代码检查

API 参考

文档版本 01

发布日期 2024-03-06

华为技术有限公司

版权所有 © 华为技术有限公司 2024。 保留一切权利。

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

商标声明

HUAWEI和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： https://www.huawei.com

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：
https://www.huawei.com/cn/psirt/vul-response-process
如企业客户须获取漏洞信息，请参见如下网址：
https://securitybulletin.huawei.com/enterprise/cn/security-advisory

目 录

1 使用前必读.....1

2 API 概览.....3

3 如何调用 API.....4

3.1 构造请求.....4

3.2 认证鉴权.....6

3.3 返回结果.....8

4 API.....10

4.1 任务管理.....10

4.1.1 新建检查任务.....10

4.1.2 删除检查任务.....17

4.1.3 查询任务列表.....21

4.1.4 执行检查任务.....26

4.1.5 终止检查任务.....31

4.1.6 查询任务执行状态.....35

4.1.7 历史扫描结果查询.....37

4.1.8 查询任务的已选规则集列表 v2.....41

4.1.9 查询任务的已选规则集列表 v3.....44

4.1.10 查询任务规则集的检查参数 v2.....47

4.1.11 查询任务规则集的检查参数 v3.....50

4.1.12 任务配置检查参数.....54

4.1.13 修改任务规则集.....60

4.1.14 查询任务检查失败日志.....65

4.1.15 获取任务的目录树.....69

4.1.16 任务配置屏蔽目录.....74

4.1.17 查询任务的高级选项.....79

4.1.18 任务配置高级选项.....82

4.2 缺陷管理.....87

4.2.1 查询缺陷概要.....87

4.2.2 查询 cmertrics 缺陷概要.....91

4.2.3 查询缺陷详情.....97

4.2.4 查询缺陷详情的统计.....103

4.2.5 修改缺陷状态.....105

 版权所有 © 华为技术有限公司

4.3 规则管理.....110
4.3.1 获取规则列表接口.....110
4.3.2 创建自定义规则集.....114
4.3.3 查询规则集列表.....120
4.3.4 查看规则集的规则列表.....124
4.3.5 删除自定义规则集.....128
4.3.6 设置每个项目对应语言的默认规则集配置.....132
5 应用示例.....137
5.1 示例 1：查询任务列表.....137
5.2 示例 2：查询缺陷概要.....139
6 附录.....141
6.1 状态码.....141
6.2 错误码.....143
6.3 获取项目 ID.....145
6.4 获取账号 ID.....146
7 修订记录.....147

1 使用前必读

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

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

在调用代码检查API之前，请确保已经充分了解代码检查**基本概念**。详细信息请参见**产品介绍**。

终端节点

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

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

表 1-1 代码检查的终端节点

区域名称	区域	终端节点（Endpoint）
华北-北京四	cn-north-4	codecheck-ext.cn-north-4.myhuaweicloud.com
华北-北京一	cn-north-1	codecheck-ext.cn-north-1.myhuaweicloud.com
华东-上海二	cn-east-2	codecheck-ext.cn-east-2.myhuaweicloud.com

基本概念

- 账号

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

版权所有 © 华为技术有限公司

- 用户

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

  在**我的凭证**下，您可以查看账号ID和用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。

- 区域（Region）

  从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region。

  – 通用Region指面向公共租户提供通用云服务的Region。
  – 专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。

- 可用区（AZ，Availability Zone）

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

- 企业项目

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

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

版权所有 © 华为技术有限公司

2 API 概览

表 2-1 代码检查 API 概览

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

版权所有 © 华为技术有限公司

3 如何调用 API

3.1 构造请求

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

您还可以通过这个视频教程了解如何构造请求调用API：**https://bbs.huaweicloud.com/videos/102987** 。

请求 URI

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

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

各参数含义如下：

参数	说明
URI-scheme	表示用于传输请求的协议，当前所有API均采用**HTTPS**协议。
Endpoint	指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从地区和终端节点中获取。 例如IAM服务在“华北-北京一”区域的Endpoint为“iam.cn-north-1.myhuaweicloud.com”。
resource-path	资源路径，即API访问路径，从具体API的URI模块获取。例如“理员创建IAM用户”API的resource-path为“/v3.0/OS-USER/users”。
query-string	查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个**?** ，形式为**参数名=参数取值**。例如，**limit=10**表示查询不超过10条数据。

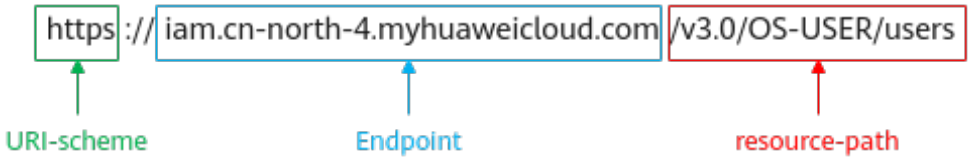
例如，您需要创建IAM用户，由于IAM为全局服务，则使用任一区域的Endpoint即可，比如“华北-北京一”区域的Endpoint（iam.cn-north-1.myhuaweicloud.com），并

版权所有 © 华为技术有限公司

在管理员创建IAM用户的URI部分找到resource-path（/v3.0/OS-USER/users），拼接起来如下所示。

```
s://iam.cn-north-1.myhuaweicloud.com/v3.0/OS-USER/users
```

图 3-1 URI 示意图



说明

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

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

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

在管理员创建IAM用户的URI部分，您可以看到其请求方法为“POST”，则其请求为：

```
https://iam.cn-north-1.myhuaweicloud.com/v3.0/OS-USER/users
```

请求消息头

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

如下公共消息头需要添加到请求中。

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

版权所有 © 华为技术有限公司

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

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

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=****************, SignedHeaders=content-type;host;x-sdk-date,
Signature=****************
```

请求消息体

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

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

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

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

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=****************, SignedHeaders=content-type;host;x-sdk-date, Signature=****************

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

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

3.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

版权所有 © 华为技术有限公司

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

Token 认证

说明

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

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

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

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

```
GET https://iam.cn-north-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

您还可以通过这个视频教程了解如何使用Token认证：**https://bbs.huaweicloud.com/videos/101333** 。

AK/SK 认证

说明

AK/SK签名认证方式仅支持消息体大小12M以内，12M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。

版权所有 © 华为技术有限公司

- 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。

您也可以通过这个视频教程了解AK/SK认证的使用：**https://bbs.huaweicloud.com/videos/100697** 。

3.3 返回结果

状态码

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

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

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

响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

对于**管理员创建IAM用户**接口，返回如**图3-2**所示的消息头。

图 3-2 管理员创建 IAM 用户响应消息头

```
"X-Frame-Options": "SAMEORIGIN",
"X-IAM-ETag-id": "2562365939-d8f6f12921974cb097338ac11fceac8a",
"Transfer-Encoding": "chunked",
"Strict-Transport-Security": "max-age=31536000; includeSubdomains;",
"Server": "api-gateway",
"X-Request-Id": "af2953f2bcc67a42325a69a19e6c32a2",
"X-Content-Type-Options": "nosniff",
"Connection": "keep-alive",
"X-Download-Options": "noopen",
"X-XSS-Protection": "1; mode=block;",
"X-IAM-Trace-Id": "token_[redacted]_null_af2953f2bcc67a42325a69a19e6c32a2",
"Date": "Tue, 21 May 2024 09:03:40 GMT",
"Content-Type": "application/json; charset=utf8"
```

响应消息体

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

版权所有 © 华为技术有限公司

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

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

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

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

其中，error_code表示错误码，error_msg表示错误描述信息。

版权所有 © 华为技术有限公司

4 API

4.1 任务管理

4.1.1 新建检查任务

功能介绍

新建检查任务但是不执行。

调用方法

请参见**如何调用API**。

URI

POST /v2/{project_id}/task

表 4-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目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）。 最小长度：**1** 最大长度：**100000**
Content-Type	是	String	设置媒体类型和编码格式

表 4-3 请求 Body 参数

参数	是否必选	参数类型	描述
check_type	否	Array of strings	检查类型，数组格式，默认为source
git_url	是	String	仓库地址
git_branch	是	String	仓库分支
language	是	Array of strings	检查语言，数组格式，支持cpp,java,js,python,php,css,html,go,typescript,csharp
rule_sets	否	Array of **RuleSetV2** objects	指定规则集
task_type	否	String	检查类型，支持full/inc两种类型，full表示全量检查，inc表示mr检查
username	否	String	仓库有权限的用户名
access_token	否	String	仓库有权限的用户token
endpoint_id	否	String	仓库有权限的用户endpointId
inc_config	否	**IncConfigV2** object	增量检查相关的参数
enable_fossbot	否	Boolean	是否打开fossbot检查,默认不开
resource_pool_id	否	String	资源池id，可以从资源池管理页面获取

表 4-4 RuleSetV2

参数	是否必选	参数类型	描述
ruleset_id	否	String	规则集id，需要从web界面获取
language	是	String	检查语言，支持cpp,java,js,python,php,css,html,go,typescript,csharp

表 4-5 IncConfigV2

参数	是否必选	参数类型	描述
parent_task_id	否	String	需要关联的父任务ID，流水线创建或MR创建任务需要该参数
git_source_branch	否	String	增量检查代码源分支
git_target_branch	否	String	增量检查代码目标分支
merge_id	否	String	MR唯一标识ID
event_type	否	String	webhook触发事件类型,merge_request/push_request
action	否	String	webhook事件状态，open/close/update
title	否	String	MR标题

响应参数

状态码： 200

表 4-6 响应 Body 参数

参数	参数类型	描述
task_id	String	检查任务id

状态码： 400

表 4-7 响应 Body 参数

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

状态码： 401

表 4-8 响应 Body 参数

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

请求示例

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

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

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

响应示例

状态码： 200

Request succeeded!

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

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

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

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

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

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

public class CreateTaskSolution {

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

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

        CodeCheckClient client = CodeCheckClient.newBuilder()
                .withCredential(auth)
                .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateTaskRequest request = new CreateTaskRequest();
        CreateTaskRequestV2 body = new CreateTaskRequestV2();
        List<String> listbodyLanguage = new ArrayList<>();
        listbodyLanguage.add("cpp");
        listbodyLanguage.add("js");
        List<String> listbodyCheckType = new ArrayList<>();
        listbodyCheckType.add("source");
        body.withLanguage(listbodyLanguage);
        body.withGitBranch("master");
        body.withGitUrl("git@code****************958.git");
        body.withCheckType(listbodyCheckType);
        request.withBody(body);
        try {
            CreateTaskResponse response = client.createTask(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
```

```
    }
}
```

Python

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

```
# coding: utf-8

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

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

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

```
package main

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

```
)

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

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

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

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

更多

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

状态码

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

错误码

请参见**错误码**。

4.1.2 删除检查任务

功能介绍

删除检查任务，执行中的任务删除无法再查看

调用方法

请参见**如何调用API**。

URI

DELETE /v2/tasks/{task_id}

表 4-9 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	检查任务ID

请求参数

表 4-10 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 400

表 4-11 响应 Body 参数

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

状态码： 401

表 4-12 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

```
null
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

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

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


public class DeleteTaskSolution {

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

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

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

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

Python

```
# coding: utf-8

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

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

    credentials = BasicCredentials(ak, sk) \

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

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

Go

```
package main

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

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

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

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

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

更多

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

状态码

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

错误码

请参见**错误码**。

4.1.3 查询任务列表

功能介绍

根据project_id查询该项目下的任务列表。

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/tasks

表 4-13 路径参数

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

表 4-14 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页索引，偏移量 最小值：**0** 缺省值：**0**
limit	否	Integer	每页显示的数量,每页最多显示100条 最小值：**1** 最大值：**100** 缺省值：**10**

请求参数

表 4-15 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-16 响应 Body 参数

参数	参数类型	描述
tasks	Array of **SimpleTaskInfoV2** objects	任务信息
total	Integer	总数

表 4-17 SimpleTaskInfoV2

参数	参数类型	描述
task_id	String	任务id
task_name	String	任务名字
creator_id	String	创建者id
git_url	String	代码仓地址
git_branch	String	代码仓分支,如果是MR模式，为源分支
created_at	String	创建时间
last_check_time	String	上一次检查时间

状态码： 400

表 4-18 响应 Body 参数

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

状态码： 401

表 4-19 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误描述

请求示例

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

响应示例

状态码： 200

Request succeeded!

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

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

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

import com.huaweicloud.sdk.core.auth.ICredential;
```

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


public class ShowTaskListByProjectIdSolution {

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

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

        CodeCheckClient client = CodeCheckClient.newBuilder()
                .withCredential(auth)
                .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
                .build();
        ShowTaskListByProjectIdRequest request = new ShowTaskListByProjectIdRequest();
        request.withOffset(<offset>);
        request.withLimit(<limit>);
        try {
            ShowTaskListByProjectIdResponse response = client.showTaskListByProjectId(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

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

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

    credentials = BasicCredentials(ak, sk) \

    client = CodeCheckClient.new_builder() \
```

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

    try:
        request = ShowTaskListByProjectIdRequest()
        request.offset = <offset>
        request.limit = <limit>
        response = client.show_task_list_by_project_id(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

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

    request := &model.ShowTaskListByProjectIdRequest{}
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    response, err := client.ShowTaskListByProjectId(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

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

错误码

请参见错误码。

4.1.4 执行检查任务

功能介绍

执行检查任务。

调用方法

请参见如何调用API。

URI

POST /v2/tasks/{task_id}/run

表 4-20 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	任务ID

请求参数

表 4-21 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	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）。 最小长度：**1** 最大长度：**100000**
Content-Type	是	String	设置媒体类型和编码格式

表 4-22 请求 Body 参数

参数	是否必选	参数类型	描述
username	否	String	该任务对应临时仓库有权限的用户名
access_token	否	String	该任务对应临时仓库有权限的用户token
git_url	否	String	该任务对应的临时仓库地址
git_branch	否	String	该任务对应的临时仓库分支

响应参数

状态码： 200

表 4-23 响应 Body 参数

参数	参数类型	描述
exec_id	String	执行id

状态码： 400

表 4-24 响应 Body 参数

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

状态码： 401

表 4-25 响应 Body 参数

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

请求示例

执行检查任务:执行任务对应代码仓URL根据实际显示，代码仓分支为master，用户名根据实际显示

```
POST https://{endpoint}/v2/tasks/{taskid}/run

{
  "git_url" : "http://github.xxxxxxxx.git",
  "git_branch" : "master",
  "username" : "test",
  "access_token" : "xLD56xxxxxxxxxJF3Sdxxxx"
}
```

响应示例

状态码： 200

Request succeeded!

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

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

执行检查任务:执行任务对应代码仓URL根据实际显示，代码仓分支为master，用户名根据实际显示

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


public class RunTaskSolution {

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

```
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

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

        CodeCheckClient client = CodeCheckClient.newBuilder()
                .withCredential(auth)
                .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
                .build();
        RunTaskRequest request = new RunTaskRequest();
        RunRequestV2 body = new RunRequestV2();
        body.withGitBranch("master");
        body.withGitUrl("http://github.xxxxxxxx.git");
        body.withAccessToken("xLD56xxxxxxxxxJF3Sdxxxx");
        body.withUsername("test");
        request.withBody(body);
        try {
            RunTaskResponse response = client.runTask(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

执行检查任务:执行任务对应代码仓URL根据实际显示，代码仓分支为master，用户名根据实际显示

```
# coding: utf-8

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

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = RunTaskRequest()
        request.body = RunRequestV2(
            git_branch="master",
            git_url="http://github.xxxxxxxx.git",
            access_token="xLD56xxxxxxxxxJF3Sdxxxx",
```

```
        username="test"
    )
    response = client.run_task(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

执行检查任务:执行任务对应代码仓URL根据实际显示，代码仓分支为master，用户名根据实际显示

```
package main

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

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

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

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

    request := &model.RunTaskRequest{}
    gitBranchRunRequestV2:= "master"
    gitUrlRunRequestV2:= "http://github.xxxxxxxx.git"
    accessTokenRunRequestV2:= "xLD56xxxxxxxxxJF3Sdxxxx"
    usernameRunRequestV2:= "test"
    request.Body = &model.RunRequestV2{
        GitBranch: &gitBranchRunRequestV2,
        GitUrl: &gitUrlRunRequestV2,
        AccessToken: &accessTokenRunRequestV2,
        Username: &usernameRunRequestV2,
    }
    response, err := client.RunTask(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

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

错误码

请参见**错误码**。

4.1.5 终止检查任务

功能介绍

根据任务ID终止检查任务。

调用方法

请参见**如何调用API**。

URI

POST /v2/tasks/{task_id}/stop

表 4-26 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	任务ID

请求参数

表 4-27 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 400

表 4-28 响应 Body 参数

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

状态码： 401

表 4-29 响应 Body 参数

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

请求示例

```
POST https://{endpoint}/v2/tasks/2b31ed520xxxxxxebedb6e57xxxxxxxx/stop
```

响应示例

状态码： 200

Request succeeded!

```
null
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

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

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


public class StopTaskByIdSolution {

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

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

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

Python

```
# coding: utf-8

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

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

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

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

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

Go

```
package main

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

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

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

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

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

更多

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

状态码

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

错误码

请参见**错误码**。

4.1.6 查询任务执行状态

功能介绍

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

调用方法

请参见**如何调用API**。

URI

GET /v2/tasks/{task_id}/progress

表 4-30 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	任务ID

请求参数

表 4-31 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-32 响应 Body 参数

参数	参数类型	描述
task_status	Integer	任务状态,0表示检查中，1表示检查失败，2表示检查成功，3表示任务中止
progress	**ProgressDetailV2** object	检查进度

表 4-33 ProgressDetailV2

参数	参数类型	描述
ratio	String	进度百分比
info	String	进度信息

状态码： 400

表 4-34 响应 Body 参数

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

状态码： 401

表 4-35 响应 Body 参数

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

请求示例

```
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxxebedb6e57xxxxxxxx/progress
```

响应示例

状态码： 200

Request succeeded!

```
{
  "task_status" : 0,
  "progress" : {
    "ratio" : "50%",
    "info" : "执行检查中"
  }
}
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

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

错误码

请参见**错误码**。

4.1.7 历史扫描结果查询

功能介绍

提供每次扫描的问题数量统计

调用方法

请参见**如何调用API**。

URI

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

表 4-36 路径参数

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

表 4-37 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页索引，偏移量 最小值：**0** 缺省值：**0**
limit	否	Integer	每页显示的数量,每页最多显示1000条 最小值：**1** 最大值：**1000** 缺省值：**10**
start_time	否	String	过滤开始时间,根据任务检查开始时间过滤
end_time	否	String	过滤结束时间,根据任务检查结束时间过滤

请求参数

表 4-38 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-39 响应 Body 参数

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

表 4-40 CheckRecordDataInfo

参数	参数类型	描述
check_time	String	检查任务执行开始时间
check_end_time	String	检查任务执行结束时间
issue_counts	**CheckRecordIssueCountsInfo** object	严重程度

表 4-41 CheckRecordIssueCountsInfo

参数	参数类型	描述
critical	Integer	致命问题
serious	Integer	严重问题
normal	Integer	常规问题
prompt	Integer	提示问题

状态码： 400

表 4-42 响应 Body 参数

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

状态码： 401

表 4-43 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

```
{
  "total" : 100,
  "data" : [ {
    "check_time" : "2020-02-20 11:37:46",
    "check_end_time" : "2020-02-20 11:47:46",
    "issue_counts" : {
      "critical" : 0,
      "serious" : 1,
      "normal" : 2,
      "prompt" : 3
    }
  }, {
    "check_time" : "2020-02-20 11:37:46",
    "check_end_time" : "2020-02-20T 11:47:46Z",
    "issue_counts" : {
      "critical" : 0,
      "serious" : 1,
      "normal" : 2,
      "prompt" : 3
    }
  } ]
}
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

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

错误码

请参见**错误码**。

4.1.8 查询任务的已选规则集列表 v2

功能介绍

查询任务的已选规则集列表v2。

调用方法

请参见**如何调用API**。

URI

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

表 4-44 路径参数

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

请求参数

表 4-45 请求 Header 参数

参数	是否必选	参数类型	描述
Content-Type	否	String	设置媒体类型和编码格式

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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-46 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of **TaskRulesetInfo** objects	Request succeeded!

表 4-47 TaskRulesetInfo

参数	参数类型	描述
template_id	String	规则集id
language	String	规则集语言
template_name	String	规则集名称
type	String	规则集状态optional：可选，selected：已选
status	String	规则集属性0 是默认用户规则集,1 是系统默认规则集

状态码： 400

表 4-48 响应 Body 参数

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

状态码： 401

表 4-49 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

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

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

状态码	描述
200	Request succeeded!
400	Bad Request

状态码	描述
401	Unauthorized

错误码

请参见错误码。

4.1.9 查询任务的已选规则集列表 v3

功能介绍

查询任务的已选规则集列表v3。

调用方法

请参见如何调用API。

URI

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

表 4-50 路径参数

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

表 4-51 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页索引，偏移量 最小值：**0** 缺省值：**0**
limit	否	Integer	每页显示的数量,每页最多显示1000条 最小值：**1** 最大值：**1000** 缺省值：**10**

请求参数

表 4-52 请求 Header 参数

参数	是否必选	参数类型	描述
Content-Type	否	String	设置媒体类型和编码格式
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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-53 响应 Body 参数

参数	参数类型	描述
info	Array of **TaskRulesetInfo** objects	规则集信息
total	Integer	总数

表 4-54 TaskRulesetInfo

参数	参数类型	描述
template_id	String	规则集id
language	String	规则集语言
template_name	String	规则集名称
type	String	规则集状态optional：可选，selected：已选
status	String	规则集属性0 是默认用户规则集,1 是系统默认规则集

状态码： 400

表 4-55 响应 Body 参数

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

状态码： 401

表 4-56 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

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

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

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

错误码

请参见**错误码**。

4.1.10 查询任务规则集的检查参数 v2

功能介绍

查询任务规则集的检查参数v2

调用方法

请参见**如何调用API**。

URI

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

表 4-57 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
task_id	是	String	任务ID
ruleset_id	是	String	规则集ID

表 4-58 Query 参数

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

请求参数

表 4-59 请求 Header 参数

参数	是否必选	参数类型	描述
Content-Type	否	String	设置媒体类型和编码格式
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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-60 响应 Body 参数

参数	参数类型	描述
[数组元素]	Array of **TaskCheckParamters** objects	Request succeeded!

表 4-61 TaskCheckParamters

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

表 4-62 CheckConfigInfo

参数	参数类型	描述
name	String	检查参数名称
cfg_key	String	检查参数对应的key值
default_value	String	检查参数默认值

参数	参数类型	描述
option_value	String	检查参数可选项
is_required	Integer	0：非必填，1：必填
description	String	检查参数说明
type	Integer	参数类型，0：文本，2：有可选项
status	String	参数状态，on：启用，off：未启用
value	String	检查参数值

状态码： 400

表 4-63 响应 Body 参数

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

状态码： 401

表 4-64 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

```
[ {
  "check_id" : 0,
  "name" : "compile_tool",
  "check_configs" : [ {
    "value" : "",
    "name" : "编译命令",
    "cfg_key" : "",
    "default_value" : "",
    "option_value" : "",
```

```
    "is_required" : 0,
    "description" : "编译命令",
    "type" : 0,
    "status" : "on"
  } ]
} ]
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

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

错误码

请参见错误码。

4.1.11 查询任务规则集的检查参数 v3

功能介绍

查询任务规则集的检查参数v3

调用方法

请参见如何调用API。

URI

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

表 4-65 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
task_id	是	String	任务ID
ruleset_id	是	String	规则集ID

表 4-66 Query 参数

参数	是否必选	参数类型	描述
language	是	String	规则集语言
offset	否	Integer	分页索引，偏移量，非必填 最小值：**0** 缺省值：**0**
limit	否	Integer	每页显示的数量，非必填 最小值：**1** 最大值：**1000** 缺省值：**10**

请求参数

表 4-67 请求 Header 参数

参数	是否必选	参数类型	描述
Content-Type	否	String	设置媒体类型和编码格式
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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-68 响应 Body 参数

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

表 4-69 TaskCheckParamters

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

表 4-70 CheckConfigInfo

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

状态码： 400

表 4-71 响应 Body 参数

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

状态码： 401

表 4-72 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

```
{
  "total" : 1,
  "data" : [ {
    "check_id" : 0,
    "name" : "compile_tool",
    "check_configs" : [ {
      "value" : "",
      "name" : "编译命令",
      "cfg_key" : "",
      "default_value" : "",
      "option_value" : "",
      "is_required" : 0,
      "description" : "编译命令",
      "type" : 0,
      "status" : "on"
    } ]
  } ]
}
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

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

错误码

请参见错误码。

4.1.12 任务配置检查参数

功能介绍

任务配置检查参数

调用方法

请参见如何调用API。

URI

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

表 4-73 路径参数

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

请求参数

表 4-74 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**1** 最大长度：**100000**
Content-Type	是	String	设置媒体类型和编码格式

表 4-75 请求 Body 参数

参数	是否必选	参数类型	描述
check_id	是	Integer	检查工具ID 最小长度：**1** 最大长度：**11**
ruleset_id	是	String	规则集ID 最小长度：**1** 最大长度：**64**
language	是	String	规则集语言 最小长度：**1** 最大长度：**128**
status	是	String	off：关闭，on：开启 最小长度：**1** 最大长度：**3**
task_check_settings	是	Array of **TaskCheckSettingsItem** objects	检查参数信息

表 4-76 TaskCheckSettingsItem

参数	是否必选	参数类型	描述
cfg_key	是	String	检查参数对应的key值 最小长度：**1** 最大长度：**50**

参数	是否必选	参数类型	描述
status	是	String	参数状态 最小长度：**1** 最大长度：**3**
cfg_value	否	String	检查参数值 最小长度：**1** 最大长度：**1000**

响应参数

状态码： 400

表 4-77 响应 Body 参数

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

状态码： 401

表 4-78 响应 Body 参数

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

请求示例

任务配置检查参：检查参数的语言类型为csharp，状态为on

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

{
  "check_id" : 11,
  "ruleset_id" : "b113bxxxxxxxx5738bee86ecxxxxxxxx",
  "language" : "csharp",
  "status" : "on",
  "task_check_settings" : [ {
    "cfg_key" : "csharpCompileTool",
    "status" : "on",
    "cfg_value" : "msbuild"
  } ]
}
```

响应示例

状态码： 200

Request succeeded!

```
null
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

任务配置检查参：检查参数的语言类型为csharp，状态为on

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

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

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

public class ListTaskParameterSolution {

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

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

        CodeCheckClient client = CodeCheckClient.newBuilder()
                .withCredential(auth)
                .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
```

```
            .build();
        ListTaskParameterRequest request = new ListTaskParameterRequest();
        ConfigTaskParameterBody body = new ConfigTaskParameterBody();
        List<TaskCheckSettingsItem> listbodyTaskCheckSettings = new ArrayList<>();
        listbodyTaskCheckSettings.add(
            new TaskCheckSettingsItem()
                .withCfgKey("csharpCompileTool")
                .withStatus("on")
                .withCfgValue("msbuild")
        );
        body.withTaskCheckSettings(listbodyTaskCheckSettings);
        body.withStatus("on");
        body.withLanguage("csharp");
        body.withRulesetId("b113bxxxxxxxx5738bee86ecxxxxxxxx");
        body.withCheckId(11);
        request.withBody(body);
        try {
            ListTaskParameterResponse response = client.listTaskParameter(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

任务配置检查参：检查参数的语言类型为csharp，状态为on

```
# coding: utf-8

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

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ListTaskParameterRequest()
        listTaskCheckSettingsbody = [
            TaskCheckSettingsItem(
                cfg_key="csharpCompileTool",
                status="on",
                cfg_value="msbuild"
            )
        ]
```

```
        request.body = ConfigTaskParameterBody(
            task_check_settings=listTaskCheckSettingsbody,
            status="on",
            language="csharp",
            ruleset_id="b113bxxxxxxxx5738bee86ecxxxxxxxx",
            check_id=11
        )
        response = client.list_task_parameter(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

任务配置检查参：检查参数的语言类型为csharp，状态为on

```
package main

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

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

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

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

    request := &model.ListTaskParameterRequest{}
    cfgValueTaskCheckSettings:= "msbuild"
    var listTaskCheckSettingsbody = []model.TaskCheckSettingsItem{
        {
            CfgKey: "csharpCompileTool",
            Status: "on",
            CfgValue: &cfgValueTaskCheckSettings,
        },
    }
    request.Body = &model.ConfigTaskParameterBody{
        TaskCheckSettings: listTaskCheckSettingsbody,
        Status: "on",
        Language: "csharp",
        RulesetId: "b113bxxxxxxxx5738bee86ecxxxxxxxx",
        CheckId: int32(11),
    }
    response, err := client.ListTaskParameter(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
```

```
        fmt.Println(err)
    }
}
```

更多

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

状态码

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

错误码

请参见**错误码**。

4.1.13 修改任务规则集

功能介绍

修改任务规则集。

调用方法

请参见**如何调用API**。

URI

PUT /v2/tasks/{task_id}/ruleset

表 4-79 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	任务ID

请求参数

表 4-80 请求 Header 参数

参数	是否必选	参数类型	描述
Content-Type	是	String	设置媒体类型和编码格式

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）。 最小长度：**1** 最大长度：**100000**

表 4-81 请求 Body 参数

参数	是否必选	参数类型	描述
[数组元素]	是	Array of **UpdateTaskRulesetItem** objects	修改规则集信息

表 4-82 UpdateTaskRulesetItem

参数	是否必选	参数类型	描述
language	是	String	规则集语言 最小长度：**1** 最大长度：**128**
rule_set_id	是	String	规则集ID,通过调用ListTaskRuleset接口，根据响应参数中的template_id获得 最小长度：**1** 最大长度：**50**
if_use	是	String	任务语言和规则集的关系是否启用，1是启用，0是未启用 最小长度：**1** 最大长度：**1**
status	是	String	新/老数据表示，默认1 最小长度：**1** 最大长度：**1**

响应参数

状态码： 400

表 4-83 响应 Body 参数

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

状态码： 401

表 4-84 响应 Body 参数

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

请求示例

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

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

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

响应示例

状态码： 200

Request succeeded!

```
null
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

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

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

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

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

public class UpdateTaskRulesetSolution {

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

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

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

Python

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

```
# coding: utf-8

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

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateTaskRulesetRequest()
        listUpdateTaskRulesetbody = [
            UpdateTaskRulesetItem(
                language="cpp",
                rule_set_id="7be4cfxxxxxxxxfd8d07077cxxxxxxxx",
                if_use="1",
                status="1"
            )
        ]
        request.body = listUpdateTaskRulesetbody
        response = client.update_task_ruleset(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

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

```
package main

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

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

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

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

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

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

更多

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

状态码

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

错误码

请参见**错误码**。

4.1.14 查询任务检查失败日志

功能介绍

查询任务检查失败日志，不传execute_id则查询最近一次的检查日志

调用方法

请参见**如何调用API**。

URI

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

表 4-85 路径参数

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

表 4-86 Query 参数

参数	是否必选	参数类型	描述
execute_id	否	String	任务单次的执行ID

请求参数

表 4-87 请求 Header 参数

参数	是否必选	参数类型	描述
Content-Type	否	String	设置媒体类型和编码格式
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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-88 响应 Body 参数

参数	参数类型	描述
param_info	**ParamInfo** object	任务详情

参数	参数类型	描述
log_info	Array of **LogInfo** objects	日志信息

表 4-89 ParamInfo

参数	参数类型	描述
url	String	仓库地址
branch	String	仓库分支
language	String	仓库语言
exclude_dir	String	排除的目录
encode	String	编码格式
compile_config	String	编译配置信息
rule_template	String	规则集名称

表 4-90 LogInfo

参数	参数类型	描述
display_name	String	日志标题
log	String	日志内容
level	String	日志级别
analysis	String	日志分析
faq	String	常见问题解答

状态码： 400

表 4-91 响应 Body 参数

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

状态码： 401

表 4-92 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

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

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

状态码	描述
200	Request succeeded!

状态码	描述
400	Bad Request
401	Unauthorized

错误码

请参见错误码。

4.1.15 获取任务的目录树

功能介绍

获取任务的目录树

调用方法

请参见如何调用API。

URI

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

表 4-93 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目id
task_id	是	String	任务id

表 4-94 Query 参数

参数	是否必选	参数类型	描述
current_path	否	String	目录或文件的路径
offset	否	Integer	分页索引，偏移量 最小值：**0** 缺省值：**0**
limit	否	Integer	每页显示的数量,每页最多显示1000条 最小值：**1** 最大值：**1000** 缺省值：**10**

请求参数

无

响应参数

状态码： 200

表 4-95 响应 Body 参数

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

表 4-96 TreeNode

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

状态码： 400

表 4-97 响应 Body 参数

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

状态码： 401

表 4-98 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误描述

请求示例

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

响应示例

状态码： 200

Request succeeded!

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

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

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

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


public class ShowTaskPathTreeSolution {
```

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

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

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

Python

```
# coding: utf-8

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

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = ShowTaskPathTreeRequest()
        request.current_path = "<current_path>"
        request.offset = <offset>
        request.limit = <limit>
        response = client.show_task_path_tree(request)
```

```
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

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

    request := &model.ShowTaskPathTreeRequest{}
    currentPathRequest:= "<current_path>"
    request.CurrentPath = &currentPathRequest
    offsetRequest:= int32(<offset>)
    request.Offset = &offsetRequest
    limitRequest:= int32(<limit>)
    request.Limit = &limitRequest
    response, err := client.ShowTaskPathTree(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

状态码	描述
200	Request succeeded!

状态码	描述
400	Bad Request
401	Unauthorized

错误码

请参见错误码。

4.1.16 任务配置屏蔽目录

功能介绍

任务配置屏蔽目录

调用方法

请参见如何调用API。

URI

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

表 4-99 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
task_id	是	String	任务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	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）。 最小长度：**1** 最大长度：**100000**
Content-Type	是	String	设置媒体类型和编码格式

表 4-101 请求 Body 参数

参数	是否必选	参数类型	描述
ignore_path_settings	是	Array of **IgnorePathSettingItem** objects	屏蔽目录的节点信息列表

表 4-102 IgnorePathSettingItem

参数	是否必选	参数类型	描述
file_path	是	String	目录或文件路径
checkbox_status	是	String	屏蔽状态，包括unchecked(不屏蔽)、all(全屏蔽)、half(半屏蔽)

响应参数

状态码： 400

表 4-103 响应 Body 参数

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

状态码： 401

表 4-104 响应 Body 参数

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

请求示例

任务配置屏蔽目录：文件路径为.LAST_RELEASE，检查状态为all

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

{
  "ignore_path_settings" : [ {
```

```
    "file_path" : ".LAST_RELEASE",
    "checkbox_status" : "all"
  } ]
}
```

响应示例

状态码： 200

Request succeeded!

```
null
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

任务配置屏蔽目录：文件路径为.LAST_RELEASE，检查状态为all

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

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

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

public class UpdateIgnorePathSolution {

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

        ICredential auth = new BasicCredentials()
                .withAk(ak)
```

```
            .withSk(sk);

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

Python

任务配置屏蔽目录：文件路径为.LAST_RELEASE，检查状态为all

```
# coding: utf-8

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

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateIgnorePathRequest()
        listIgnorePathSettingsbody = [
            IgnorePathSettingItem(
                file_path=".LAST_RELEASE",
                checkbox_status="all"
            )
        ]
        request.body = UpdateIgnorePathRequestBody(
```

```
        ignore_path_settings=listIgnorePathSettingsbody
    )
    response = client.update_ignore_path(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

任务配置屏蔽目录：文件路径为.LAST_RELEASE，检查状态为all

```
package main

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

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

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

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

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

更多

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

状态码

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

错误码

请参见错误码。

4.1.17 查询任务的高级选项

功能介绍

查询任务的高级选项

调用方法

请参见如何调用API。

URI

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

表 4-105 路径参数

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

表 4-106 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页索引，偏移量，非必填 最小值：**0** 缺省值：**0**
limit	否	Integer	每页显示的数量，非必填 最小值：**1** 最大值：**1000** 缺省值：**10**

请求参数

表 4-107 请求 Header 参数

参数	是否必选	参数类型	描述
Content-Type	是	String	设置媒体类型和编码格式
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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-108 响应 Body 参数

参数	参数类型	描述
info	Array of **TaskAdvancedSettings** objects	高级选项的相关信息
total	Integer	总数

表 4-109 TaskAdvancedSettings

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

状态码： 400

表 4-110 响应 Body 参数

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

状态码： 401

表 4-111 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

```
{
  "info" : [ {
    "key" : "customImage",
    "value" : "xxxxx",
    "description" : "自定义镜像",
    "option_value" : ""
  } ],
  "total" : 1
}
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

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

错误码

请参见**错误码**。

4.1.18 任务配置高级选项

功能介绍

任务配置高级选项，如自定义镜像

调用方法

请参见**如何调用API**。

URI

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

表 4-112 路径参数

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

请求参数

表 4-113 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**1** 最大长度：**100000**
Content-Type	是	String	设置媒体类型和编码格式

表 4-114 请求 Body 参数

参数	是否必选	参数类型	描述
task_advanced_settings	是	Array of **TaskAdvancedSettingsItem** objects	高级选项参数的相关信息

表 4-115 TaskAdvancedSettingsItem

参数	是否必选	参数类型	描述
key	是	String	高级选项对应的名称 最小长度：**1** 最大长度：**50**
value	是	String	高级选项对应的取值 最小长度：**1** 最大长度：**1000**

响应参数

状态码： 400

表 4-116 响应 Body 参数

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

状态码： 401

表 4-117 响应 Body 参数

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

请求示例

任务配置高级选项，如自定义镜像等

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

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

响应示例

状态码： 200

Request succeeded!

```
null
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

任务配置高级选项，如自定义镜像等

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

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

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

public class UpdateTaskSettingsSolution {

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

```
        String sk = System.getenv("CLOUD_SDK_SK");

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

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

Python

任务配置高级选项，如自定义镜像等

```
# coding: utf-8

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

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateTaskSettingsRequest()
        listTaskAdvancedSettingsbody = [
            TaskAdvancedSettingsItem(
                key="customImage",
```

```
            value="xxxx"
        )
    ]
    request.body = UpdateTaskSettingsRequestBody(
        task_advanced_settings=listTaskAdvancedSettingsbody
    )
    response = client.update_task_settings(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

任务配置高级选项，如自定义镜像等

```
package main

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

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

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

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

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

更多

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

状态码

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

错误码

请参见**错误码**。

4.2 缺陷管理

4.2.1 查询缺陷概要

功能介绍

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

调用方法

请参见**如何调用API**。

URI

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

表 4-118 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	任务ID

请求参数

表 4-119 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-120 响应 Body 参数

参数	参数类型	描述
task_id	String	任务id
task_name	String	任务名字
creator_id	String	创建者id
git_url	String	代码仓地址
git_branch	String	代码仓分支,如果是MR模式，为源分支
last_check_time	String	上一次检查时间
code_line_total	Integer	代码总行数
code_line	Integer	代码有效行数
code_quality	Number	代码质量
issue_count	Integer	问题数
risk_coefficient	Number	危险系数
duplication_ratio	String	重复比例
complexity_count	Integer	复杂度
duplicated_lines	Integer	重复行数

参数	参数类型	描述
comment_lines	Integer	注释行数
comment_ratio	String	注释比例
duplicated_blocks	Integer	重复块
last_exec_time	String	上次执行时间
check_type	String	检查类型
created_at	String	创建时间
cyclomatic_complexity_per_method	String	代码平均复杂度
cyclomatic_complexity_per_file	String	代码平均复杂度(文件)
critical_count	String	致命问题数
major_count	String	严重问题数
minor_count	String	一般问题数
suggestion_count	String	提示问题数
is_access	String	门禁质量是否通过
trigger_type	String	任务触发方式
file_duplication_ratio	String	文件重复率
new_count	Integer	新增问题数
solve_count	Integer	已解决问题数
duplicated_files	Integer	重复文件数目
new_critical_count	String	新增致命问题数
new_major_count	String	新增严重问题数
new_minor_count	String	新增一般问题数

参数	参数类型	描述
new_suggestion_count	String	新增提示问题数

状态码： 400

表 4-121 响应 Body 参数

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

状态码： 401

表 4-122 响应 Body 参数

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

请求示例

```
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxxebedb6e57xxxxxxxx/defects-summary
```

响应示例

状态码： 200

Request succeeded!

```
{
  "task_id" : "435b58ecf7f54c45907c1384d01c94a6",
  "task_name" : "CSharp_02201119",
  "creator_id" : "ecadebb5041146cc96f9780f3d905e20",
  "git_url" : "git@****_02201119.git",
  "git_branch" : "master",
  "last_check_time" : "2020-02-20 11:37:46",
  "code_line_total" : 20,
  "code_line" : 156,
  "code_quality" : 147.4,
  "issue_count" : 3,
  "risk_coefficient" : 23,
  "duplication_ratio" : "0.0%",
  "complexity_count" : 0,
  "duplicated_lines" : 0,
  "comment_lines" : 0,
  "comment_ratio" : "0",
  "duplicated_blocks" : 0,
  "last_exec_time" : "2 分 28 秒",
```

```
  "check_type" : "source",
  "created_at" : "2020-02-20 11:21:11",
  "cyclomatic_complexity_per_method" : "84.0",
  "cyclomatic_complexity_per_file" : "112.0",
  "critical_count" : 2,
  "major_count" : 1,
  "minor_count" : 1,
  "suggestion_count" : 1,
  "is_access" : 0,
  "trigger_type" : 1,
  "file_duplication_ratio" : "33.3%",
  "new_count" : 0,
  "solve_count" : 0,
  "duplicated_files" : 0,
  "new_critical_count" : "0",
  "new_major_count" : "0",
  "new_minor_count" : "0",
  "new_suggestion_count" : "0"
}
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

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

错误码

请参见错误码。

4.2.2 查询 cmertrics 缺陷概要

功能介绍

根据检查任务ID查询cmertrics缺陷概要。

调用方法

请参见如何调用API。

URI

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

表 4-123 路径参数

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

请求参数

表 4-124 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-125 响应 Body 参数

参数	参数类型	描述
task_id	String	任务id
task_name	String	任务名字
creator_id	String	创建者id
git_url	String	代码仓地址
git_branch	String	代码仓分支
last_check_time	String	上一次检查时间
last_exec_time	String	上次执行时间
check_type	String	检查类型
created_at	String	创建时间

参数	参数类型	描述
metric_info	**MetricInfo** object	度量项

表 4-126 MetricInfo

参数	参数类型	描述
code_size	String	代码规模
raw_lines	String	原始代码行数
methods_total	String	函数总数
cyclomatic_complexity_total	String	圈复杂度总数
cyclomatic_complexity_per_method	String	平均圈复杂度
maximum_cyclomatic_complexity	String	最大圈复杂度
huge_cyclomatic_complexity_total	String	超大圈复杂度数
huge_cyclomatic_complexity_ratio	String	超大圈复杂度比例
cca_cyclomatic_complexity_total	String	cca圈复杂度总数
cca_cyclomatic_complexity_per_method	String	cca平均圈复杂度
maximum_cca_cyclomatic_complexity	String	cca最大圈复杂度
huge_cca_cyclomatic_complexity_total	String	超大圈复杂度函数总数
cyclomatic_complexity_adequacy	String	圈复杂度满足度

参数	参数类型	描述
maximum_depth	String	最大深度
huge_depth_total	String	超大深度数
huge_depth_ratio	String	超大深度占比
method_lines	String	函数总行数
lines_per_method	String	函数平均代码行
huge_method_total	String	超大函数数
huge_method_ratio	String	超大函数占比
files_total	String	文件总数
folders_total	String	目录总数
lines_per_file	String	文件平均代码行
huge_headerfile_total	String	超大头文件数
huge_headerfile_ratio	String	超大头文件占比
huge_non_headerfile_total	String	超大源文件数
huge_non_headerfile_ratio	String	超大源文件占比
huge_folder_total	String	超大目录数
huge_folder_ratio	String	超大目录占比
file_duplication_total	String	重复文件数
file_duplication_ratio	String	文件重复率
non_hfile_duplication_total	String	重复源文件数
non_hfile_duplication_ratio	String	源文件重复率

参数	参数类型	描述
code_duplication_total	String	代码重复数
code_duplication_ratio	String	代码重复率
non_hfile_code_duplication_total	String	源文件代码重复数
non_hfile_code_duplication_ratio	String	源文件代码重复率
unsafe_functions_total	String	危险函数总数
unsafe_functions_kloc	String	危险函数密度
redundant_code_total	String	冗余代码数
redundant_code_kloc	String	冗余代码块密度
warning_suppression_total	String	抑制告警数
warning_suppression_kloc	String	抑制告警密度

状态码： 400

表 4-127 响应 Body 参数

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

状态码： 401

表 4-128 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误描述

请求示例

```
GET https://{endpoint}/v2/{project_id}/tasks/{task_id}/metrics-summary
```

响应示例

状态码： 200

Request succeeded!


```
{
  "task_id" : "435b58ecf7f54c45907c1384d01c94a6",
  "task_name" : "CSharp_02201119",
  "creator_id" : "ecadebb5041146cc96f9780f3d905e20",
  "git_url" : "git@****_02201119.git",
  "git_branch" : "master",
  "last_check_time" : "2020-02-20 11:37:46",
  "last_exec_time" : "2020-02-20 11:37:46",
  "check_type" : "source",
  "created_at" : "2020-02-20 11:21:11",
  "metric_info" : {
    "code_size" : "133426.00",
    "raw_lines" : "182082.00",
    "methods_total" : "6840.00",
    "cyclomatic_complexity_total" : "27435.00",
    "cyclomatic_complexity_per_method" : "7.43",
    "maximum_cyclomatic_complexity" : "346.00",
    "huge_cyclomatic_complexity_total" : "167.00",
    "huge_cyclomatic_complexity_ratio" : "4.13",
    "cca_cyclomatic_complexity_total" : "26693.00",
    "cca_cyclomatic_complexity_per_method" : "7.25",
    "maximum_cca_cyclomatic_complexity" : "342.00",
    "huge_cca_cyclomatic_complexity_total" : "148.00",
    "cyclomatic_complexity_adequacy" : "21.00",
    "maximum_depth" : "333.00",
    "huge_depth_total" : "8.85",
    "huge_depth_ratio" : "91546.00",
    "method_lines" : "24.91",
    "lines_per_method" : "8.04",
    "huge_method_total" : "12300",
    "huge_method_ratio" : "12300",
    "files_total" : "629.00",
    "folders_total" : "426.00",
    "lines_per_file" : "12300",
    "huge_headerfile_total" : "12300",
    "huge_headerfile_ratio" : "12300",
    "huge_non_headerfile_total" : "12300",
    "huge_non_headerfile_ratio" : "12300",
    "huge_folder_total" : "12300",
    "huge_folder_ratio" : "12300",
    "file_duplication_total" : "12300",
    "file_duplication_ratio" : "12300",
    "non_hfile_duplication_total" : "12300",
    "non_hfile_duplication_ratio" : "12300",
    "code_duplication_total" : "3.28",
    "code_duplication_ratio" : "11782.00",
    "non_hfile_code_duplication_total" : "8.59",
    "non_hfile_code_duplication_ratio" : "11782.00",
    "unsafe_functions_total" : "17.17",
    "unsafe_functions_kloc" : "0.00",
    "redundant_code_total" : "0.00",
```

```
    "redundant_code_kloc" : "674.00",
    "warning_suppression_total" : "17.00",
    "warning_suppression_kloc" : "0.45"
  }
}
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

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

错误码

请参见**错误码**。

4.2.3 查询缺陷详情

功能介绍

根据检查任务ID分页查询缺陷结果详情。

调用方法

请参见**如何调用API**。

URI

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

表 4-129 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	任务ID

表 4-130 Query 参数

参数	是否必选	参数类型	描述
offset	否	Integer	分页索引，偏移量 最小值：**0** 缺省值：**0**
limit	否	Integer	每页显示的数量,每页最多显示100条 最小值：**1** 最大值：**100** 缺省值：**10**
status_ids	否	String	问题状态筛选
severity	否	String	严重级别，0致命，1严重，2一般，3提示

请求参数

表 4-131 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-132 响应 Body 参数

参数	参数类型	描述
defects	Array of **DefectInfoV2** objects	缺陷详情信息
total	Integer	总数

表 4-133 DefectInfoV2

参数	参数类型	描述
defect_id	String	缺陷的id
defect_checker_name	String	缺陷对应检查项的名称
defect_status	String	缺陷的状态0为解决 1已解决 2已忽略
rule_system_tags	String	规则标签,多个标签用逗号隔开
rule_id	String	规则id
rule_name	String	规则名
line_number	String	缺陷所在文件行号
defect_content	String	缺陷描述
defect_level	String	缺陷等级，0致命，1严重，2一般，3提示
file_path	String	缺陷文件路径
created_at	String	创建时间
issue_key	String	问题唯一标识
fragment	Array of **DefectFragmentV2** objects	缺陷代码片段详情
events	Array of **DefectEvents** objects	调用链信息

表 4-134 DefectFragmentV2

参数	参数类型	描述
line_num	String	行号

参数	参数类型	描述
line_content	String	该行代码内容
start_offset	Integer	缺陷开始列号
end_offset	Integer	缺陷结束列号

表 4-135 DefectEvents

参数	参数类型	描述
events	Array of **DefectEvents** objects	调用链信息
description	String	描述
fix_suggestions	Array of strings	修复建议
line	Integer	缺陷所在行
end_line	Integer	缺陷结束行
code_context_start_line	Integer	缺陷开始行
main	Boolean	代码片段
path	String	路径
tag	String	标签
main_buggy_code	String	主事件对应的代码片段
code_context	String	代码上下文

状态码： 400

表 4-136 响应 Body 参数

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

状态码： 401

表 4-137 响应 Body 参数

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

请求示例

```
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxxebedb6e57xxxxxxxx/defects-summary
```

响应示例

状态码： 200

Request succeeded!


```
{
  "defects" : [ {
    "line_number" : "81",
    "defect_id" : "5b71a4594bcb4a69a33f576c3564e6cc",
    "rule_system_tags" : "fossscan",
    "rule_id" : "111",
    "rule_name" : "FossScan.1 OpenSource Software",
    "defect_checker_name" : "Disallow fallthrough of `case` statements",
    "defect_status" : "0",
    "defect_content" : "Update or refactor this function so that its implementation doesn't duplicate the one
on line 60.",
    "defect_level" : "1",
    "file_path" : "code/types/functions.ts",
    "created_at" : "2020-07-21T02:12:01Z",
    "issue_key" : "AA76E96XXXXXXXXXXXXXXXXXCAAE5",
    "fragment" : [ {
      "line_num" : "78",
      "line_content" : "    export function padding(all: number);",
      "start_offset" : -1,
      "end_offset" : -1
    }, {
      "line_num" : "79",
      "line_content" : "    export function padding(topAndBottom: number, leftAndRight: number);",
      "start_offset" : -1,
      "end_offset" : -1
    }, {
      "line_num" : "80",
      "line_content" : "    export function padding(top: number, right: number, bottom: number, left:
number);",
      "start_offset" : -1,
      "end_offset" : -1
    }, {
      "line_num" : "81",
      "line_content" : "    export function padding(a: number, b?: number, c?: number, d?: number) {",
      "start_offset" : 11,
      "end_offset" : 19
    }, {
      "line_num" : "82",
      "line_content" : "        if (b === undefined && c === undefined && d === undefined) {",
      "start_offset" : -1,
      "end_offset" : -1
    } ]
  }, {
    "line_number" : "190",
    "defect_id" : "5b71a4594wtrfsgrew546t466c3564e6cc",
    "rule_system_tags" : "fossscan",
    "rule_id" : "112",
```

```
    "rule_name" : "FossScan.1 OpenSource Software",
    "defect_checker_name" : "Disallow fallthrough of `case` statements",
    "defect_status" : "0",
    "defect_content" : "Update or refactor this function so that its implementation doesn't duplicate the one
on line 173.",
    "defect_level" : "1",
    "file_path" : "code/types/types.ts",
    "created_at" : "2020-07-21T02:12:01Z",
    "issue_key" : "AA76E96XXXXXXXXXXXXXXXXXFE3B6",
    "fragment" : [ {
      "line_num" : "187",
      "line_content" : "    var _value;",
      "start_offset" : -1,
      "end_offset" : -1
    }, {
      "line_num" : "188",
      "line_content" : "    function getOrSet(): number;",
      "start_offset" : -1,
      "end_offset" : -1
    }, {
      "line_num" : "189",
      "line_content" : "    function getOrSet(value: number);",
      "start_offset" : -1,
      "end_offset" : -1
    }, {
      "line_num" : "190",
      "line_content" : " \"    function getOrSet(value?: number) {\"",
      "start_offset" : 4,
      "end_offset" : 12
    }, {
      "line_num" : "191",
      "line_content" : " \"        if (value === undefined) {\"",
      "start_offset" : -1,
      "end_offset" : -1
    } ]
  } ],
  "total" : 2
}
```


状态码： 400

Bad Request


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


状态码： 401

Unauthorized


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


状态码

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

错误码

请参见错误码。

4.2.4 查询缺陷详情的统计

功能介绍

根据检查任务ID查询缺陷详情的统计

调用方法

请参见如何调用API。

URI

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

表 4-138 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	任务ID

请求参数

表 4-139 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-140 响应 Body 参数

参数	参数类型	描述
severity	**StatisticSeverityV2** object	问题级别统计

参数	参数类型	描述
status	StatisticStatusV2 object	问题状态统计

表 4-141 StatisticSeverityV2

参数	参数类型	描述
critical	Integer	致命问题数
major	Integer	严重问题数
minor	Integer	一般问题数
suggestion	Integer	提示问题数

表 4-142 StatisticStatusV2

参数	参数类型	描述
unresolved	Integer	未解决
resolved	Integer	已解决
dismissed	Integer	已忽略

状态码： 400

表 4-143 响应 Body 参数

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

状态码： 401

表 4-144 响应 Body 参数

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

请求示例

```
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxxebedb6e57xxxxxxxx/defects-statistic
```

响应示例

状态码： 200

Request succeeded!

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

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

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

错误码

请参见错误码。

4.2.5 修改缺陷状态

功能介绍

修改检查出的缺陷的状态为已解决、已忽略

调用方法

请参见**如何调用API**。

URI

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

表 4-145 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	任务id

请求参数

表 4-146 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**1** 最大长度：**100000**
Content-Type	是	String	设置媒体类型和编码格式

表 4-147 请求 Body 参数

参数	是否必选	参数类型	描述
defect_id	是	String	问题id,多个时英文逗号分隔
defect_status	是	String	状态2：已忽略 1：已解决 0：未解决

响应参数

状态码： 400

表 4-148 响应 Body 参数

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

状态码： 401

表 4-149 响应 Body 参数

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

请求示例

修改任务缺陷的状态：缺陷的状态ID根据实际显示，缺陷的状态修改为1

```
PUT https://{endpoint}/v2/tasks/{taskid}/defect-status

{
  "defect_id" : "363540xxxxxxxx5105099944xxxxxxxx",
  "defect_status" : "1"
}
```

响应示例

状态码： 200

Request succeeded!

```
null
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

修改任务缺陷的状态：缺陷的状态ID根据实际显示，缺陷的状态修改为1

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

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


public class UpdateDefectStatusSolution {

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

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

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

Python

修改任务缺陷的状态：缺陷的状态ID根据实际显示，缺陷的状态修改为1

```
# coding: utf-8

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

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

```
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = UpdateDefectStatusRequest()
        request.body = UpdateDefectRequestBody(
            defect_status="1",
            defect_id="363540xxxxxxxx5105099944xxxxxxxx"
        )
        response = client.update_defect_status(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

修改任务缺陷的状态：缺陷的状态ID根据实际显示，缺陷的状态修改为1

```
package main

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

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

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

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

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

```
  }
}
```

更多

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

状态码

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

错误码

请参见**错误码**。

4.3 规则管理

4.3.1 获取规则列表接口

功能介绍

根据语言、问题级别等条件查询规则列表。

调用方法

请参见**如何调用API**。

URI

GET /v2/rules

表 4-150 Query 参数

参数	是否必选	参数类型	描述
rule_languages	否	String	规则对应的语言
rule_severity	否	String	缺陷等级，0致命，1严重，2一般，3提示

参数	是否必选	参数类型	描述
offset	否	Integer	分页索引，偏移量 最小值：**0** 缺省值：**0**
limit	否	Integer	每页显示的数量 最小值：**1** 最大值：**1000** 缺省值：**100**

请求参数

表 4-151 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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-152 响应 Body 参数

参数	参数类型	描述
info	Array of **RuleListItem** objects	规则信息
total	Integer	总数

表 4-153 RuleListItem

参数	参数类型	描述
rule_set	String	规则集规范分类
rule_id	String	规则id

参数	参数类型	描述
rule_language	String	规则所属语言
rule_name	String	规则名称
rule_severity	String	规则问题级别
rule_tages	String	规则标签
right_example	String	正确示例
error_example	String	错误示例
revise_opinion	String	修改建议
rule_desc	String	规则描述

状态码： 400

表 4-154 响应 Body 参数

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

状态码： 401

表 4-155 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

```
{
  "info" : [ {
    "rule_id" : "2561",
    "rule_language" : "Java",
    "rule_name" : "\"=+\"不能用\"+=\"代替",
```

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


状态码： 400

Bad Request


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


状态码： 401

Unauthorized


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


状态码

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

错误码

请参见错误码。

4.3.2 创建自定义规则集

功能介绍

可根据需求灵活的组合规则。

调用方法

请参见**如何调用API**。

URI

POST /v2/ruleset

请求参数

表 4-156 请求 Header 参数

参数	是否必选	参数类型	描述
Content-Type	是	String	设置媒体类型和编码格式
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）。 最小长度：**1** 最大长度：**100000**

表 4-157 请求 Body 参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID 最小长度：**1** 最大长度：**50**
template_name	是	String	新规则集名称 最小长度：**1** 最大长度：**500**
language	是	String	规则集语言 最小长度：**1** 最大长度：**128**

参数	是否必选	参数类型	描述
is_default	是	String	如果有基于的规则集则是1，没有基于的规则集则是0 最小长度：**1** 最大长度：**1**
rule_ids	是	String	新启用规则ids 最小长度：**1** 最大长度：**4000**
uncheck_ids	否	String	新关闭规则id 最小长度：**1** 最大长度：**4000**
template_id	否	String	规则集ID 最小长度：**1** 最大长度：**50**
custom_attributes	否	Array of **CustomAttributes** objects	自定义规则参数项，支持修改规则阈值

表 4-158 CustomAttributes

参数	是否必选	参数类型	描述
attribute	否	String	配置项属性，severity：为问题级别
rules	否	Array of **CustomAttributesRule** objects	规则详细

表 4-159 CustomAttributesRule

参数	是否必选	参数类型	描述
rule_id	否	String	规则ID
value	否	String	attribute的问题级别，0致命，1严重，2一般，3提示
rule_config_list	否	Array of **RuleConfig** objects	规则阈值详细

表 4-160 RuleConfig

参数	是否必选	参数类型	描述
id	否	Integer	规则配置ID
rule_id	否	Integer	规则ID
default_value	否	String	默认值
option_value	否	String	当前
option_key	否	String	当前规则配置项key
option_name	否	String	当前规则配置项名称
template_id	否	String	规则集id
description	否	String	描述

响应参数

状态码： 201

表 4-161 响应 Body 参数

参数	参数类型	描述
template_id	String	规则集ID

状态码： 400

表 4-162 响应 Body 参数

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

状态码： 401

表 4-163 响应 Body 参数

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

请求示例

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

```
POST https://{endpoint}/v2/ruleset

{
  "project_id" : "24b97exxxxxxxxfb912625b14cxxxxxx",
  "template_id" : "da303dxxxxxxxxaca60dbcc2e2xxxxxx",
  "template_name" : "test_cpp",
  "language" : "cpp",
  "is_default" : "1",
  "rule_ids" : "8139,8138",
  "uncheck_ids" : "1101,1102"
}
```

响应示例

状态码： 201

Request succeeded!

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

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

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

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

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

```
public class CreateRulesetSolution {

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

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

        CodeCheckClient client = CodeCheckClient.newBuilder()
                .withCredential(auth)
                .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
                .build();
        CreateRulesetRequest request = new CreateRulesetRequest();
        Ruleset body = new Ruleset();
        body.withTemplateId("da303dxxxxxxxxaca60dbcc2e2xxxxxx");
        body.withUncheckIds("1101,1102");
        body.withRuleIds("8139,8138");
        body.withIsDefault("1");
        body.withLanguage("cpp");
        body.withTemplateName("test_cpp");
        body.withProjectId("24b97exxxxxxxxfb912625b14cxxxxxx");
        request.withBody(body);
        try {
            CreateRulesetResponse response = client.createRuleset(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

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

```
# coding: utf-8

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

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

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

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

try:
    request = CreateRulesetRequest()
    request.body = Ruleset(
        template_id="da303dxxxxxxxxaca60dbcc2e2xxxxxx",
        uncheck_ids="1101,1102",
        rule_ids="8139,8138",
        is_default="1",
        language="cpp",
        template_name="test_cpp",
        project_id="24b97exxxxxxxxfb912625b14cxxxxxx"
    )
    response = client.create_ruleset(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

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

```
package main

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

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

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

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

    request := &model.CreateRulesetRequest{}
    templateIdRuleset:= "da303dxxxxxxxxaca60dbcc2e2xxxxxx"
    uncheckIdsRuleset:= "1101,1102"
    request.Body = &model.Ruleset{
        TemplateId: &templateIdRuleset,
        UncheckIds: &uncheckIdsRuleset,
        RuleIds: "8139,8138",
        IsDefault: "1",
        Language: "cpp",
```

```
        TemplateName: "test_cpp",
        ProjectId: "24b97exxxxxxxxfb912625b14cxxxxxx",
    }
    response, err := client.CreateRuleset(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

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

错误码

请参见**错误码**。

4.3.3 查询规则集列表

功能介绍

根据项目ID、语言等条件查询规则集列表。

调用方法

请参见**如何调用API**。

URI

GET /v2/{project_id}/rulesets

表 4-164 路径参数

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

表 4-165 Query 参数

参数	是否必选	参数类型	描述
category	否	String	规则集类别 '0只查询系统规则集，1只查询当前用户自定义规则集，2只查询其他用户自定义规则集，'0,1,2'或''查所有'
offset	否	Integer	分页索引，偏移量 最小值：**0** 缺省值：**0**
limit	否	Integer	每页显示的数量 最小值：**1** 最大值：**1000** 缺省值：**100**

请求参数

表 4-166 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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-167 响应 Body 参数

参数	参数类型	描述
info	Array of **RulesetItem** objects	规则集列表信息
total	Integer	总数

表 4-168 RulesetItem

参数	参数类型	描述
template_id	String	规则集id
language	String	规则集语言
template_name	String	规则集名称
creator_id	String	创建者id
creator_name	String	创建人名称
template_create_time	String	创建人时间
is_used	String	使用状态1使用中，0空闲中
rule_ids	String	规则集包含的规则id
is_default	String	是否该语言默认规则集，0不是，1是
is_devcloud_project_default	String	是否是项目下语言默认规则集，0不是，1是
is_default_template	String	是否是系统规则集，0不是，1是

状态码： 400

表 4-169 响应 Body 参数

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

状态码： 401

表 4-170 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!


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


状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

状态码

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

错误码

请参见错误码。

4.3.4 查看规则集的规则列表

功能介绍

根据项目ID、规则集ID等条件查询规则列表。

调用方法

请参见如何调用API。

URI

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

表 4-171 路径参数

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

表 4-172 Query 参数

参数	是否必选	参数类型	描述
types	是	String	规则状态 '1查询全部，2已启动，3未启用' 缺省值：**1**
languages	否	String	规则语言
tags	否	String	规则所属标签
offset	否	Integer	分页索引，偏移量 最小值：**0** 缺省值：**0**
limit	否	Integer	每页显示的数量 最小值：**1** 最大值：**1000** 缺省值：**100**

请求参数

表 4-173 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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 200

表 4-174 响应 Body 参数

参数	参数类型	描述
info	Array of **RuleItem** objects	规则集的规则列表信息
total	Integer	总数

表 4-175 RuleItem

参数	参数类型	描述
rule_id	String	规则id
rule_language	String	规则所属语言
rule_name	String	规则名称
rule_severity	String	规则问题级别
tags	String	规则标签
checked	String	规则状态0：未启用，1：已启用
rule_config_list	Array of **RuleConfig** objects	规则配置参数阈值相关信息

表 4-176 RuleConfig

参数	参数类型	描述
id	Integer	规则配置ID
rule_id	Integer	规则ID
default_value	String	默认值
option_value	String	当前
option_key	String	当前规则配置项key
option_name	String	当前规则配置项名称
template_id	String	规则集id
description	String	描述

状态码： 400

表 4-177 响应 Body 参数

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

状态码： 401

表 4-178 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

```
{
  "info" : [ {
    "rule_language" : "cpp",
```

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


状态码： 400

Bad Request


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


状态码： 401

Unauthorized


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


状态码

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

错误码

请参见错误码。

4.3.5 删除自定义规则集

功能介绍

删除自定义规则集，正在使用中的或默认规则集不能删除

调用方法

请参见如何调用API。

URI

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

表 4-179 路径参数

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

请求参数

表 4-180 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）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 400

表 4-181 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误描述

状态码： 401

表 4-182 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

```
null
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

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

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

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


public class DeleteRulesetSolution {

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

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

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

Python

```
# coding: utf-8

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

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = DeleteRulesetRequest()
```

```
        response = client.delete_ruleset(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

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

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

更多

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

状态码

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

错误码

请参见错误码。

4.3.6 设置每个项目对应语言的默认规则集配置

功能介绍

设置每个项目对应语言的默认规则集配置。

调用方法

请参见如何调用API。

URI

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

表 4-183 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	项目ID
ruleset_id	是	String	规则集ID
language	是	String	规则集语言

请求参数

表 4-184 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 最小长度：**1** 最大长度：**100000**

响应参数

状态码： 400

表 4-185 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误描述

状态码： 401

表 4-186 响应 Body 参数

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

请求示例

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

响应示例

状态码： 200

Request succeeded!

```
null
```

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

SDK 代码示例

SDK代码示例如下。

Java

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

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

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


public class SetDefaulTemplateSolution {

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

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

        CodeCheckClient client = CodeCheckClient.newBuilder()
                .withCredential(auth)
                .withRegion(CodeCheckRegion.valueOf("<YOUR REGION>"))
                .build();
        SetDefaulTemplateRequest request = new SetDefaulTemplateRequest();
        try {
            SetDefaulTemplateResponse response = client.setDefaulTemplate(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

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

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

    credentials = BasicCredentials(ak, sk) \

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

    try:
        request = SetDefaulTemplateRequest()
```

```
        response = client.set_defaul_template(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

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

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

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

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

    request := &model.SetDefaulTemplateRequest{}
    response, err := client.SetDefaulTemplate(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

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

状态码

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

错误码

请参见**错误码**。

5 应用示例

5.1 示例 1：查询任务列表

场景描述

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

约束限制

无。

涉及接口

涉及的接口如下：

查询任务列表：根据DEVCLOUD_PROJECT_UUID查询任务列表。

操作步骤

查询任务列表。

- 接口相关信息
  URI格式：GET /v2/{project_id}/tasks
- 请求示例
  ```
  GET https://{endpoint}/v2/2b31ed520xxxxxxebedb6e57xxxxxxxx/tasks
  ```
  信息请从**终端节点**获取。
- 响应示例
  ```
  {
  "result":[
  {
  "projectId":"b9db371361724226937f3280df1d4fc4",
  "devCloudProjectId":"a2b9a42ee6494e0794f5655f8535920e",
  "devCloudProjectName":"TestDemo",
  "codeHubName":"portal-ts",
  "codeHubWebUrl":"https://xxxxx/portal-ts",
  "codeLineTotal":"63371",
  "projectName":"portal-ts-abctest111222",
  "creatorId":"ae161856f2604229ae12056478a7919f",
  ```


版权所有 © 华为技术有限公司

"creatorName":"name",
"domainId":"78d0e09c1ad0425b9b4f3a8fdd066164",
"domainName":"name",
"gitUrl":"git@xxxxx/portal-ts.git",
"branch":"master",
"language":"[
\"html\"
]",
"filePath":"[
\"\"
]",
"lastCheckTime\"":"2019-09-24 15:00:41",
"codeQuality":"141.7",
"codeLine":"55777",
"projectStatus":"2",
"riskCoefficient":"7905",
"starLevel":"0",
"defectInfo":{
"totalCount":"0",
"criticalCount":"0",
"seriousCount":"0",
"normalCount":"0",
"promptCount":"0"
},
"defectStatusInfo":{
"totalCount":"2952",
"newCount":"2952",
"ignoredCount":"0",
"fixedCount":"0"
},
"progress":{

},
"createdAt":"2019-08-08 17:56:40",
"duplications":"24.0%",
"complexityNum":"0",
"complexityAvg":"0",
"duplicatedLines":"15206",
"commentLines":"0",
"commentRatio":"0",
"duplicatedBlocks":"0",
"lastExecTime":"2 分 4 秒",
"checkType":"source",
"dependenceType":"",
"logDatasStr":{

}
},
{
"projectId":"b9db371361724226937f3280df1d4fc4",
"devCloudProjectId":"a2b9a42ee6494e0794f5655f8535920e",
"devCloudProjectName":"TestDemo",
"codeHubName":"portal-ts",
"codeHubWebUrl":"https://xxxxx/portal-ts",
"codeLineTotal":"63371",
"projectName":"portal-ts-abctest111222",
"creatorId":"ae161856f2604229ae12056478a7919f",
"creatorName":"devcloud_devcloud_y00336947_01",
"domainId":"78d0e09c1ad0425b9b4f3a8fdd066164",
"domainName":"name",
"gitUrl":"git@xxxxx/portal-ts.git",
"branch":"master",
"language":"[
\"html\"
]",
"filePath":"[
\"\"
]",
"lastCheckTime\"":"2019-09-24 15:00:41",

```
"codeQuality":"141.7",
"codeLine":"55777",
"projectStatus":"2",
"riskCoefficient":"7905",
"starLevel":"0",
"defectInfo":{
"totalCount":"0",
"criticalCount":"0",
"seriousCount":"0",
"normalCount":"0",
"promptCount":"0"
},
"defectStatusInfo":{
"totalCount":"2952",
"newCount":"2952",
"ignoredCount":"0",
"fixedCount":"0"
},
"progress":{

},
"createdAt":"2019-08-08 17:56:40",
"duplications":"24.0%",
"complexityNum":"0",
"complexityAvg":"0",
"duplicatedLines":"15206",
"commentLines":"0",
"commentRatio":"0",
"duplicatedBlocks":"0",
"lastExecTime":"2 分 4 秒",
"checkType":"source",
"dependenceType":"",
"logDatasStr":{

}
}
],
"status":"success"
}
```


5.2 示例 2：查询缺陷概要

场景描述

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

约束限制

任务存在。

涉及接口

涉及的接口如下：

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

操作步骤

- 接口相关信息

  URI格式：GET /v2/tasks/{task_id}/defects-summary

版权所有 © 华为技术有限公司

- 请求示例

```
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxxebedb6e57xxxxxxxx/defects-summary
```

信息请从**终端节点**获取。

- 响应示例


```
{
"result":{
"info":{
"projectId":"b9db371361724226937f3280df1d4fc4",
"devCloudProjectId":"a2b9a42ee6494e0794f5655f8535920e",
"devCloudProjectName":"TestDemo",
"codeHubName":"portal-ts",
"codeHubWebUrl":"https://xxxxx/portal-ts",
"codeLineTotal":"63371",
"projectName":"portal-ts-abctest111222",
"creatorId":"ae161856f2604229ae12056478a7919f",
"creatorName":"name",
"domainId":"78d0e09c1ad0425b9b4f3a8fdd066164",
"domainName":"name",
"gitUrl":"git@xxxxx/portal-ts.git",
"branch":"master",
"language":"[
\"html\"
]",
"filePath":"[
\"\"
]",
"lastCheckTime\"":"2019-09-24 15:00:41",
"codeQuality":"141.7",
"codeLine":"55777",
"projectStatus":"2",
"riskCoefficient":"790",
"starLevel":"0",
"defectInfo":{
"totalCount":"0",
"criticalCount":"0",
"seriousCount":"0",
"normalCount":"0",
"promptCount":"0"
},
"defectStatusInfo":{
"totalCount":"2952",
"newCount":"2952",
"ignoredCount":"0",
"fixedCount":"0"
},
"progress":{

},
"createdAt":"2019-08-08 17:56:40",
"duplications":"24.0%",
"complexityNum":"0",
"complexityAvg":"0",
"duplicatedLines":"15206",
"commentLines":"0",
"commentRatio":"0",
"duplicatedBlocks":"0",
"lastExecTime":"2 分 4 秒",
"checkType":"source",
"dependenceType":"",
"logDatasStr":{

}
}
},
"status":"success"
}
```


版权所有 © 华为技术有限公司

6 附录

6.1 状态码

状态码如表6-1所示。

表 6-1 状态码

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

版权所有 © 华为技术有限公司

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

版权所有 © 华为技术有限公司

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

6.2 错误码

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

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

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

错误响应 Body 体格式说明

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

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

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

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

错误码说明

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

更多服务错误码请参见**API错误中心**。

表 6-2 错误码

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

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

6.3 获取项目 ID

调用 API 获取项目 ID

项目ID还用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": "cn-north-4",


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

   说明

   多项目时，展开“所属区域”，从“项目ID”列获取子项目ID。

6.4 获取账号 ID

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

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

7 修订记录

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

修订记录	发布日期
第二十四次正式发布。本次变更说明如下： ● 修改**错误码**。	2023-01-09
第二十三次正式发布。本次变更说明如下： ● 增加**获取任务的目录树**。 ● 增加**任务配置屏蔽目录**。 ● 增加**查询任务的高级选项**。 ● 增加**任务配置高级选项**。	2022-12-26
第二十二次正式发布。本次变更说明如下： ● 修改**查询任务的已选规则集列表v3**。 ● 修改**查询任务规则集的检查参数v3**。 ● 修改**错误码**。	2022-09-22
第二十一次正式发布。本次变更说明如下： ● 修改**错误码**。	2022-09-06
第二十次正式发布。本次变更说明如下： ● 修改**示例1：查询任务列表**。 ● 修改**示例2：查询缺陷概要**。	2022-08-29
第十九次正式发布。本次变更说明如下： ● 修改**查询缺陷概要**。 ● 修改**获取规则列表接口**。	2022-06-30
第十八次正式发布。本次变更说明如下： ● 增加**历史扫描结果查询**。	2022-02-26

版权所有 © 华为技术有限公司

修订记录	发布日期
第十七次正式发布。本次变更说明如下： ● 增加查询任务检查失败日志。 ● 增加删除自定义规则集。 ● 增加设置每个项目对应语言的默认规则集配置。 ● 修改新建检查任务。 ● 修改查询缺陷概要。 ● 修改创建自定义规则集。 ● 修改查看规则集的规则列表。 ● 修改错误码。	2022-01-27
第十六次正式发布。本次变更说明如下： ● 修改错误码。	2022-01-04
第十五次正式发布。本次变更说明如下： ● 增加任务配置检查参数。 ● 增加查询cmertrics缺陷概要。 ● 增加查看规则集的规则列表。	2021-12-27
第十四次正式发布。本次变更说明如下： ● 修改查询缺陷概要：增加返回值is_access和trigger_type。	2021-12-15
第十三次正式发布。本次变更说明如下： ● 增加规则管理。	2021-11-27
第十二次正式发布。本次变更说明如下： ● 增加删除检查任务。 ● 增加查询缺陷详情的统计。 ● 增加修改缺陷状态。	2021-09-29
第十一次正式发布。本次变更说明如下： ● 修改错误码。	2021-05-08
第十次正式发布。本次变更说明如下： ● 删除约束与限制。	2021-03-26
第九次正式发布。本次变更说明如下： ● 终端节点中增加“华东-上海二”节点。	2021-02-05
第八次正式发布。本次变更说明如下： ● 修改任务管理。 ● 修改缺陷管理。	2020-11-23

版权所有 © 华为技术有限公司

修订记录	发布日期
第七次正式发布。本次变更说明如下： ● 修改**示例1：查询任务列表**。 ● 修改**示例2：查询缺陷概要**。	2020-11-05
第六次正式发布。本次变更说明如下： ● 修改**构造请求**。 ● 修改**认证鉴权**。	2020-08-06
第五次正式发布。本次变更说明如下： 增加应用示例。 ● **示例1：查询任务列表**。 ● **示例2：查询缺陷概要**。	2020-07-30
第四次正式发布。本次变更说明如下： ● 修改“请求示例”中URL地址信息。 ● 修改**终端节点**。	2020-07-01
第三次正式发布。本次变更说明如下： 修改代码示例中代码仓库地址信息。	2020-03-13
第二次正式发布。本次变更说明如下： 代码检查API接口项增加变更说明，如有疑问，请在帮助页面下方提交反馈。	2020-01-03
第一次正式发布。	2019-12-28

版权所有 © 华为技术有限公司