



代码检查

## API 参考

文档版本 08

发布日期 2020-11-23

版权所有 © 华为技术有限公司 2020。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默示的声明或保证。

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

---

# 目录

---

<b>1 使用前必读.....</b>	<b>1</b>
1.1 概述.....	1
1.2 调用说明.....	1
1.3 终端节点.....	1
1.4 约束与限制.....	2
1.5 基本概念.....	2
<b>2 API 概览.....</b>	<b>3</b>
<b>3 如何调用 API.....</b>	<b>4</b>
3.1 构造请求.....	4
3.2 认证鉴权.....	7
3.3 返回结果.....	8
<b>4 API.....</b>	<b>10</b>
4.1 缺陷管理.....	10
4.1.1 查询缺陷概要.....	10
4.1.2 查询缺陷详情.....	13
4.2 任务管理.....	17
4.2.1 新建检查任务.....	17
4.2.2 查询任务列表.....	19
4.2.3 执行检查任务.....	22
4.2.4 终止检查任务.....	24
4.2.5 查询任务执行状态.....	25
<b>5 应用示例.....</b>	<b>28</b>
5.1 示例 1: 查询任务列表.....	28
5.2 示例 2: 查询缺陷概要.....	30
<b>6 附录.....</b>	<b>33</b>
6.1 状态码.....	33
6.2 错误码.....	36
6.3 获取项目 ID.....	37
6.4 获取账号 ID.....	38
<b>7 历史 API.....</b>	<b>39</b>
7.1 查询任务详情.....	39

---

7.2 执行检查任务.....	42
7.3 新建检查任务.....	44
7.4 根据项目 Id 查询任务列表.....	46
<b>8 文档修订记录.....</b>	<b>49</b>

# 1 使用前必读

- 1.1 概述
- 1.2 调用说明
- 1.3 终端节点
- 1.4 约束与限制
- 1.5 基本概念

## 1.1 概述

代码检查（CodeCheck）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相关概念，详细信息请参见[产品介绍](#)。

## 1.2 调用说明

代码检查提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用，调用方法请参见[如何调用API](#)。

## 1.3 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。

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

## 1.4 约束与限制

详细的限制请参见具体API的说明。

## 1.5 基本概念

- 账号

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

- 用户

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

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

- 区域 ( Region )

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

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

详情请参见[区域和可用区](#)。

- 可用区 ( AZ, Availability Zone )

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

- 企业项目

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

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

# 2 API 概览

表 2-1 代码检查 API 概览

类型	子类型	说明
代码检查	<a href="#">4.1 缺陷管理</a>	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。
	<a href="#">4.2 任务管理</a>	任务管理包括：新建、执行、查询、终止任务等。

# 3 如何调用 API

## 3.1 构造请求

### 3.2 认证鉴权

### 3.3 返回结果

## 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[获取用户Token](#)获取请求认证接口说明如何调用API，该API获取用户的Token，Token可以用于调用其他API时鉴权。

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

请求URI由如下部分组成。

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

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

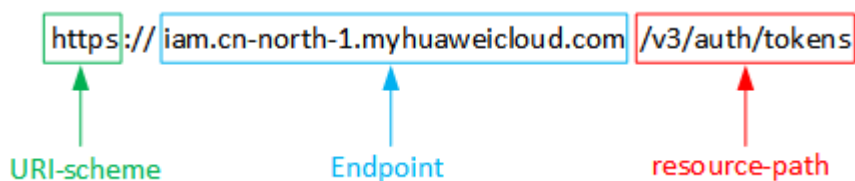
- **URI-scheme**：表示用于传输请求的协议，当前所有API均采用HTTPS协议。
- **Endpoint**：指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从[1.3 终端节点](#)中获取。例如IAM服务在“华北-北京一”区域的Endpoint为“iam.cn-north-1.myhuaweicloud.com”。
- **resource-path**：资源路径，也即API访问路径。从具体API的URI模块获取，例如“获取用户Token”API的resource-path为“/v3/auth/tokens”。
- **query-string**：查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“limit=10”，表示查询不超过10条数据。

例如，您需要获取IAM在“华北-北京一”区域的Token，则需使用“华北-北京一”区域的Endpoint（iam.cn-north-1.myhuaweicloud.com），并在[获取用户Token](#)获取请求认证的URI部分找到resource-path（/v3/auth/tokens），拼接起来如下所示。

```
https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
```



图 3-1 URI 示意图



### 说明

为查看方便，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，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**获取用户Token**获取请求认证的URI部分，您可以看到其请求方法为“POST”，则其请求为：

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
```

## 请求消息头

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

如下公共消息头需要添加到请求中。

- **Content-Type**：消息体的类型（格式），必选，默认取值为“application/json”，有其他取值时会在具体接口中专门说明。
- **X-Auth-Token**：用户Token，可选，当使用Token方式认证时，必须填充该字段。用户Token也就是调用**获取用户Token**获取请求认证接口的响应值，该接口是唯一不需要认证的接口。

### 说明

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

AK/SK认证的详细说明请参见**3.2 认证鉴权**。

- **X-Project-ID**：子项目ID，可选，在多项目场景中使用。
- **X-Domain-ID**：账号ID。

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

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

## 请求消息体

请求消息体通常以结构化格式发出，与请求消息头中Content-type对应，传递除请求消息头之外的内容。若请求消息体中参数支持中文，则中文字符必须为UTF-8编码。

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

对于**获取用户Token**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，其中***username***为用户名，***domainname***为用户所属的账号名称，***\*\*\*\*\****为用户登录密码，***xxxxxxxx***为project的名称，如cn-north-1，您可以从[1.3 终端节点](#)中获取。

### 说明

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

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

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

到这里为止这个请求需要的内容就具备齐全了，您可以使用[curl](#)、[Postman](#)或直接编写代码等方式发送请求调用API。对于**获取用户Token**获取请求认证接口，返回的响应消息头中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

## 3.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- Token认证：通过Token认证通用请求。
- 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。

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

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

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

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

```
GET https://iam.cn-north-1.myhuaweicloud.com/v3/auth/projects
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。

- AK(Access Key ID)：访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和私有访问密钥一起使用，对请求进行加密签名。
- SK(Secret Access Key)：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。

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

### 须知

签名SDK只提供签名功能，与服务提供的SDK不同，使用时请注意。

## 3.3 返回结果

### 状态码

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

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

对于[获取用户Token](#)获取请求认证接口，如果调用后返回状态码为“201”，则表示请求成功。

### 响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

对于[获取用户Token](#)获取请求认证接口，返回如[图3-2](#)所示的消息头，其中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

图 3-2 获取用户 Token 响应消息头

```

connection → keep-alive

content-type → application/json

date → Tue, 12 Feb 2019 06:52:13 GMT

server → Web Server

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

transfer-encoding → chunked

via → proxy A

x-content-type-options → nosniff

x-download-options → noopen

x-frame-options → SAMEORIGIN

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

x-subject-token
→ MIiYXQYJKoZIhvcNAQcCoIIYTCCEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOansiZXhwaXJlc19hdCI6IiwMTktMDItMTNUMC
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkajACgklqO1wi4JIGzrpd18LGXK5bdfq4lqHCYb8P4NaYONYejeAgzVefYtLWT1GSO0zxKZmlQHq82HBqHdgIZO9fuEbL5dMhdavj+33wEl
xHRCe9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXl1jipPEGA270g1FruooL6jggIFkNPQuFSOU8+uSsttVwRtnfC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUvhVpxk8pxiX1wTEboX-
RzT6MUbvpGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==

x-xss-protection → 1; mode=block;

```

## 响应消息体

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

对于[获取用户Token](#)获取请求认证接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

```

{
  "token": {
    "expires_at": "2019-02-13T06:52:13.855000Z",
    "methods": [
      "password"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "region_id": "cn-north-1",
            .....

```

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

```

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

```

其中，error\_code表示错误码，error\_msg表示错误描述信息。

# 4 API

## 4.1 缺陷管理

### 4.2 任务管理

## 4.1 缺陷管理

### 4.1.1 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。

#### 调试

您可以在[API Explorer](#)中调试该接口。

#### URI

GET /v2/tasks/{task\_id}/defects-summary

表 4-1 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 4-2 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 4-3 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
code_line_total	String	代码总行数
code_line	String	代码有效行数
code_quality	String	代码质量
issue_count	String	问题数
risk_coefficient	String	危险系数
duplication_ratio	String	重复比例
complexity_count	String	复杂度
duplicated_lines	String	重复行数

参数	参数类型	描述
comment_lines	String	注释行数
comment_ratio	String	注释比例
duplicated_blocks	String	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

## 请求示例

```
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxxeb6e57xxxxxxx/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@codehub.dev****_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"
}
```

**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](#)。

## 4.1.2 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。

### 调试

您可以在[API Explorer](#)中调试该接口。

### URI

GET /v2/tasks/{task\_id}/defects-detail

表 4-4 路径参数

参数	是否必选	参数类型	描述
task_id	是	String	任务ID

表 4-5 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

参数	是否必选	参数类型	描述
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 4-6 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 4-7 响应 Body 参数

参数	参数类型	描述
defects	Array of <a href="#">DefectInfoV2</a> objects	缺陷详情信息
total	Integer	总数

表 4-8 DefectInfoV2

参数	参数类型	描述
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

参数	参数类型	描述
created_at	String	创建时间
fragment	Array of <b>DefectFragmentV2</b> objects	缺陷代码片段详情

表 4-9 DefectFragmentV2

参数	参数类型	描述
line_num	String	行号
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

## 请求示例

GET <https://{endpoint}/v2/tasks/2b31ed520xxxxxxebdb6e57xxxxxxx/defects-summary>

## 响应示例

状态码: 200

Request succeeded!

```
{
  "defects": [ {
    "line_number": "81",
    "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",
    "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_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",
  "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) {}",
    "startOffset": 4,
    "endOffset": 12
  }, {
    "line_num": "191",
    "line_content": "    if (value === undefined) {}",
    "start_offset": -1,
    "end_offset": -1
  }
  ],
  "total": 2
}

```

### 状态码： 400

Bad Request

```

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

```

### 状态码： 401

Unauthorized

```

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

```

## 状态码

状态码	描述
200	Request succeeded!

状态码	描述
400	Bad Request
401	Unauthorized

## 错误码

请参见[错误码](#)。

## 4.2 任务管理

### 4.2.1 新建检查任务

#### 功能介绍

新建检查任务但是不执行。

#### 调试

您可以在[API Explorer](#)中调试该接口。

#### URI

POST /v2/{project\_id}/task

表 4-10 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 4-11 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： <b>1</b> 最大长度： <b>100000</b>

表 4-12 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

## 响应参数

状态码： 200

表 4-13 响应 Body 参数

参数	参数类型	描述
task_id	String	检查任务id

## 请求示例

```
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"
}
```

## 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 查询任务列表

### 功能介绍

根据DEVCLLOUD\_PROJECT\_UUID查询该项目下的任务列表。

### 调试

您可以在[API Explorer](#)中调试该接口。

### URI

GET /v2/{project\_id}/tasks

表 4-14 路径参数

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

表 4-15 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

参数	是否必选	参数类型	描述
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 4-16 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 4-17 响应 Body 参数

参数	参数类型	描述
tasks	Array of <a href="#">SimpleTaskInfoV2</a> objects	任务信息
total	Integer	总数

表 4-18 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



参数	参数类型	描述
git_branch	String	代码仓分支
created_at	String	创建时间
last_check_time	String	上一次检查时间

## 请求示例

```
GET https://{endpoint}/v2/2b31ed520xxxxxebedb6e57xxxxxxx/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@codehub.dev****_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@codehub.dev****_02201119.git",
    "git_branch": "master",
    "created_at": "2020-02-20 11:21:11",
    "last_check_time": "2020-02-20 11:37:46"
  } ],
  "total": 2
}
```

**状态码： 400**

Bad Request

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

**状态码： 401**

Unauthorized

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

## 状态码

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

## 错误码

请参见[错误码](#)。

## 4.2.3 执行检查任务

### 功能介绍

执行检查任务。

### 调试

您可以在[API Explorer](#)中调试该接口。

### URI

POST /v2/tasks/{task\_id}/run

表 4-19 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 4-20 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： <b>1</b> 最大长度： <b>100000</b>

表 4-21 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 4-22 响应 Body 参数

参数	参数类型	描述
exec_id	String	执行id

## 请求示例

POST https://{endpoint}/v2/tasks/d161fd0\*\*\*\*\*cb3c1d6a783e/run

## 响应示例

状态码： 200

Request succeeded!

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

状态码： 400

Bad Request

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

状态码： 401

Unauthorized

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

## 状态码

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

## 错误码

请参见[错误码](#)。

## 4.2.4 终止检查任务

### 功能介绍

根据任务ID终止检查任务。

### 调试

您可以在[API Explorer](#)中调试该接口。

### URI

POST /v2/tasks/{task\_id}/stop

表 4-23 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 4-24 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： <b>1</b> 最大长度： <b>100000</b>

## 响应参数

无

## 请求示例

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

## 响应示例

**状态码： 400**

Bad Request

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

**状态码： 401**

Unauthorized

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

## 状态码

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

## 错误码

请参见[错误码](#)。

## 4.2.5 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。

### 调试

您可以在[API Explorer](#)中调试该接口。

### URI

```
GET /v2/tasks/{task_id}/progress
```

表 4-25 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 4-26 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 4-27 响应 Body 参数

参数	参数类型	描述
task_status	String	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	<b>ProgressDetailV2</b> object	检查进度

表 4-28 ProgressDetailV2

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

## 请求示例

```
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxxeb6e57xxxxxxx/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](#)。

# 5 应用示例

[5.1 示例1: 查询任务列表](#)

[5.2 示例2: 查询缺陷概要](#)

## 5.1 示例 1: 查询任务列表

### 场景描述

本章节指导用户根据DEVCLLOUD\_PROJECT\_UUID查询该项目下的任务列表。

### 约束限制

无

### 涉及接口

涉及的接口如下:

[4.2.2 查询任务列表](#): 根据DEVCLLOUD\_PROJECT\_UUID查询任务列表。

### 操作步骤

查询任务列表。

- 接口相关信息

URI格式: GET /v2/{project\_id}/tasks

- 请求示例

```
GET https://{endpoint}/v2/2b31ed520xxxxxxebedb6e57xxxxxxx/tasks
```

信息请从[1.3 终端节点](#)获取。

- 响应示例

```
{
  "result": [
    {
      "projectId": "b9db371361724226937f3280df1d4fc4",
      "devCloudProjectId": "a2b9a42ee6494e0794f5655f8535920e",
      "devCloudProjectName": "TestDemo",
      "codeHubName": "portal-ts",
```



```

"codeHubWebUrl":"https://xxxxx/portal-ts",
"codeLineTotal":"63371",
"projectName":"portal-ts-abctest111222",
"creatorId":"ae161856f2604229ae12056478a7919f",
"creatorName":"devcloud_devcloud_y00336947_01",
"domainId":"78d0e09c1ad0425b9b4f3a8fdd066164",
"domainName":"devcloud_devcloud_y00336947_01",
"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":"devcloud_devcloud_y00336947_01",
"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：

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

## 操作步骤

- 接口相关信息  
URI格式：GET /v2/tasks/{task\_id}/defects-summary
- 请求示例  
GET https://{endpoint}/v2/tasks/2b31ed520xxxxxebedb6e57xxxxxxx/defects-summary  
信息请从 [1.3 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":"devcloud_devcloud_y00336947_01",
      "domainId":"78d0e09c1ad0425b9b4f3a8fdd066164",
      "domainName":"devcloud_devcloud_y00336947_01",
      "gitUrl":"git@xxxxx/portal-ts.git",
      "branch":"master",
      "language":["
      \"html\"
      ],
      "filePath":["
      \"\"
      ],
      "lastCheckTime\":"2019-09-24 15:00:41",
      "codeQuality":"141.7",
      "codeLine":"55777",
      "projectStatus":"2",
      "riskCoefficient":"790",
      "starLevel":"0",
      "defectInfo":{
        "totalCount":"0",
        "criticalCount":"0",
        "seriousCount":"0",
        "normalCount":"0",
        "promptCount":"0"
      },
      "defectStatusInfo":{
        "totalCount":"2952",
        "newCount":"2952",
        "ignoredCount":"0",
        "fixedCount":"0"
      },
      "progress":{
      },
      "createdAt":"2019-08-08 17:56:40",
      "duplications":"24.0%",
      "complexityNum":"0",
      "complexityAvg":"0",
      "duplicatedLines":"15206",
      "commentLines":"0",
      "commentRatio":"0",
      "duplicatedBlocks":"0",
      "lastExecTime":"2 分 4 秒",
      "checkType":"source",
      "dependenceType":"",
      "logDatasStr":{
      }
    }
  }
}
```

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

# 6 附录

6.1 状态码

6.2 错误码

6.3 获取项目ID

6.4 获取账号ID

## 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。

状态码	编码	错误码说明
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。
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。

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

## 6.2 错误码

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

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

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

### 错误响应 Body 体格式说明

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

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

其中，“error\_code”表示错误码，“error\_msg”表示错误描述信息。

### 错误码说明

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

更多服务错误码请参见[API错误中心](#)。

表 6-2 错误码

错误码	错误信息	描述	处理措施
DEV-31-50000	parameters not valid	参数非法或无效	检查参数格式
DEV-31-50001	query task list error	查询任务列表异常	联系服务负责人
DEV-31-50002	query task detail error	查询任务详情异常	联系服务负责人
DEV-31-50003	query templates error	查看任务配置模板异常	联系服务负责人
DEV-31-50004	query task history error	查询任务历史异常	联系服务负责人
DEV-31-50005	create task error	创建任务异常	联系服务负责人
DEV-31-50006	service inner error	服务内部错误	联系服务负责人



错误码	错误信息	描述	处理措施
DEV-31-50007	IAM authentication failed	IAM校验失败	检查认证token是否有效
DEV-31-50010	current user does not have permission	权限不足	检查权限
DEV-31-50011	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
DEV-31-50012	retry later	网络繁忙	待会重试或者联系服务负责人
DEV-31-50013	project not exist	项目不存在或项目已被删除	修改项目相关参数
DEV-31-50014	call getTaskBasicInfo failed	调用 getTaskBasicInfo 接口异常	待会重试或者联系服务负责人

## 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[1.3 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. 单击用户名，在下拉列表中单击“基本信息”。
  3. 在基本信息页面单击“管理我的凭证”。
- 在“API凭证”页面的项目列表中查看项目ID。

图 6-1 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. 单击用户名，在下拉列表中单击“我的凭证”。
- 在“API凭证”页面中查看账号ID。

图 6-2 获取账号 ID



# 7 历史 API

- [7.1 查询任务详情](#)
- [7.2 执行检查任务](#)
- [7.3 新建检查任务](#)
- [7.4 根据项目Id查询任务列表](#)

## 7.1 查询任务详情

### 功能介绍

已废弃，请使用V2版本。查询任务检查结果的详情。包括问题概述、问题状态、圈复杂度、代码重复率等。

### 调试

您可以在[API Explorer](#)中调试该接口。

### URI

GET /v1/tasks/{task\_id}

表 7-1 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 7-2 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）。

## 响应参数

状态码： 200

表 7-3 响应 Body 参数

参数	参数类型	描述
result	<b>TaskInfo</b> object	返回的任务信息详情
error	String	错误信息
status	String	接口执行结果状态

表 7-4 TaskInfo

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
code_line_total	String	代码总行数
code_line	String	代码有效行数
code_quality	String	代码质量
issue_count	String	问题数
task_status	String	任务状态

参数	参数类型	描述
risk_coefficient	String	危险系数
duplication_ratio	String	重复比例
complexity_count	String	复杂度
duplicated_lines	String	重复行数
comment_lines	String	注释行数
comment_ratio	String	注释比例
duplicated_blocks	String	重复块
last_execute_time	String	上次执行时间
check_type	String	检查类型
created_at	String	创建时间

### 请求示例

GET https://{endpoint}/v1/tasks/2b31ed520xxxxxxebdb6e57xxxxxxx

### 响应示例

状态码： 200

Request succeeded!

```
{
  "result": {
    "task_id": "435b58ecf7f54c45907c1384d01c94a6",
    "task_name": "CSharp_02201119",
    "creator_id": "ecadebb5041146cc96f9780f3d905e20",
    "git_url": "git@codehub.dev****_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,
    "task_status": 2,
    "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",  
"progress": {  
  "ratio": "50%",  
  "info": "执行检查中"  
},  
},  
"error": null,  
"status": "success"  
}
```

## 状态码

状态码	描述
200	Request succeeded!
401	Unauthorized
403	Forbidden
404	Not Found

## 错误码

请参见[错误码](#)。

## 7.2 执行检查任务

### 功能介绍

已废弃，请使用V2版本。执行检查任务。

### 调试

您可以在[API Explorer](#)中调试该接口。

### URI

POST /v1/tasks/execute

### 请求参数

表 7-5 请求 Header 参数

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

表 7-6 请求 Body 参数

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

## 响应参数

状态码： 200

表 7-7 响应 Body 参数

参数	参数类型	描述
result	<a href="#">ExecuteTaskInfo</a> object	检查类型，数组格式，默认为source
error	String	错误信息
status	String	接口执行结果状态

表 7-8 ExecuteTaskInfo

参数	参数类型	描述
taskId	String	任务id
execId	String	执行id

## 请求示例

POST https://{endpoint}/v1/tasks/execute

```
{
  "project_id": "yu87r4938y*****jju"
}
```

## 响应示例

状态码： 200

Demo Information

```
{
  "result": {
    "taskId": "d161fd0*****cb3c1d6a783e",
    "execId": "d163535d0*****81d6a7654443e"
  },
  "error": null,
  "status": "success"
}
```

## 状态码

状态码	描述
200	Demo Information
401	Unauthorized
403	Forbidden
404	Not Found

## 错误码

请参见[错误码](#)。

## 7.3 新建检查任务

### 功能介绍

已废弃，请使用V2版本。新建检查任务但是不执行。

### 调试

您可以在[API Explorer](#)中调试该接口。

### URI

POST /v1/projects/{project\_id}/tasks

表 7-9 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 7-10 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）。



表 7-11 请求 Body 参数

参数	是否必选	参数类型	描述
check_type	是	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

## 响应参数

状态码： 200

表 7-12 响应 Body 参数

参数	参数类型	描述
result	CreateTaskInfo object	检查类型，数组格式，默认为source
error	String	错误信息
status	String	接口执行结果状态

表 7-13 CreateTaskInfo

参数	参数类型	描述
TaskId	String	创建的检查任务id

## 请求示例

```
POST https://{endpoint}/v1/projects/{project_id}/tasks
{
  "check_type": [ "source" ],
  "git_url": "git@code*****958.git",
  "git_branch": "master"
}
```

## 响应示例

状态码： 200

Demo Information

```
{
  "result": {
    "TaskId": "d161fd0*****cb3c1d6a783e"
  },
  "error": null,
}
```

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

## 状态码

状态码	描述
200	Demo Information
401	Unauthorized
403	Forbidden
404	Not Found

## 错误码

请参见[错误码](#)。

## 7.4 根据项目 Id 查询任务列表

### 功能介绍

已废弃，请使用V2版本。根据DEVCLLOUD\_PROJECT\_UUID查询任务列表。

### 调试

您可以在[API Explorer](#)中调试该接口。

### URI

GET /v1/projects/{project\_id}/tasks

表 7-14 路径参数

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

表 7-15 Query 参数

参数	是否必选	参数类型	描述
start_index	否	Integer	开始索引 缺省值：1
page_size	否	Integer	每页个数 缺省值：10

## 请求参数

表 7-16 请求 Header 参数

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

## 响应参数

状态码： 200

表 7-17 响应 Body 参数

参数	参数类型	描述
result	Array of <a href="#">SimpleTaskInfo</a> objects	返回的任务信息详情列表
error	String	错误信息
status	String	接口执行结果状态

表 7-18 SimpleTaskInfo

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
created_at	String	创建时间
last_check_time	String	上一次检查时间

## 请求示例

无

## 响应示例

**状态码： 200**

Request succeeded!

```
{
  "result": [ {
    "task_id": "435b58e*****d01c94a6",
    "task_name": "CSharp_02201119",
    "creator_id": "ecadebb5041***9780f3d905e20",
    "git_url": "git@codehub.dev****_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@codehub.dev****_02201119.git",
    "git_branch": "master",
    "created_at": "2020-02-20 11:21:11",
    "last_check_time": "2020-02-20 11:37:46"
  } ],
  "error": null,
  "status": "success"
}
```

## 状态码

状态码	描述
200	Request succeeded!
401	Unauthorized
403	Forbidden
404	Not Found

## 错误码

请参见[错误码](#)。

# 8 文档修订记录

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
第八次正式发布。本次变更说明如下： <ul style="list-style-type: none"><li>• 修改4.1 缺陷管理。</li><li>• 修改4.2 任务管理。</li></ul>	2020-11-23
第七次正式发布。本次变更说明如下： <ul style="list-style-type: none"><li>• 修改5.1 示例1：查询任务列表。</li><li>• 修改5.2 示例2：查询缺陷概要。</li></ul>	2020-11-05
第六次正式发布。本次变更说明如下： <ul style="list-style-type: none"><li>• 修改1.4 约束与限制。</li><li>• 修改1.5 基本概念。</li><li>• 修改3.1 构造请求。</li><li>• 修改3.2 认证鉴权。</li></ul>	2020-08-06
第五次正式发布。本次变更说明如下： 增加应用示例。 <ul style="list-style-type: none"><li>• 5.1 示例1：查询任务列表。</li><li>• 5.2 示例2：查询缺陷概要。</li></ul>	2020-07-30
第四次正式发布。本次变更说明如下： <ul style="list-style-type: none"><li>• 修改“请求示例”中URL地址信息。</li><li>• 修改1.3 终端节点。</li></ul>	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

修订记录	发布日期
第二次正式发布。本次变更说明如下： 代码检查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