象

表 4-689 BatchOperateDataobjectResult

参数	参数类型	描述
error_ids	Array of strings	失败id
success_ids	Array of strings	成功id

状态码： 400

表 4-690 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-691 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

删除一条ID为f60bf0e7-73b8-4832-8fc4-8c2a12830552的事件关系。

```
{
  "ids" : [ "f60bf0e7-73b8-4832-8fc4-8c2a12830552" ]
}
```

响应示例

状态码： 200

取消关联Dataobject返回body体

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "request_id" : "Error message",
  "success" : false,
  "total" : 41,
  "limit" : 3,
  "offset" : 10,
  "data" : {
    "success_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ],
    "error_ids" : [ "909494e3-558e-46b6-a9eb-07a8e18ca62f" ]
  }
}
```

SDK 代码示例

SDK代码示例如下。

Java

删除一条ID为f60bf0e7-73b8-4832-8fc4-8c2a12830552的事件关系。

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class DeleteDataobjectRelationsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        DeleteDataobjectRelationsRequest request = new DeleteDataobjectRelationsRequest();
        request.withWorkspaceId("{workspace_id}");
        request.withDataclassType("{dataclass_type}");
        request.withDataObjectId("{data_object_id}");
        request.withRelatedDataclassType("{related_dataclass_type}");
        CreateDataobjectRelationsRequestBody body = new CreateDataobjectRelationsRequestBody();
        List<String> listbodyIds = new ArrayList<>();
        listbodyIds.add("f60bf0e7-73b8-4832-8fc4-8c2a12830552");
        body.withIds(listbodyIds);
        request.withBody(body);
        try {
            DeleteDataobjectRelationsResponse response = client.deleteDataobjectRelations(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

删除一条ID为f60bf0e7-73b8-4832-8fc4-8c2a12830552的事件关系。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

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



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

    try:
        request = DeleteDataobjectRelationsRequest()
        request.workspace_id = "{workspace_id}"
        request.dataclass_type = "{dataclass_type}"
        request.data_object_id = "{data_object_id}"
        request.related_dataclass_type = "{related_dataclass_type}"
        listIdsbody = [
            "f60bf0e7-73b8-4832-8fc4-8c2a12830552"
        ]
        request.body = CreateDataobjectRelationsRequestBody(
            ids=listIdsbody
        )
        response = client.delete_dataobject_relations(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

删除一条ID为f60bf0e7-73b8-4832-8fc4-8c2a12830552的事件关系。

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDataobjectRelationsRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.DataclassType = "{dataclass_type}"
    request.DataObjectId = "{data_object_id}"
    request.RelatedDataclassType = "{related_dataclass_type}"
    var listIdsbody = []string{
        "f60bf0e7-73b8-4832-8fc4-8c2a12830552",
    }
```

```
    request.Body = &model.CreateDataobjectRelationsRequestBody{
        Ids: &listIdsbody,
    }
    response, err := client.DeleteDataobjectRelations(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	取消关联Dataobject返回body体
400	取消关联Dataobject错误返回body体

错误码

请参见**错误码**。

4.12 数据类管理

4.12.1 查询数据类列表

功能介绍

查询数据类列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/dataclasses

表 4-692 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

表 4-693 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	偏移量
limit	否	Integer	数据量
name	否	String	名称查询
business_code	否	String	业务编码
description	否	String	描述
is_built_in	否	Boolean	是否内置

请求参数

表 4-694 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）
content-type	是	String	内容类型

响应参数

状态码： 200

表 4-695 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-696 响应 Body 参数

参数	参数类型	描述
dataclass_details	Array of **DataClassResponseBody** objects	数据类详情
total	Number	数据总量

表 4-697 DataClassResponseBody

参数	参数类型	描述
id	String	数据类ID
create_time	String	创建时间
update_time	String	更新时间
creator_id	String	创建者ID
creator_name	String	创建者名称
modifier_id	String	修改者ID
modifier_name	String	修改这名称
cloud_pack_version	String	订阅包版本
region_id	String	区域ID
project_id	String	租户ID
workspace_id	String	工作空间ID
domain_id	String	domain id
name	String	数据类名称
business_code	String	数据类业务编码
description	String	数据类描述
is_built_in	Boolean	是否内置，true内置，false非内置
parent_id	String	父级id
type_num	Number	子类型数量

状态码： 400

表 4-698 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-699 响应 Body 参数

参数	参数类型	描述
code	String	错误码

参数	参数类型	描述
message	String	错误描述

请求示例

查询数据类列表，偏移量为10，查询3条

```
{
  "limit" : 3,
  "offset" : 10
}
```

响应示例

状态码： 200

请求成功

```
{
  "total" : 41,
  "dataclass_details" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "create_time" : "2021-01-30T23:00:00Z+0800",
    "update_time" : "2021-01-30T23:00:00Z+0800",
    "creator_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "creator_name" : "张三",
    "modifier_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "modifier_name" : "李四",
    "cloud_pack_version" : "订阅包版本",
    "region_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "domain_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "证据",
    "business_code" : "Evidence",
    "description" : "我的数据类描述",
    "is_built_in" : false,
    "parent_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "type_num" : 9
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

查询数据类列表，偏移量为10，查询3条

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;
```

```
public class ListDataclassSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListDataclassRequest request = new ListDataclassRequest();
        request.withWorkspaceId("{workspace_id}");
        try {
            ListDataclassResponse response = client.listDataclass(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

查询数据类列表，偏移量为10，查询3条

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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



```
        .build()

    try:
        request = ListDataclassRequest()
        request.workspace_id = "{workspace_id}"
        response = client.list_dataclass(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

查询数据类列表，偏移量为10，查询3条

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDataclassRequest{}
    request.WorkspaceId = "{workspace_id}"
    response, err := client.ListDataclass(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	查询数据类列表错误返回body体

错误码

请参见**错误码**。

4.12.2 查询字段列表

功能介绍

查询字段列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/dataclasses/{dataclass_id}/fields

表 4-700 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id
dataclass_id	是	String	数据类id

表 4-701 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	偏移量
limit	否	Integer	数据量
name	否	String	名称查询
is_built_in	否	Boolean	是否内置
field_category	否	String	字段分类

参数	是否必选	参数类型	描述
mapping	否	Boolean	是否展示在分类映射外的其他地方

请求参数

表 4-702 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）
content-type	是	String	内容类型

响应参数

状态码： 200

表 4-703 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-704 响应 Body 参数

参数	参数类型	描述
field_details	Array of **FieldResponseBody** objects	list of informations of field
total	Number	数据总量

表 4-705 FieldResponseBody

参数	参数类型	描述
id	String	Id value

参数	参数类型	描述
cloud_pack_version	String	订阅包版本
business_id	String	关联业务id
business_type	String	关联业务
dataclass_name	String	数据类名称
business_code	String	字段业务编码
field_key	String	字段key
name	String	字段名称
description	String	字段描述
default_value	String	默认值
display_type	String	显示类型
field_type	String	字段类型，如shorttext,radio,grid等
extra_json	String	附加json
field_tooltip	String	工具提示
iu_type	String	输入输出类型
used_by	String	使用业务
json_schema	String	json模式
is_built_in	Boolean	是否内置，true内置，false非内置
case_sensitive	Boolean	大小写敏感，true敏感，false不敏感
read_only	Boolean	只读模式，true只读，false非只读
required	Boolean	是否必填，true必填，false非必填
searchable	Boolean	可搜索，true可搜索，false非可搜索
visible	Boolean	可见，true可见，false非可见
maintainable	Boolean	可维护，true可维护，false非可维护
editable	Boolean	可编辑，true可编辑，false非可编辑
creatable	Boolean	可创建，true可创建，false非可创建
mapping	Boolean	是否展示在分类映射外的其他地方
target_api	String	目标api
creator_id	String	Creator id value
creator_name	String	Creator name value

参数	参数类型	描述
modifier_id	String	Modifier id value
modifier_name	String	Modifier name value
create_time	String	Create time
update_time	String	Update time

状态码： 400

表 4-706 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-707 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

查询字段列表，偏移量为10，查询3条

```
{
  "limit" : 3,
  "offset" : 10
}
```

响应示例

状态码： 200

请求成功

```
{
  "total" : 41,
  "field_details" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "cloud_pack_version" : "订阅包版本",
    "business_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "business_type" : "业务类型",
    "dataclass_name" : "业务id",
    "business_code" : "My Field",
    "field_key" : "字段key",
    "name" : "字段名称",
    "description" : "字段描述",
```

```
  "default_value" : "默认值",
  "display_type" : "显示类型",
  "field_type" : "shorttext",
  "extra_json" : "{}",
  "field_tooltip" : "工具提示",
  "iu_type" : "输入输出类型",
  "used_by" : "使用业务",
  "json_schema" : "{}",
  "is_built_in" : false,
  "case_sensitive" : false,
  "read_only" : false,
  "required" : false,
  "searchable" : false,
  "visible" : false,
  "maintainable" : false,
  "editable" : false,
  "creatable" : false,
  "mapping" : true,
  "target_api" : "目标api",
  "creator_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "creator_name" : "张三",
  "modifier_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "modifier_name" : "李四",
  "create_time" : "2021-01-30T23:00:00Z+0800",
  "update_time" : "2021-01-30T23:00:00Z+0800"
 } ]
}
```


状态码

状态码	描述
200	请求成功
400	查询数据类列表错误返回body体

错误码

请参见**错误码**。

4.13 流程管理

4.13.1 查询流程列表

功能介绍

查询流程列表

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces/{workspace_id}/soc/workflows

表 4-708 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

表 4-709 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	偏移量
limit	否	Integer	数据量
order	否	String	排序顺序，asc：升序，desc：降序
sortby	否	String	排序字段，create_time：创建时间，category：类型分类名称
enabled	否	Boolean	是否启用
last_version	否	Boolean	最新版本号
name	否	String	流程名称
description	否	String	流程描述
dataclass_id	否	String	数据类ID
dataclass_name	否	String	数据类名称
aop_type	否	String	流程类型

请求参数

表 4-710 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）
content-type	是	String	内容类型

响应参数

状态码： 200

表 4-711 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-712 响应 Body 参数

参数	参数类型	描述
code	String	返回码
total	Integer	数据总条数
offset	Integer	当前页大小
limit	Integer	当前页码
message	String	请求ID
success	Boolean	是否成功
data	Array of **AopWorkflowInfo** objects	流程信息列表

表 4-713 AopWorkflowInfo

参数	参数类型	描述
id	String	流程ID
name	String	流程名称
description	String	描述
project_id	String	租户ID
owner_id	String	所有者ID
creator_id	String	创建者ID
edit_role	String	编辑角色
use_role	String	是用角色
approve_role	String	审核人
enabled	Boolean	是否已启用

参数	参数类型	描述
workspace_id	String	工作空间ID
version_id	String	流程版本ID
current_approval_version_id	String	当前待审核版本号
current_rejected_versoin_id	String	当前拒绝的版本号
aop_type	String	aop的类型有以下的值 NORMAL, 通用 SURVEY, 调查 HEMOSTASIS,止血 EASE;缓解
engine_type	String	引擎的类型分为共享版和专项版
dataclass_id	String	数据类的ID

状态码： 400

表 4-714 响应 Header 参数

参数	参数类型	描述
X-request-id	String	请求ID,格式为：request_uuid-timestamp-hostname

表 4-715 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

查询流程列表，偏移量为10，查询3条

```
{
  "limit" : 3,
  "offset" : 10
}
```

响应示例

状态码： 200

请求成功

```
{
  "code" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
  "message" : "Error message",
  "total" : 41,
  "limit" : 2,
  "offset" : 1,
  "success" : true,
  "data" : [ {
    "id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "name" : "流程名称",
    "description" : "描述",
    "project_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "owner_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "creator_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "edit_role" : "编辑者",
    "use_role" : "使用者",
    "approve_role" : "审批者",
    "enabled" : true,
    "workspace_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "version_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f",
    "current_approval_version_id" : "v2",
    "current_rejected_versoin_id" : "v1",
    "aop_type" : "EASE;缓解",
    "engine_type" : "public_engine",
    "dataclass_id" : "909494e3-558e-46b6-a9eb-07a8e18ca62f"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

查询流程列表，偏移量为10，查询3条

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListWorkflowsSolution {

    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);

    SecMasterClient client = SecMasterClient.newBuilder()
            .withCredential(auth)
            .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
            .build();
    ListWorkflowsRequest request = new ListWorkflowsRequest();
    request.withWorkspaceId("{workspace_id}");
    try {
        ListWorkflowsResponse response = client.listWorkflows(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

查询流程列表，偏移量为10，查询3条

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListWorkflowsRequest()
        request.workspace_id = "{workspace_id}"
        response = client.list_workflows(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

查询流程列表，偏移量为10，查询3条

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListWorkflowsRequest{}
    request.WorkspaceId = "{workspace_id}"
    response, err := client.ListWorkflows(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	查询数据类列表错误返回body体

错误码

请参见**错误码**。

4.14 数据空间管理

4.14.1 创建数据空间

功能介绍

创建数据空间

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/siem/dataspaces

表 4-716 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

请求参数

表 4-717 请求 Body 参数

参数	是否必选	参数类型	描述
dataspace_name	是	String	数据空间名称
description	是	String	描述

响应参数

状态码： 200

表 4-718 响应 Body 参数

参数	参数类型	描述
domain_id	String	账号ID
region_id	String	region ID

参数	参数类型	描述
project_id	String	项目ID
dataspace_id	String	工作空间ID
dataspace_name	String	工作空间名称
dataspace_type	String	数据空间类型；可选值：system-defined(系统预定义)、user-defined(用户自定义)
description	String	描述
create_by	String	创建者
create_time	Long	创建时间
update_by	String	更新者
update_time	Long	更新时间

请求示例

```
{
  "dataspace_name" : "dataspace-01",
  "description" : "test dataspace"
}
```

响应示例

状态码： 200

创建成功返回值

```
{
  "domain_id" : "0531ed520xxxxxxebedb6e57xxxxxxxx",
  "region_id" : "cn-north-1",
  "project_id" : "2b31ed520xxxxxxebedb6e57xxxxxxxx",
  "dataspace_id" : "a00106ba-bede-453c-8488-b60c70bd6aed",
  "dataspace_name" : "dataspace-01",
  "dataspace_type" : "system-defined",
  "description" : "test dataspace",
  "create_by" : "0642ed520xxxxxxebedb6e57xxxxxxxx",
  "create_time" : 1584883694354,
  "update_by" : "0642ed520xxxxxxebedb6e57xxxxxxxx",
  "update_time" : 1584883694354
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class CreateDataspaceSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateDataspaceRequest request = new CreateDataspaceRequest();
        request.withWorkspaceId("{workspace_id}");
        CreateDataspaceRequestBody body = new CreateDataspaceRequestBody();
        request.withBody(body);
        try {
            CreateDataspaceResponse response = client.createDataspace(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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



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

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

    try:
        request = CreateDataspaceRequest()
        request.workspace_id = "{workspace_id}"
        request.body = CreateDataspaceRequestBody(
        )
        response = client.create_dataspace(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDataspaceRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.Body = &model.CreateDataspaceRequestBody{
    }
    response, err := client.CreateDataspace(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	创建成功返回值

错误码

请参见**错误码**。

4.15 管道管理

4.15.1 创建数据管道

功能介绍

创建数据管道

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/siem/pipes

表 4-719 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

请求参数

表 4-720 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 4-721 请求 Body 参数

参数	是否必选	参数类型	描述
dataspace_id	是	String	工作空间ID
pipe_name	是	String	数据管道名称
description	否	String	描述
storage_period	是	Integer	数据的保存时间，单位为天；默认30天，取值范围为1~3600
shards	是	Integer	数据管道分区个数；默认创建1个，最大支持创建64个分区
timestamp_field	否	String	时间戳字段
mapping	否	Map<String,**KeyIndex**>	索引字段映射；每个key对象承载一个字段的信息；存在多个key对象，key可变，表示字段名称；可嵌套

表 4-722 KeyIndex

参数	是否必选	参数类型	描述
type	否	String	字段类型；text 全文索引字段、keyword 结构化字段、long Long、integer Integer、double Double、float Float、date 时间字段
is_chinese_exist	否	Boolean	是否包含中文
properties	否	Map<String,**KeyIndex**>	嵌套结构

响应参数

状态码： 200

表 4-723 响应 Body 参数

参数	参数类型	描述
domain_id	String	用户domainId
project_id	String	项目id
dataspace_id	String	数据空间id
dataspace_name	String	数据空间名称
pipe_id	String	管道id
pipe_name	String	管道名称
pipe_type	String	管道类型（system-defined，系统预定义)、1（user-defined，用户自定义)
description	String	描述信息
storage_period	Integer	索引存储天数
shards	Integer	索引分片数量
create_by	String	创建者
create_time	Integer	创建时间
update_by	String	更新者
update_time	Integer	更新时间

状态码： 400

表 4-724 响应 Body 参数

参数	参数类型	描述
error_msg	String	无效请求提示信息
error_code	String	错误码

状态码： 401

表 4-725 响应 Body 参数

参数	参数类型	描述
error_msg	String	权限错误
error_code	String	错误码

状态码： 403

表 4-726 响应 Body 参数

参数	参数类型	描述
-	String	

状态码： 500

表 4-727 响应 Body 参数

参数	参数类型	描述
error_msg	String	系统内部错误
error_code	String	错误码

请求示例

```
{
  "dataspace_id" : "a00106ba-bede-453c-8488-b60c70bd6aed",
  "pipe_name" : "pipe-01",
  "description" : "test pipe",
  "storage_period" : 30,
  "shards" : 3,
  "mapping" : {
    "name" : {
      "type" : "text"
    },
    "id" : {
      "type" : "text"
    },
    "publish_time" : {
      "type" : "data"
    }
  }
}
```

响应示例

状态码： 200

创建成功返回值

```
{
  "domain_id" : "0531ed520xxxxxxebedb6e57xxxxxxxx",
```

```
 "project_id" : "2b31ed520xxxxxxebedb6e57xxxxxxxx",
 "dataspace_id" : "a00106ba-bede-453c-8488-b60c70bd6aed",
 "dataspace_name" : "dataspace-01",
 "pipe_id" : "b22106ba-bede-453c-8488-b60c70bd6aed",
 "pipe_name" : "pipe-01",
 "pipe_type" : "system-defined",
 "description" : "test pipe",
 "storage_period" : 30,
 "shards" : 3,
 "create_by" : "0642ed520xxxxxxebedb6e57xxxxxxxx",
 "create_time" : 1584883694354,
 "update_by" : "0642ed520xxxxxxebedb6e57xxxxxxxx",
 "update_time" : 1584883694354
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class CreatePipeSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreatePipeRequest request = new CreatePipeRequest();
        request.withWorkspaceId("{workspace_id}");
        CreatePipeRequestBody body = new CreatePipeRequestBody();
        request.withBody(body);
        try {
            CreatePipeResponse response = client.createPipe(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreatePipeRequest()
        request.workspace_id = "{workspace_id}"
        request.body = CreatePipeRequestBody(
        )
        response = client.create_pipe(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePipeRequest{}
    request.WorkspaceId = "{workspace_id}"
    request.Body = &model.CreatePipeRequestBody{
    }
    response, err := client.CreatePipe(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	创建成功返回值
400	错误请求
401	认证失败
403	访问拒绝
500	系统内部错误

错误码

请参见**错误码**。

4.16 工作空间管理

4.16.1 新建工作空间

功能介绍

在使用安全云脑的基线检查、告警管理、安全分析、安全编排等功能前，需要创建工作空间，它可以将资源划分为各个不同的工作场景，避免资源冗余查找不便，影响日常使用。

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces

表 4-728 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 4-729 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）。
content-type	是	String	application/json;charset=UTF-8

表 4-730 请求 Body 参数

参数	是否必选	参数类型	描述
region_id	是	String	区域id
enterprise_project_id	否	String	企业项目id
enterprise_project_name	否	String	企业项目名称
view_bind_id	否	String	视图绑定的空间id
is_view	否	Boolean	是否是视图
name	是	String	工作空间名称
description	否	String	工作空间描述
project_name	是	String	项目名称

参数	是否必选	参数类型	描述
tags	否	Array of TagsPojo objects	通过给账号下的资源添加标签，可以对资源进行自定义标记，实现资源的分类。可到标签管理服务使用可视化表格操作资源标签，并对标签进行批量编辑。

表 4-731 TagsPojo

参数	是否必选	参数类型	描述
key	否	String	标签key
value	否	String	标签value

响应参数

状态码： 200

表 4-732 响应 Body 参数

参数	参数类型	描述
id	String	工作空间id
create_time	String	创建时间
update_time	String	更新时间
name	String	工作空间名称
description	String	工作空间描述
creator_id	String	创建人id
creator_name	String	创建人名称
modifier_id	String	修改人id
modifier_name	String	修改人名称
project_id	String	所属项目id
project_name	String	所属项目名称
domain_id	String	所属租户id
domain_name	String	所属租户名称
enterprise_project_id	String	企业项目id

参数	参数类型	描述
enterprise_project_name	String	企业项目名称
is_view	Boolean	是否是视图
region_id	String	区域id
view_bind_id	String	视图绑定的空间id
view_bind_name	String	视图绑定的空间名称
workspace_agency_list	Array of **workspace_agency_list** objects	仅用于视图场景，列出了该视图纳管的空间列表

表 4-733 workspace_agency_list

参数	参数类型	描述
project_id	String	委托空间所属项目id
id	String	空间委托id
name	String	空间委托名称
region_id	String	委托空间所属region id
workspace_attribution	String	THIS_ACCOUNT:本账号空间,CROSS_ACCOUNT:跨账号空间
agency_version	String	用户创建托管空间时使用的IAM委托版本，V3或者V5
domain_id	String	委托租户id
domain_name	String	委托租户名称
iam_agency_id	String	iam委托id
iam_agency_name	String	iam委托名称
resource_spec_code	Array of strings	委托空间购买版本
selected	Boolean	是否被视图选中

状态码： 400

表 4-734 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误消息

状态码： 500

表 4-735 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误消息

请求示例

创建工作空间请求体

```
{
  "name" : "我的工作空间",
  "region_id" : "cn-north-4",
  "project_name" : "cn-north-4",
  "enterprise_project_id" : "",
  "enterprise_project_name" : "",
  "tags" : [ {
    "key" : "tag1",
    "value" : "value1"
  } ],
  "description" : "我的工作空间"
}
```

响应示例

状态码： 200

请求成功

```
{
  "create_time" : "2024-07-02T09:25:17Z+0800",
  "creator_id" : "b4****************************46a",
  "creator_name" : "l00644738",
  "description" : "我的工作空间",
  "domain_id" : "ac****************************bf4",
  "domain_name" : "scc****09",
  "enterprise_project_id" : "",
  "enterprise_project_name" : "",
  "id" : "39************bf",
  "is_view" : false,
  "modifier_id" : "",
  "modifier_name" : "",
  "name" : "我的工作空间",
  "project_id" : "15************************da6",
  "project_name" : "cn-north-4",
  "region_id" : "cn-north-4",
  "update_time" : "2024-07-02T09:25:17Z+0800",
```

```
  "view_bind_id" : "",
  "view_bind_name" : "",
  "workspace_agency_list" : [ ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建工作空间请求体

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class CreateWorkspaceSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateWorkspaceRequest request = new CreateWorkspaceRequest();
        CreateWorkspaceRequestBody body = new CreateWorkspaceRequestBody();
        List<TagsPojo> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new TagsPojo()
                .withKey("tag1")
                .withValue("value1")
        );
        body.withTags(listbodyTags);
        body.withProjectName("cn-north-4");
        body.withDescription("我的工作空间");
        body.withName("我的工作空间");
        body.withEnterpriseProjectName("");
        body.withEnterpriseProjectId("");
        body.withRegionId("cn-north-4");
        request.withBody(body);
        try {
            CreateWorkspaceResponse response = client.createWorkspace(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

创建工作空间请求体

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreateWorkspaceRequest()
        listTagsbody = [
            TagsPojo(
                key="tag1",
                value="value1"
            )
        ]
        request.body = CreateWorkspaceRequestBody(
            tags=listTagsbody,
            project_name="cn-north-4",
            description="我的工作空间",
            name="我的工作空间",
            enterprise_project_name="",
            enterprise_project_id="",
            region_id="cn-north-4"
        )
        response = client.create_workspace(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

创建工作空间请求体

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateWorkspaceRequest{}
    keyTags:= "tag1"
    valueTags:= "value1"
    var listTagsbody = []model.TagsPojo{
        {
            Key: &keyTags,
            Value: &valueTags,
        },
    }
    descriptionCreateWorkspaceRequestBody:= "我的工作空间"
    enterpriseProjectNameCreateWorkspaceRequestBody:= ""
    enterpriseProjectIdCreateWorkspaceRequestBody:= ""
    request.Body = &model.CreateWorkspaceRequestBody{
        Tags: &listTagsbody,
        ProjectName: "cn-north-4",
        Description: &descriptionCreateWorkspaceRequestBody,
        Name: "我的工作空间",
        EnterpriseProjectName: &enterpriseProjectNameCreateWorkspaceRequestBody,
        EnterpriseProjectId: &enterpriseProjectIdCreateWorkspaceRequestBody,
        RegionId: "cn-north-4",
    }
    response, err := client.CreateWorkspace(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	请求参数有误
500	请求失败

错误码

请参见**错误码**。

4.16.2 工作空间列表查询

功能介绍

工作空间列表查询:可通过工作空间名称、工作空间描述、创建时间等条件对租户的工作空间进行筛选。

调用方法

请参见**如何调用API**。

URI

GET /v1/{project_id}/workspaces

表 4-736 路径参数

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

表 4-737 Query 参数

参数	是否必选	参数类型	描述
offset	是	Number	偏移量 指定返回记录的开始位置，必须为数字，取值范围为大于或等于0，
limit	是	Number	每页显示个数

参数	是否必选	参数类型	描述
region_id	否	String	区域id
name	否	String	名称查询
description	否	String	描述查询
view_bind_id	否	String	视图绑定的空间id
view_bind_name	否	String	视图绑定的空间名称
create_time_start	否	String	创建时间开始，例如2024-04-26T16:08:09Z+0800
create_time_end	否	String	创建时间结束，例如2024-04-2T16:08:09Z+0800
is_view	否	Boolean	是否查询视图 true or false
ids	否	String	工作空间id数组，英文逗号分隔
normal_project_id	否	String	普通项目的项目id
enterprise_project_id	否	String	企业项目的项目id

请求参数

表 4-738 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）。
content-type	是	String	application/json;charset=UTF-8

响应参数

状态码： 200

表 4-739 响应 Body 参数

参数	参数类型	描述
workspaces	Array of **CreateWorkspaceResponseBody** objects	空间信息
count	Number	数据总量

表 4-740 CreateWorkspaceResponseBody

参数	参数类型	描述
id	String	工作空间id
create_time	String	创建时间
update_time	String	更新时间
name	String	工作空间名称
description	String	工作空间描述
creator_id	String	创建人id
creator_name	String	创建人名称
modifier_id	String	修改人id
modifier_name	String	修改人名称
project_id	String	所属项目id
project_name	String	所属项目名称
domain_id	String	所属租户id
domain_name	String	所属租户名称
enterprise_project_id	String	企业项目id
enterprise_project_name	String	企业项目名称
is_view	Boolean	是否是视图
region_id	String	区域id
view_bind_id	String	视图绑定的空间id
view_bind_name	String	视图绑定的空间名称
workspace_agency_list	Array of **workspace_agency_list** objects	仅用于视图场景，列出了该视图纳管的空间列表

表 4-741 workspace_agency_list

参数	参数类型	描述
project_id	String	委托空间所属项目id
id	String	空间委托id
name	String	空间委托名称
region_id	String	委托空间所属region id
workspace_attribution	String	THIS_ACCOUNT:本账号空间,CROSS_ACCOUNT:跨账号空间
agency_version	String	用户创建托管空间时使用的IAM委托版本，V3或者V5
domain_id	String	委托租户id
domain_name	String	委托租户名称
iam_agency_id	String	iam委托id
iam_agency_name	String	iam委托名称
resource_spec_code	Array of strings	委托空间购买版本
selected	Boolean	是否被视图选中

状态码： 400

表 4-742 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误消息

状态码： 500

表 4-743 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误消息

请求示例

无

响应示例

状态码： 200

请求成功

```
{
  "count" : 1,
  "workspaces" : [ {
    "create_time" : "2024-07-02T09:25:17Z+0800",
    "creator_id" : "b4****************************46a",
    "creator_name" : "l00644738",
    "description" : "我的工作空间",
    "domain_id" : "ac****************************bf4",
    "domain_name" : "scc****09",
    "enterprise_project_id" : "",
    "enterprise_project_name" : "",
    "id" : "39************bf",
    "is_view" : false,
    "modifier_id" : "",
    "modifier_name" : "",
    "name" : "我的工作空间",
    "project_id" : "15************************da6",
    "project_name" : "cn-north-4",
    "region_id" : "cn-north-4",
    "update_time" : "2024-07-02T09:25:17Z+0800",
    "view_bind_id" : "",
    "view_bind_name" : "",
    "workspace_agency_list" : [ ]
  } ]
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class ListWorkspacesSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        ListWorkspacesRequest request = new ListWorkspacesRequest();
        try {
            ListWorkspacesResponse response = client.listWorkspaces(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = ListWorkspacesRequest()
        response = client.list_workspaces(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"
    secmaster "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/secmaster/v2/region"
)

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListWorkspacesRequest{}
    response, err := client.ListWorkspaces(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	请求参数有误
500	请求失败

错误码

请参见**错误码**。

4.17 计量计费管理

4.17.1 安全云脑按需订购

功能介绍

开通安全云脑按需服务

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/subscriptions/orders

表 4-744 路径参数

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

请求参数

表 4-745 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）。
X-Language	是	String	用户当前语言环境 zh-cn or en-us.

表 4-746 请求 Body 参数

参数	是否必选	参数类型	描述
region_id	是	String	区域ID，如cn-north-4
domain_id	是	String	domainId
tag_list	否	Array of **TagInfo** objects	计费标签
product_list	否	Array of **ProductPostPaid** objects	商品列表

参数	是否必选	参数类型	描述
operate_type	否	String	操作类型：create新购, addition增加配额

表 4-747 TagInfo

参数	是否必选	参数类型	描述
key	是	String	标识 中文、字母、数字、_或者-，且长度范围[2, 36]
value	是	String	内容 中文、字母、数字、_或者-，且长度范围[2, 36]

表 4-748 ProductPostPaid

参数	是否必选	参数类型	描述
id	是	String	ID标识，同一次询价中不能重复，用于标识返回询价结果和请求的映射关系
product_id	是	String	产品Id，通过向CBC询价获取该商品的标识
cloud_service_type	是	String	云服务类型，固定值为hws.service.type.sa
resource_type	是	String	用户购买云服务产品的资源类型，例如SecMaster中的典型场景配置，资源类型为hws.resource.type.secmaster.typical
resource_spec_code	是	String	用户购买云服务产品的资源规格，例如安全云脑中的的基础版，资源规格为secmaster.basic

参数	是否必选	参数类型	描述
usage_measure_id	是	Integer	使用量单位标识，按需询价必填，例如按小时询价，使用量值为1，使用量单位为小时，枚举值如下： 4：小时 10：GB（带宽按流量询价使用） 11：MB（带宽按流量询价使用）
usage_value	是	Number	使用量值，按需询价必填，例如按小时询价，使用量值为1，使用量单位为小时
resource_size	是	Integer	配额个数
usage_factor	是	String	使用量因子，按需计费必填，取值和话单中的使用量因子一致，云服务和使用量因子对应关系如下： 典型场景配置：Duration 态势管理：duration 安全编排：count 智能分析：flow
resource_id	否	String	资源id，仅在增加配额的时候传入

响应参数

无

请求示例

```
https://{endpoint}/v1/{projectId}/subscriptions/orders

{
  "domain_id" : "abcdef8a41164a2280ec65f1f4c4mlnyz",
  "region_id" : "cn-north-4",
  "product_list" : [ {
    "product_id" : "OFFI908269345109094402",
    "cloud_service_type" : "hws.service.type.sa",
    "id" : "E52E1A22-9408-459A-9F67-7B5C11B1E71A",
    "resource_spec_code" : "secmaster.professional",
    "resource_type" : "hws.resource.type.secmaster.typical",
    "usage_factor" : "duration",
    "usage_value" : 1,
    "usage_measure_id" : 4,
    "resource_size" : 1
  } ]
}
```

响应示例

状态码： 400

参数异常

```
{
  "error_msg" : "云脑已包含【 标准版 】，如有需要请升级版本或增加配额",
  "error_code" : "SecMaster.00010201"
}
```

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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

public class CreatePostPaidOrderSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        CreatePostPaidOrderRequest request = new CreatePostPaidOrderRequest();
        PostPaidParam body = new PostPaidParam();
        List<ProductPostPaid> listbodyProductList = new ArrayList<>();
        listbodyProductList.add(
            new ProductPostPaid()
                .withId("E52E1A22-9408-459A-9F67-7B5C11B1E71A")
                .withProductId("OFFI908269345109094402")
                .withCloudServiceType("hws.service.type.sa")
                .withResourceType("hws.resource.type.secmaster.typical")
                .withResourceSpecCode("secmaster.professional")
                .withUsageMeasureId(ProductPostPaid.UsageMeasureIdEnum.NUMBER_4)
                .withUsageValue(java.math.BigDecimal.valueOf(1))
                .withResourceSize(1)
                .withUsageFactor("duration")
        );
```

```
        body.withProductList(listbodyProductList);
        body.withDomainId("abcdef8a41164a2280ec65f1f4c4mlnyz");
        body.withRegionId("cn-north-4");
        request.withBody(body);
        try {
            CreatePostPaidOrderResponse response = client.createPostPaidOrder(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 huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = CreatePostPaidOrderRequest()
        listProductListbody = [
            ProductPostPaid(
                id="E52E1A22-9408-459A-9F67-7B5C11B1E71A",
                product_id="OFFI908269345109094402",
                cloud_service_type="hws.service.type.sa",
                resource_type="hws.resource.type.secmaster.typical",
                resource_spec_code="secmaster.professional",
                usage_measure_id=4,
                usage_value=1,
                resource_size=1,
                usage_factor="duration"
            )
        ]
        request.body = PostPaidParam(
            product_list=listProductListbody,
            domain_id="abcdef8a41164a2280ec65f1f4c4mlnyz",
            region_id="cn-north-4"
        )
        response = client.create_post_paid_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

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreatePostPaidOrderRequest{}
    var listProductListbody = []model.ProductPostPaid{
        {
            Id: "E52E1A22-9408-459A-9F67-7B5C11B1E71A",
            ProductId: "OFFI908269345109094402",
            CloudServiceType: "hws.service.type.sa",
            ResourceType: "hws.resource.type.secmaster.typical",
            ResourceSpecCode: "secmaster.professional",
            UsageMeasureId: model.GetProductPostPaidUsageMeasureIdEnum().E_4,
            UsageValue: float32(1),
            ResourceSize: int32(1),
            UsageFactor: "duration",
        },
    }
    request.Body = &model.PostPaidParam{
        ProductList: &listProductListbody,
        DomainId: "abcdef8a41164a2280ec65f1f4c4mlnyz",
        RegionId: "cn-north-4",
    }
    response, err := client.CreatePostPaidOrder(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	参数异常
403	权限不足

错误码

请参见**错误码**。

4.18 指标查询

4.18.1 批量查询指标结果

功能介绍

批量查询指标结果

调用方法

请参见**如何调用API**。

URI

POST /v1/{project_id}/workspaces/{workspace_id}/sa/metrics/hits

表 4-749 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
workspace_id	是	String	工作空间ID

表 4-750 Query 参数

参数	是否必选	参数类型	描述
timespan	否	String	查询指标的时间范围，ISO8601格式，例如： 2007-03-01T13:00:00Z/2008-05-11T15:30:00Z或 2007-03-01T13:00:00Z/P1Y2M10DT2H30M或 P1Y2M10DT2H30M/2008-05-11T15:30:00Z
cache	否	Boolean	是否启用缓存，默认true, 禁用缓存 false

请求参数

表 4-751 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 4-752 请求 Body 参数

参数	是否必选	参数类型	描述
metric_ids	是	Array of strings	待查询的指标Id列表, 可参照附录中指标信息说明获取已有指标信息。
workspace_ids	否	Array of strings	工作空间列表, 当指标支持获取多工作空间数据时填写。
params	否	Array of Map<String,String> objects	待查询指标的参数列表，列表内每个元素为<String, String>的K-V形式，元素数量必须与metric_ids列表相同，具体填写方式请参照附录。
interactive_params	否	Array of Map<String,String> objects	交互式参数查询，当指标支持交互式参数时，填写<String, String>的K-V形式的参数列表，具体填写方式请参照附录。
field_ids	否	Array of strings	指标卡片ID列表

响应参数

状态码： 200

表 4-753 响应 Body 参数

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

表 4-754 ShowMetricResultResponseBody

参数	参数类型	描述
metric_id	String	指标ID
result	**result** object	指标查询结果内容
metric_format	Array of **MetricFormat** objects	指标显示格式，根据不同指标固定返回。
log_msg	String	结果日志信息
status	String	查询结果状态，SUCCESS：查询成功，FAILED：查询失败，FALLBACK：使用默认值

表 4-755 result

参数	参数类型	描述
labels	Array of strings	指标查询结果表格标题
datarows	Array<Array<Object>>	指标查询结果内容表格
effective_column	String	生效的列, 当有该参数时，使用指定列作为指标数据结果

表 4-756 MetricFormat

参数	参数类型	描述
data	String	数据格式
display	String	显示格式

参数	参数类型	描述
display_param	Map<String,String>	显示参数
data_param	Map<String,String>	数据参数

请求示例

通过指标接口查询从6月25日至当前的告警等级分布

```
https://{endpoint}/v1/{project_id}/workspaces/{workspace_id}/sa/metrics/hits

{
  "metric_ids" : [ "1f0f5e29-5a92-17a5-2c16-5f37c6dc109c" ],
  "params" : [ {
    "start_date" : "2024-06-25T00:00:00.000+08:00"
  } ]
}
```

响应示例

状态码： 200

请求成功

```
[ {
  "metric_id" : "1f0f5e29-5a92-17a5-2c16-5f37c6dc109c",
  "result" : {
    "labels" : [ "label1" ],
    "datarows" : [ [ { } ] ],
    "effective_column" : "0:1"
  },
  "status" : "SUCCESS"
} ]
```

SDK 代码示例

SDK代码示例如下。

Java

通过指标接口查询从6月25日至当前的告警等级分布

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;

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

```
public class BatchSearchMetricHitsSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        BatchSearchMetricHitsRequest request = new BatchSearchMetricHitsRequest();
        request.withWorkspaceId("{workspace_id}");
        BatchSearchMetricHitsRequestBody body = new BatchSearchMetricHitsRequestBody();
        Map<String, String> listParamsParams = new HashMap<>();
        listParamsParams.put("start_date", "2024-06-25T00:00:00.000+08:00");
        List<Map<String, String>> listbodyParams = new ArrayList<>();
        listbodyParams.add(listParamsParams);
        List<String> listbodyMetricIds = new ArrayList<>();
        listbodyMetricIds.add("1f0f5e29-5a92-17a5-2c16-5f37c6dc109c");
        body.withParams(listbodyParams);
        body.withMetricIds(listbodyMetricIds);
        request.withBody(body);
        try {
            BatchSearchMetricHitsResponse response = client.batchSearchMetricHits(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

通过指标接口查询从6月25日至当前的告警等级分布

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

```
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = BatchSearchMetricHitsRequest()
        request.workspace_id = "{workspace_id}"
        listParamsParams = {
            "start_date": "2024-06-25T00:00:00.000+08:00"
        }
        listParamsbody = [
            listParamsParams
        ]
        listMetricIdsbody = [
            "1f0f5e29-5a92-17a5-2c16-5f37c6dc109c"
        ]
        request.body = BatchSearchMetricHitsRequestBody(
            params=listParamsbody,
            metric_ids=listMetricIdsbody
        )
        response = client.batch_search_metric_hits(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

通过指标接口查询从6月25日至当前的告警等级分布

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
    request := &model.BatchSearchMetricHitsRequest{}
    request.WorkspaceId = "{workspace_id}"
    var listParamsParams = map[string]string{
        "start_date": "2024-06-25T00:00:00.000+08:00",
    }
    var listParamsbody = []map[string]string{
        listParamsParams,
    }
    var listMetricIdsbody = []string{
        "1f0f5e29-5a92-17a5-2c16-5f37c6dc109c",
    }
    request.Body = &model.BatchSearchMetricHitsRequestBody{
        Params: &listParamsbody,
        MetricIds: listMetricIdsbody,
    }
    response, err := client.BatchSearchMetricHits(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

错误码

请参见**错误码**。

4.19 基线检查

4.19.1 搜索基线检查结果列表

功能介绍

搜索基线检查结果列表

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/workspaces/{workspace_id}/sa/baseline/search

表 4-757 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
workspace_id	是	String	工作空间id

请求参数

表 4-758 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
X-Language	是	String	语言，参考值：zh-CN、en-US
content-type	是	String	内容类型

表 4-759 请求 Body 参数

参数	是否必选	参数类型	描述
limit	否	Integer	分页大小
offset	否	Integer	偏移量，表示查询该偏移量后面的记录
sort_by	否	String	排序关键字
order	否	String	降序或升序, DESC|ESC
from_date	否	String	起始时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
to_date	否	String	截止时间，格式ISO8601：YYYY-MM-DDTHH:mm:ss.ms+timezone。时区信息为事件发生时区，无法解析时区的时间，默认时区填东八区
condition	否	Object	搜索条件表达式

响应参数

状态码： 200

表 4-760 响应 Body 参数

参数	参数类型	描述
code	String	错误码
total	Integer	查询结果总数
size	Integer	分页大小
page	Integer	偏移量
success	Boolean	是否成功
data	Array of strings	查询结果列表

状态码： 400

表 4-761 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

状态码： 401

表 4-762 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

状态码： 500

表 4-763 响应 Body 参数

参数	参数类型	描述
code	String	错误码
message	String	错误描述

请求示例

查询基线检查结果的列表请求样例，查询2024年6月20号到2024年6月27号，遵从包ID为6add7d71-2261-4195-bab7-8ada0f0ed4d2，目录ID为

0b78937f-4d9b-4223-9a46-2361e5090be0， 资源类型为iam_user， 按照最近更新时间降序排序，返回第一页，每页10条数据

```
{
  "limit" : 10,
  "offset" : 0,
  "sort_by" : "last_observed_time",
  "order" : "DESC",
  "from_date" : "2024-06-20T00:00:00.000Z",
  "to_date" : "2024-06-27T23:59:59.999Z",
  "condition" : {
    "conditions" : [ {
      "name" : "compliance_package_id",
      "data" : [ "compliance_package_id", "=", "6add7d71-2261-4195-bab7-8ada0f0ed4d2" ]
    }, {
      "name" : "catalog_id",
      "data" : [ "catalog_id", "=", "0b78937f-4d9b-4223-9a46-2361e5090be0" ]
    }, {
      "name" : "resource.type",
      "data" : [ "resource.type", "=", "iam_user" ]
    } ],
    "logics" : [ "compliance_package_id", "AND", "catalog_id", "AND", "resource.type" ]
  }
}
```

响应示例

状态码： 200

请求成功

```
{
  "code" : "00000000",
  "data" : [ {
    "create_time" : "2024-01-03T01:16:21.666+08:00",
    "data_object" : {
      "arrive_time" : "2024-01-03T11:28:03.993Z+0800",
      "baseline_type" : {
        "baseline_type" : "合规检查",
        "baseline_type_en" : "Compliance Check",
        "baseline_type_zh" : "合规检查",
        "category" : "",
        "category_en" : "",
        "category_zh" : "",
        "id" : "23f48a58cXXX162846076cd0"
      },
      "catalog_id" : "9378d1e8-XXX-4aae-XXX-c41cf6829ede",
      "checkitem_id" : "13fcc967-cb49-XXX-811a-9f72ce6ce8ac",
      "compliance_package_id" : "39488f96-XXX-4cc6-XXX-ad3c29b3a6c2",
      "create_time" : "2024-01-02T17:16:21.666Z+0800",
      "data_source" : {
        "company_name" : "xxx",
        "domain_id" : "ac7438b990efXXXb45e8bf4",
        "product_feature" : "SA",
        "product_module" : "Base-line",
        "product_name" : "SecMaster",
        "project_id" : "15645222e8XXX93dab6341da6",
        "region_id" : "cn-north-7",
        "source_type" : 1
      },
      "dataclass_id" : "f846c8e0-XXX-XXX-bcbf-f77190847f08",
      "domain_id" : "ac7438b990eXXX1004eb45e8bf4",
      "domain_name" : "ac7438b99XXX1004eb45e8bf4",
      "end_time" : "2024-01-03T11:28:51.564Z+0800",
      "execitem_id" : "ca2a1361-5738-479c-8c40-d078e775a23a",
      "execitem_version" : 1,
      "first_observed_time" : "2024-01-03T11:28:50.955Z+0800",
      "handle_status" : "qualified",
```

```
    "id" : "39c56d70a9c2492XXXd91934cb5cb_13fcc967-XXX-494b-XXX-9f72ce6ce8ac",
    "is_deleted" : false,
    "last_observed_time" : "2024-01-03T11:28:51.564Z+0800",
    "method" : 1,
    "origin_id" : "",
    "project_id" : "15645222e874XXX93dab6341da6",
    "region_id" : "cn-north-7",
    "region_name" : "cn-north-7",
    "resource" : {
      "domain_id" : "ac7438b990eXXX04eb45e8bf4",
      "id" : "39c56d70a9cXXX1934cb5cb",
      "name" : "adfasd",
      "project_id" : "15645222XXXc93dab6341da6",
      "provider" : "xxx",
      "region_id" : "cn-north-7",
      "type" : "agency"
    },
    "severity" : "informational",
    "start_time" : "2024-01-03T11:28:50.955Z+0800",
    "task_id" : "10da8403-XXX-442d-XXX-fa2fdf42a3a1",
    "title" : "项目服务中的委托权限配置检查",
    "trigger_flag" : false,
    "update_time" : "2024-01-03T11:28:51.887Z+0800",
    "workspace_id" : "1350a050-XXX-45e2-XXX-9cbfef116de7"
  },
  "dataclass_ref" : {
    "id" : "f846c8e0-XXX-3767-XXX-f77190847f08"
  },
  "format_version" : 0,
  "id" : "39c56d7XXX278fXXX934cb5cb_13fcc967-cb49-XXX-811a-9f72ce6ce8ac",
  "update_time" : "2024-01-03T19:28:51.887+08:00",
  "version" : 0
}, {
  "create_time" : "2024-01-03T01:16:21.821+08:00",
  "data_object" : {
    "arrive_time" : "2024-01-03T11:28:03.993Z+0800",
    "baseline_type" : {
      "baseline_type" : "合规检查",
      "baseline_type_en" : "Compliance Check",
      "baseline_type_zh" : "合规检查",
      "category" : "",
      "category_en" : "",
      "category_zh" : "",
      "id" : "23f48a58c5b2fXXX162846076cd0"
    },
    "catalog_id" : "9378d1e8-XXX-4aae-XXX-c41cf6829ede",
    "checkitem_id" : "13fcc967-cb49-XXX-811a-9f72ce6ce8ac",
    "compliance_package_id" : "39488f96-XXX-4cc6-XXX-ad3c29b3a6c2",
    "create_time" : "2024-01-02T17:16:21.821Z+0800",
    "data_source" : {
      "company_name" : "xxx",
      "domain_id" : "ac7438b990efXXX004eb45e8bf4",
      "product_feature" : "SA",
      "product_module" : "Base-line",
      "product_name" : "SecMaster",
      "project_id" : "15645222XXX5c93dab6341da6",
      "region_id" : "cn-north-7",
      "source_type" : 1
    },
    "dataclass_id" : "f846c8e0-XXX-3767-bcbf-f77190847f08",
    "domain_id" : "ac7438b990eXXXb741004eb45e8bf4",
    "domain_name" : "ac7438bXXX37b741004eb45e8bf4",
    "end_time" : "2024-01-03T11:28:51.701Z+0800",
    "execitem_id" : "ca2a1361-XXX-479c-XXX-d078e775a23a",
    "execitem_version" : 1,
    "first_observed_time" : "2024-01-03T11:28:51.565Z+0800",
    "handle_status" : "qualified",
    "id" : "f295575ab57XXX977d9be93ca9fe_13fcc967-XXX-494b-XXX-9f72ce6ce8ac",
    "is_deleted" : false,
```

```
    "last_observed_time" : "2024-01-03T11:28:51.701Z+0800",
    "method" : 1,
    "origin_id" : "",
    "project_id" : "15645222e8XXXa985c93dab6341da6",
    "region_id" : "cn-north-7",
    "region_name" : "cn-north-7",
    "resource" : {
      "domain_id" : "ac7438b99XXX1004eb45e8bf4",
      "id" : "f295575ab57bXXXd9be93ca9fe",
      "name" : "apigw_admin_trust_secmaster",
      "project_id" : "15645222e8XXX93dab6341da6",
      "provider" : "xxx",
      "region_id" : "cn-north-7",
      "type" : "agency"
    },
    "severity" : "informational",
    "start_time" : "2024-01-03T11:28:51.565Z+0800",
    "task_id" : "10da8403-4955XXXd-a974-faXXX2a3a1",
    "title" : "项目服务中的委托权限配置检查",
    "trigger_flag" : false,
    "update_time" : "2024-01-03T11:28:52.023Z+0800",
    "workspace_id" : "1350a050-d09a-4XXX-9503-9cbfef116de7"
  },
  "dataclass_ref" : {
    "id" : "f846c8e0-cf0e-XXX-bcbf-XXX7f08"
  },
  "format_version" : 0,
  "id" : "f295575ab57b49XXXe93ca9fe_13fcc967-XXX-494b-XXX-9f72ce6ce8ac",
  "update_time" : "2024-01-03T19:28:52.023+08:00",
  "version" : 0
 } ],
 "page" : 0,
 "size" : 10,
 "success" : true,
 "total" : 2
}
```


状态码： 400

请求失败


```
{
 "error_code" : "SecMaster.00040006",
 "error_msg" : "Invalid request parameters"
}
```


状态码： 401

权限不足


```
{
 "error_code" : "SecMaster.90010015",
 "error_msg" : "Unauthorized request"
}
```


状态码： 500

请求失败


```
{
 "error_code" : "SecMaster.00040011",
 "error_msg" : "Internal system error."
}
```


SDK 代码示例

SDK代码示例如下。

Java

查询基线检查结果的列表请求样例，查询2024年6月20号到2024年6月27号，遵从包ID为6add7d71-2261-4195-bab7-8ada0f0ed4d2，目录ID为0b78937f-4d9b-4223-9a46-2361e5090be0， 资源类型为iam_user， 按照最近更新时间降序排序，返回第一页，每页10条数据

```
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.secmaster.v2.region.SecMasterRegion;
import com.huaweicloud.sdk.secmaster.v2.*;
import com.huaweicloud.sdk.secmaster.v2.model.*;


public class SearchBaselineSolution {

    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);

        SecMasterClient client = SecMasterClient.newBuilder()
                .withCredential(auth)
                .withRegion(SecMasterRegion.valueOf("<YOUR REGION>"))
                .build();
        SearchBaselineRequest request = new SearchBaselineRequest();
        request.withWorkspaceId("{workspace_id}");
        BaselineSearchRequestBody body = new BaselineSearchRequestBody();
        body.withCondition("{\"logics\":[\"compliance_package_id\",\"AND\",\"catalog_id\",\"AND\",\"resource.type\"],\"conditions\":[{\"data\":[\"compliance_package_id\",\"=\",\"6add7d71-2261-4195-bab7-8ada0f0ed4d2\"],\"name\":\"compliance_package_id\"},{\"data\":[\"catalog_id\",\"=\",\"0b78937f-4d9b-4223-9a46-2361e5090be0\"],\"name\":\"catalog_id\"},{\"data\":[\"resource.type\",\"=\",\"iam_user\"],\"name\":\"resource.type\"}]}");
        body.withToDate("2024-06-27T23:59:59.999Z");
        body.withFromDate("2024-06-20T00:00:00.000Z");
        body.withOrder("DESC");
        body.withSortBy("last_observed_time");
        body.withOffset(0);
        body.withLimit(10);
        request.withBody(body);
        try {
            SearchBaselineResponse response = client.searchBaseline(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

查询基线检查结果的列表请求样例，查询2024年6月20号到2024年6月27号，遵从包ID为6add7d71-2261-4195-bab7-8ada0f0ed4d2，目录ID为0b78937f-4d9b-4223-9a46-2361e5090be0， 资源类型为iam_user， 按照最近更新时间降序排序，返回第一页，每页10条数据

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdksecmaster.v2.region.secmaster_region import SecMasterRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdksecmaster.v2 import *

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

    credentials = BasicCredentials(ak, sk, projectId)

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

    try:
        request = SearchBaselineRequest()
        request.workspace_id = "{workspace_id}"
        request.body = BaselineSearchRequestBody(
            condition="{\"logics\":[\"compliance_package_id\",\"AND\",\"catalog_id\",\"AND\",\"resource.type\"],\"conditions\":[{\"data\":[\"compliance_package_id\",\"=\",\"6add7d71-2261-4195-bab7-8ada0f0ed4d2\"],\"name\":\"compliance_package_id\"},{\"data\":[\"catalog_id\",\"=\",\"0b78937f-4d9b-4223-9a46-2361e5090be0\"],\"name\":\"catalog_id\"},{\"data\":[\"resource.type\",\"=\",\"iam_user\"],\"name\":\"resource.type\"}]}",
            to_date="2024-06-27T23:59:59.999Z",
            from_date="2024-06-20T00:00:00.000Z",
            order="DESC",
            sort_by="last_observed_time",
            offset=0,
            limit=10
        )
        response = client.search_baseline(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

查询基线检查结果的列表请求样例，查询2024年6月20号到2024年6月27号，遵从包ID为6add7d71-2261-4195-bab7-8ada0f0ed4d2，目录ID为0b78937f-4d9b-4223-9a46-2361e5090be0， 资源类型为iam_user， 按照最近更新时间降序排序，返回第一页，每页10条数据

```
package main

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

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

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

    client := secmaster.NewSecMasterClient(
        secmaster.SecMasterClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.SearchBaselineRequest{}
    request.WorkspaceId = "{workspace_id}"
    var conditionBaselineSearchRequestBody interface{} = "{\"logics\":[\"compliance_package_id\",\"AND\",\"catalog_id\",\"AND\",\"resource.type\"],\"conditions\":[{\"data\":[\"compliance_package_id\",\"=\",\"6add7d71-2261-4195-bab7-8ada0f0ed4d2\"],\"name\":\"compliance_package_id\"},{\"data\":[\"catalog_id\",\"=\",\"0b78937f-4d9b-4223-9a46-2361e5090be0\"],\"name\":\"catalog_id\"},{\"data\":[\"resource.type\",\"=\",\"iam_user\"],\"name\":\"resource.type\"}]}"
    toDateBaselineSearchRequestBody:= "2024-06-27T23:59:59.999Z"
    fromDateBaselineSearchRequestBody:= "2024-06-20T00:00:00.000Z"
    orderBaselineSearchRequestBody:= "DESC"
    sortByBaselineSearchRequestBody:= "last_observed_time"
    offsetBaselineSearchRequestBody:= int32(0)
    limitBaselineSearchRequestBody:= int32(10)
    request.Body = &model.BaselineSearchRequestBody{
        Condition: &conditionBaselineSearchRequestBody,
        ToDate: &toDateBaselineSearchRequestBody,
        FromDate: &fromDateBaselineSearchRequestBody,
        Order: &orderBaselineSearchRequestBody,
        SortBy: &sortByBaselineSearchRequestBody,
        Offset: &offsetBaselineSearchRequestBody,
        Limit: &limitBaselineSearchRequestBody,
    }
    response, err := client.SearchBaseline(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	请求失败
401	权限不足
500	请求失败

错误码

请参见**错误码**。

A 附录

A.1 状态码

- 正常

返回值	说明
200	成功。
201	成功。

- 异常

状态码	编码	说明
400	Bad Request	参数错误。
401	Unauthorized	认证失败。
403	Forbidden	拒绝访问。
500	Internal Server Error	系统内部错误。

A.2 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
400	SecMaster.11061001	进程状态有误
400	SecMaster.11061002	模型数量超出范围限制
400	SecMaster.11061003	schedule参数超出范围

状态码	错误码	错误信息
400	SecMaster.11061004	告警名称已存在
400	SecMaster.20010001	无效的工作空间ID
400	SecMaster.20030001	无效的参数
400	SecMaster.20030002	无效的项目ID
400	SecMaster.20030003	无效的名称
400	SecMaster.20030004	创建数据对象失败
400	SecMaster.20030005	获取数据对象失败
400	SecMaster.20030009	无效的排序字段
400	SecMaster.20030010	无效的排序
400	SecMaster.20030011	更新数据对象错误
400	SecMaster.20030012	删除数据对象错误
400	SecMaster.20030013	搜索数据对象错误
400	SecMaster.20030022	查询特定数据类失败
400	SecMaster.20030025	验证数据对象失败
400	SecMaster.20039999	未知错误
400	SecMaster.20040000	未知错误
400	SecMaster.20040402	查询数据类失败
400	SecMaster.20040516	字段超过最大限制
400	SecMaster.20041001	无效的工作空间ID
400	SecMaster.20041002	无效的参数
400	SecMaster.20041003	无效的项目ID
400	SecMaster.20041031	获取数据对象失败
400	SecMaster.20041033	未选择关联数据对象
400	SecMaster.20041504	创建事件失败
400	SecMaster.20041507	更新事件失败
400	SecMaster.20041508	删除事件失败
400	SecMaster.20041509	单日事件创建个数超过最大限制
400	SecMaster.20041804	告警转事件请求内容错误
400	SecMaster.20041805	创建告警失败
400	SecMaster.20041808	更新告警失败

状态码	错误码	错误信息
400	SecMaster.20041809	删除告警失败
400	SecMaster.20041810	单日告警创建个数超过最大限制
400	SecMaster.20041811	单日告警转事件个数超过最大限制
400	SecMaster.20041903	获取数据类失败
400	SecMaster.20041904	威胁情报数据不存在
400	SecMaster.20041905	创建威胁情报失败
400	SecMaster.20041906	更新威胁情报失败
400	SecMaster.20041907	删除威胁情报失败
400	SecMaster.20042501	单日指标创建个数超过最大限制
400	SecMaster.20048001	剧本存在正在运行的实例或存在激活版本不能删除
400	SecMaster.20048002	剧本不存在激活版本，不能启用
400	SecMaster.20048003	剧本状态错误，不能审核
400	SecMaster.20048004	资源不存在
400	SecMaster.20048005	剧本审核不通过，不能激活
400	SecMaster.20048006	剧本ID错误
400	SecMaster.20048007	剧本版本ID错误
400	SecMaster.20048008	剧本动作ID错误
400	SecMaster.20048009	剧本规则ID错误
400	SecMaster.20048013	剧本启用中，不能失活版本
400	SecMaster.20048014	剧本已经发布，不能编辑
400	SecMaster.20048015	剧本名称重复
400	SecMaster.20048016	剧本定时任务时间范围错误
400	SecMaster.20048017	剧本定时任务Corn表达式错误
400	SecMaster.20048018	版本数量已达到上限
400	SecMaster.20048019	剧本存在审核中版本，不能新建版本
400	SecMaster.20048020	数据对象ID错误
400	SecMaster.20048021	搜索内容无效
400	SecMaster.20048022	查询结束时间必须大于查询起始时间
400	SecMaster.20048023	注册剧本定时任务失败

状态码	错误码	错误信息
400	SecMaster.20048024	禁用剧本定时任务失败
400	SecMaster.20048025	结束时间必须大于开始时间
400	SecMaster.20048026	无效的剧本结束时间
400	SecMaster.20048027	数据类ID不能为空
400	SecMaster.20048028	存在未启用的匹配流程，不能提交版本
400	SecMaster.20048029	剧本数据转换错误
400	SecMaster.20048030	剧本个数超过最大限制
400	SecMaster.20048031	剧本关联匹配流程个数超过限制
400	SecMaster.20048032	无效的剧本调度时间间隔
400	SecMaster.20048033	剧本关联的匹配流程不能为空
400	SecMaster.20048034	匹配流程与剧本数据类不一致
400	SecMaster.20048035	系统内置剧本不允许修改
400	SecMaster.20048036	系统内置剧本不允许删除

A.3 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.4 指标信息说明

表 A-1 指标信息说明

指标名称	指标ID	指标描述	指标参数说明
所有资产数量	6f8d4892-713c-4d12-8584-dc04f7847b32	工作空间内所有资产数量	无
高危资产数量	a5597747-8cef-4342-9855-3fdaf00ad460	工作空间内高危资产数量	无
其它风险资产数量	09ca4eb8-a4ca-4ef4-b75f-e9172f39393b	工作空间内除高危风险资产外资产数量	无
告警等级数量分布	1f0f5e29-5a92-17a5-2c16-5f37c6dc109c	从参数起始时间到当前时间范围内，工作空间内告警等级数量分布	“params”参数填写“start_date”表示统计起始时间，例如： "start_date": "2024-06-21T00:00:00.000+08:00"
漏洞等级数量分布	815c8a73-c855-fd29-63e2-b093d05a7ef0	工作空间内漏洞不同等级漏洞分布	无
基线检查不合格结果等级分布	fee4d416-25b4-46c6-aa1b-851c7251e04b	工作空间内30天内不同等级基线检查结果分布	无

指标名称	指标ID	指标描述	指标参数说明
安全评分趋势	39d386dc-5868-adb6-a8e9-d5e92bb75663	工作空间内近七天每天安全评分得分	无
威胁告警事件Top5	aaa6e851-601b-53c5-61ef-ffc95889ebf3	从参数起始时间到当前时间范围内，工作空间内威胁告警事件TOP5	“params”参数填写"start_date"表示统计起始时间，例如： "start_date": "2024-06-21T00:00:00.000+08:00"
工作空间安全评分	cf6cce38-bc32-fd89-c0b5-3ba2cdf98eda	工作空间内最新一次安全评分得分	无