

天翼云微服务云应用平台产品 OpenAPI 标准

第一部分

OpenAPI Standard for ServiceStage of China Telecom Cloud

-Part One

目 录

天翼云微服务云应用平台产品 OPENAPI 标准	2
1 范围	3
2 规范性引用文件	3
3 缩略语、术语和定义	3
3.1 缩略语	3
3.2 术语和定义	3
4 调用前必知	3
4.1 概述	4
4.2 终端节点	4
4.3 请求状态码	4
5 API 概览	4
6 如何调用 API	5
6.1 构造请求与鉴权	5
6.2 信息的获取	5
6.3 基本签名流程	5
7 API	7
7.1 新建应用	7
7.2 修改应用	8
7.3 删除应用	9
7.4 应用列表	9
7.5 应用详情	10
7.6 应用编码合法性检测	10
7.7 租户实例统计	11
8 更新历史	12
9 其他	12

天翼云微服务云应用平台产品 OpenAPI 标准

1 范围

本说明提供了天翼云微服务云应用平台品API的描述、语法、参数说明及示例等内容。
本说明仅适用于天翼云公有云的自研产品，不涉及合营云产品的API调用说明。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的，本文件遵循了下列文件的技术规范。
天翼云OpenAPI技术规范（征求意见稿）

3 缩略语、术语和定义

3.1 缩略语

下列缩略语适用于本文件。

3.2 术语和定义

下列术语和定义适用于本文件。

3.2.1 应用

应用是一个功能相对完备的业务系统，由一个或多个特性相关的应用组件组成。

3.2.2 应用组件

应用组件是组成应用的某个业务特性实现，以代码或者软件包为载体，可独立部署运行。

3.2.3 微服务

微服务是业务概念，某个进程提供某种服务，那它就是个微服务。每一个服务都具有自主运行的业务功能，对外开放不受语言限制的API（最常用的是HTTP）。多个微服务组成应用程序。在ServiceStage中，微服务对应应用组件

3.2.4 微服务实例

微服务的最小运行和部署单元，通常对应一个应用进程。

4 调用前必知

4.1 概述

天翼云微服务云应用平台（CT-ServiceStage）是面向企业的微服务应用管理与运维平台，提供应用发布、部署、监控与运维等一站式解决方案。支持Java、Go、PHP、Node.js、Docker、Tomcat等运行环境。支持Web应用，以及Apache ServiceComb、Spring Cloud、Dubbo等微服务应用和通用应用，让企业应用上云更简单。

4.2 终端节点

终端节点（Endpoint）即调用API的**请求地址**，不同服务不同区域的终端节点不同。

资源池	Endpoint
重庆 2	servicestage-cq2-a
南京 3	servicestage-jsnj3-a

4.3 请求状态码

正常状态码	描述
800	操作成功
900	失败
2201	应用不存在或已被删除
2203	应用编码不合规，只能由小写英文及数字组成，同时以小写英文开头
2204	应用编码已存在，请重新填写其他的应用名称
2205	应用名称已存在，请重新填写其他的应用名称
2207	应用下服务不为空

5 API 概览

微服务云应用平台产品API说明

类型	描述
新建应用	新建微服务应用
修改应用	修改微服务应用
删除应用	删除微服务应用，需要先删除应用下所有服务和环境
应用列表	部分也查询应用列表
应用详情	获取微服务应用明细
应用编码合法性检测	应用编码只能由小写英文及数字组成，同时以小写英文开头，长度在 5-32 个字符间
租户实例统计	统计某段时间内微服务实例个数

6 如何调用 API

6.1 构造请求与鉴权

6.2 信息的获取

登录云网门户，在“控制台”->“个人中心”->“**第三方账号绑定**”，通过创建或者查看获取 ak, sk。

第三方账号绑定:	第三方账号绑定情况, 可进行解绑操作	解绑
用户AccessKey:	支持新建、查询用户AccessKey, 用于第三方控制管理	***** 查看



6.3 基本签名流程

ctyun-eop-ak/ctyun-eop-sk 基本签名流程

- 1、待签字符串：使用规范请求和其他信息创建待签字符串；
- 2、计算密钥：使用 HEADER、ctyun-eop-sk、ctyun-eop-ak 来创建 Hmac 算法的密钥；
- 3、计算签名：使用第三步的密钥和待签字符串在通过 hmacsha256 来计算签名。
- 4、签名应用：将生成的签名信息作为请求消息头添加到 HTTP 请求中。

创建待签名字符串

待签名字符串的构造规则如下：

待签名字符串=需要进行签名的 Header 排序后的组合列表+ “\n” + 排序的 query + “\n” + toHex (sha256(原封的 body))

<p>^a 需要进行签名的 Header 排序后的组合列表</p> <p>^b (排序的 header)</p>	<p>^c header 以 header_name:header_value 来一个一个通过\n 拼接起来, EOP 是强制要求 ctyun-eop-request-id 和 eop-date 这个头作为 Header 中的一部分, 并且必须是待签名 Header 里的一个。需要进行签名算法的 Header 需要进行排序(将它们的 header_name 以 26 个英文字母的顺序来排序), 将排序后得到的列表进行遍历组装成待签名的 header。</p>
<p>^d 排序的 query</p>	<p>^e query 以&作为拼接, key 和值以=连接, 排序规则使用 26 个英文字母的顺序来排序, Query 参数全部都需要进行签名</p>
<p>^f toHex (sha256(原封的 body))</p>	<p>^g 传进来的 body 参数进行 sha256 摘要, 对摘要出来的结果转十六进制</p>

排序的 header 例子:

假设你需要将 ctyun-eop-request-id、eop-date、host 都要签名, 则待签名的 header 构造出来是:

```
ctyun-eop-request-id:123456789\neop-date:20210531T100101Z\nhost:1.1.1.1:9080\n
```

ctyun-eop-request-id、eop-date 和 host 的排序就是这个顺序, 如果你加入一个 ccad 的 header; 同时这个 header 也要是进行签名, 则待签名的 header 组合:

```
ccda:123\n ctyun-eop-request-id:123456789\neop-date:20210531T100101Z\nhost:1.1.1.1:9080\n
```

构造动态密钥

发起请求时, 需要构造一个 eop-date 的时间, 这个时间的格式是 yyyyymmddTHHMMSSZ; 言简意赅一些, 就是年月日 T 时分秒 Z

- 1、先是拿你申请来的 ctyun-eop-sk 作为密钥, eop-date 作为数据, 算出 ktime
- 2、拿 ktime 作为密钥, 你申请来的 ctyun-eop-ak 数据, 算出 kAk;
- 3、拿 kAk 作为密钥, eop-date 的年月日值作为数据; 算出 kdate

<p>^h eop-date</p>	<p>ⁱ yyyyymmddTHHMMSSZ (20211221T163614Z) (年月日 T 时分秒 Z)</p>
<p>^j Ktime</p>	<p>^k 使用 ctyun-eop-sk 作为密钥, eop-date 作为数据, 算出 ktime;</p> <p>^l Ktime = hmacSha256(ctyun-eop-sk, eop-date)</p>
<p>^m kAk</p>	<p>ⁿ 使用 ktime 作为密钥, 你申请来的 ctyun-eop-ak 数据, 算出 kAk;</p> <p>^o kAk = hmacsha256(ktime, ctyun-eop-ak)</p>

^p kdate	^q 使用 kAk 作为密钥, eop-date 的年月日值作为数据; 算出 kdate; ^r kdate = hmacsha256(kAk, eop-date)
--------------------	---

签名应用及示例

由“构造动态密钥”和“创建待签名字符串”分别的出来的待签名字符串 string_sigture、kdate 生成出 Signature;

^s Signature	^t 待 签 名 字 符 串 string_sigture 、 kdate ; 再 根 据 hmacsha256(kdate, string_sigture)得出的结果, 再将结果进行 base64 编码得出 Signature
^u Eop-Authorization	^v ctyun-eop-ak Header=你构造待签名字符串时的 header 排序 Signature (注意中间有空格) header 排序以分号” ;” 拼接。 ^w 例子: 你待签名的字符串 header 顺序是 eop-date 和 host; 那么你加到 header 里的值就是 ^x Eop-Authorization: ctyun-eop-ak Header=eop-date;host Signature=xad01/ada

由上得到 Eop-Authorization, 然后将数据整合成 HEADER 放在 http_client 内, 发出即可。

http_client 所需请求头部如下:

Eop-Authorization: ctyun-eop-ak Header= ctyun-eop-request-id;eop-date Signature=xad01/ada

eop-date:20211221T163614Z

ctyun-eop-request-id: 123456789

(注: 若需要进行签名的 Header 不止默认的 ctyun-eop-request-id 和 eop-date, 需要在 http_client 的请求头部中加上, 并且 Eop-Authorization 中也需要增加)

7 API

7.1 新建应用

URL: https://[endpoint].ctapi.ctyun.cn/v1/application/create

Method: POST

请求:

参数名	是否必填	参数类型	说明
-----	------	------	----

customInfo	是	Header String	公共信息
regionId	是	Header String	资源池 Id
accountId	是	Header String	PaaS 组件实例所归属天翼云账号标识
request	是	RequestBody ApplicationCreateVO	应用信息

ApplicationCreateVO:

参数名	是否必填	参数类型	说明
appCode	是	String	应用编码
appName	是	String	应用名称
appDesc	否	String	应用描述
iconPath	否	String	应用图标地址

响应:

成功:

```
{ "statusCode": 800, "message": "操作成功" }
```

失败:

```
{ "statusCode": 2204, "message": "应用编码已存在，请重新填写其他的应用名称" }
```

7.2 修改应用

URL: [https://\[endpoint\].ctapi.ctyun.cn/v1/application/update](https://[endpoint].ctapi.ctyun.cn/v1/application/update)

Method: POST

请求:

参数名	是否必填	参数类型	说明
customInfo	是	Header String	公共信息
regionId	是	Header String	资源池 Id
accountId	是	Header String	PaaS 组件实例所归属天翼云账号标识
appId	是	RequestParam Long	应用 ID
request	是	RequestBody ApplicationUpdateVO	应用信息

ApplicationUpdateVO:

参数名	是否必填	参数类型	说明
appName	是	String	应用名称
appDesc	否	String	应用描述
iconPath	否	String	应用图标地址

响应:

成功:

```
{ "statusCode": 800, "message": "操作成功" }
```

失败:

```
{ "statusCode": 2205, "message": "应用名称已存在, 请重新填写其他的应用名称" }
```

7.3 删除应用

URL: [https://\[endpoint\].ctapi.ctyun.cn/v1/application/delete](https://[endpoint].ctapi.ctyun.cn/v1/application/delete)

Method: POST

请求:

参数名	是否必填	参数类型	说明
customInfo	是	Header String	公共信息
regionId	是	Header String	资源池 Id
accountId	是	Header String	PaaS 组件实例所归属天翼云账号标识
appld	是	RequestParam Long	应用 ID

响应:

成功:

```
{ "statusCode": 800, "message": "操作成功" }
```

失败:

```
{ "statusCode": 2207, "message": "应用下服务不为空" }
```

7.4 应用列表

URL: [https://\[endpoint\].ctapi.ctyun.cn/v1/application](https://[endpoint].ctapi.ctyun.cn/v1/application)

Method: GET

请求:

参数名	是否必填	参数类型	说明
customInfo	是	Header String	公共信息
regionId	是	Header String	资源池 Id
accountId	是	Header String	PaaS 组件实例所归属天翼云账号标识

应:

成功:

```
{  
  "statusCode": 0,  
  "message": "操作成功",  
  "returnObj": [{
```

响

```

    "appName":"应用 1",
    "appCode":"myapp1",
    "appDesc":"应用 1",
    "iconPath":"应用描述 1",
    "tenantId":1,
    "statusCd":"S0A",
    "createTime":"2019-11-01 14:32:21",
    "updateTime":"2019-11-02 15:32:12"
  }
]
失败:
{"statusCode":900,"message":"操作异常","returnObj":null}

```

7.5 应用详情

URL: https://[endpoint].ctapi.ctyun.cn/v1/application/byid

Method: GET

请求:

应:

参数名	是否必填	参数类型	说明
customInfo	是	Header String	公共信息
regionId	是	Header String	资源池 Id
accountId	是	Header String	PaaS 组件实例所归属天翼云账号标识
appId	是	RequestParam Long	应用 ID

响

成功:

{

```

"statusCode": 800,
"message": "操作成功",
"returnObj":{
  "appName":"应用 1",
  "appCode":"myapp1",
  "appDesc":"应用 1",
  "iconPath":"应用描述 1",
  "tenantId":1,
  "statusCd":"S0A",
  "createTime":"2019-11-01 14:32:21",
  "updateTime":"2019-11-02 15:32:12"
}
}
失败:
{"statusCode":900,"message":"操作异常","returnObj":null}

```

7.6 应用编码合法性检测

URL: https://[endpoint].ctapi.ctyun.cn/v1/application/checkAppCode

Method: GET

请求:

参数名	是否必填	参数类型	说明
customInfo	是	Header String	公共信息
regionId	是	Header String	资源池 Id
accountId	是	Header String	PaaS 组件实例所归属天翼云账号标识
appCode	是	RequestParam String	应用编码
appId	否	RequestParam Long	应用 ID

响应:

成功:

```
{"statusCode": 800,"message": "合法","returnObj":true}
```

失败:

```
{"statusCode":2204,"message":"应用编码已存在"}
```

7.7 租户实例统计

URL: [https://\[endpoint\].ctapi.ctyun.cn/v1/application/run-status](https://[endpoint].ctapi.ctyun.cn/v1/application/run-status)

Method: GET

请求:

参数名	是否必填	参数类型	说明
customInfo	是	Header String	公共信息
regionId	是	Header String	资源池 Id
accountId	是	Header String	PaaS 组件实例所归属天翼云账号标识
startTime	是	RequestParam String	开始时间, 格式为 yyyy-MM-dd HH:mm:ss
endTime	是	RequestParam String	结束时间, 格式为 yyyy-MM-dd HH:mm:ss

响应:

成功:

```
{
  "statusCode": 800,
  "message": "合法",
  "returnObj": {
    "tenantId": 1,
    "instanceCount": 3
  }
}
```

失败:

```
{"statusCode":900,"message":"操作异常"}
```

8 更新历史

9 其他