天翼云函数计算产品 OpenAPI 标准

第一部分

OpenAPI Standard for CF of China Telecom Cloud

-Part One

2022-04 发布 2022-04 实施

目　　录

天翼云函数计算产品 OpenAPI 标准

1 范围

本说明提供了天翼云函数计算品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。

缩略语	英文全称	中文全称
CF	Cloud Function	云函数

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

应用指的是支撑某个特定业务，功能或服务相对内聚的一个逻辑单元，一个应用可以由多个函数组成。在函数计算平台中，主要用于聚合函数。

3.2.2 函数

函数是函数计算平台管理和运行的最小单元，也指用户提交的一组代码。当事件（请求）到达时，平台会启用函数容器执行函数代码处理请求。

3.2.3 运行环境

开发语言对应的运行时环境。

3.2.4 函数入口

代码执行入口，类似于Java的main。

3.2.5 内存

指函数容器运行时向平台申请的内存。

3.2.6 超时时间

触发事件（请求）时，期望在阈值内得到响应的时间值，超过则终止。

3.2.7 触发器

触发函数执行的事件源组件。常见的触发器有定时触发器，各种云服务（例如对象存储）的触发器。

3.2.8 版本

函数运行时的快照。利用版本可以支持灰度发布。

4 调用前必知

4.1 概述

本说明提供了函数计算产品API的描述、语法、参数说明及示例等内容。

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
上海 7	cf-sh7-a
保定	cf-hbbd1-a

4.3 分页查询

无

4.4 请求状态码

正常状态码	描述
0	成功
500	服务器内部错误
1000	用户未登录或登录失败
1001	参数校验失败
1002	禁止越权操作
1005	数据库操作失败
1006	文件操作失败
1007	K8S 操作失败
1008	操作超时
1010	查询失败

1101	记录已存在
1102	记录不存在
1104	存在关联数据
2000	开通失败
2001	超过最大可创函数
2002	尚未开通
2003	操作最大可创版本
2004	因关联函数无法删除应用

5 API 概览

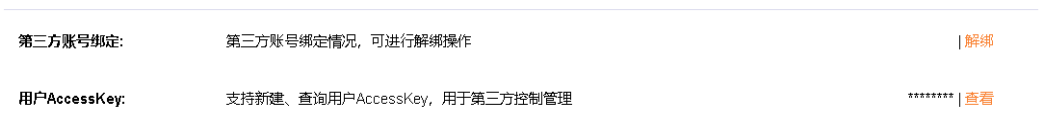
函数计算产品API说明

类型	描述
应用管理（锚点链接至具体的 API 中）	包含了创建应用、删除应用、查询应用等 API
函数管理（锚点链接至具体的 API 中）	包含了创建函数等 API

6 如何调用 API

6.1 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。



6.2 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))

[a] 需要进行签名的 Header 排序后的组合列表 [b] （排序的 header）	[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。
[d] 排序的 query	[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

f toHex(sha256(原封的 body))	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

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 yyyymmddTHHMMSSZ;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

h eop-date	i yyyymmddTHHMMSSZ（20211221T163614Z）(年月日 T 时分秒 Z)
j Ktime	k 使用 ctyun-eop-sk 作为密钥，eop-date 作为数据，算出 ktime； l Ktime = hmacSha256(ctyun-eop-sk, eop-date)
m kAk	n 使用 ktime 作为密钥，你申请来的 ctyun-eop-ak 数据，算出 kAk； o kAk = hmacsha256(ktime,ctyun-eop-ak)
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“构造动态秘钥”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 创建应用

URL：CF/openapi/v1/cf/application

Method：POST

请求：

```
y {
z   "appName": "app",
aa   "appDesc": "application description"
bb }
```

cc 字段	dd 名称	ee 类型	ff 必填	gg 默认值	hh 描述
ii appName	jj 应用名称	kk String	ll 是	mm 无	nn
oo appDesc	pp 应用描述	qq String	rr 否	ss 无	tt

响应：

```
uu {
vv   "code": 0,
ww   "data": 10000
xx }
```

yy 字段	zz 名称	aaa 类型	bbb 必填	ccc 默认值	ddd 描述
eee code	fff 编码	ggg int	hhh 是	iii 无	jjj 0-表示成功
kkk data	lll 应用ID	mmm long	nnn 否	ooo 无	ppp

7.2 删除应用

URL：https://[endpoint].ctapi.ctyun.cn/v1/cf/application/{appID}

Method：DELETE

请求：（无）

响应：

```
{
  "code": 0,
  "data": true
}
```

字段	名称	类型	必填	默认值	描述
code	编码	int	是	无	0-表示成功
data	操作标记	boolean	否	无	

7.3 查询应用

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

Method：POST

请求：

```
{
  "appName": "app",
  "appDesc": "application description"
}
```

字段	名称	类型	必填	默认值	描述
appName	应用名称	String	否	无	
appDesc	应用描述	String	否	无	

响应：

```
{
  "code": 0,
  "data": [
    {
      "id": 10000,
```

```
    "appName": "app",
    "appDesc": "application description",
    "createdBy": 1,
    "createdTime": "2020-12-12 12:12:12",
    "modifiedBy": 1,
    "modifiedTime": "2020-12-12 12:12:12"
  }
 ]
}
```

字段	名称	类型	必填	默认值	描述
code	编码	int	是	无	0-表示成功
data	应用列表	Array	否	无	
id	应用ID	long	是	无	
appName	应用名称	String	是	无	
appDesc	应用描述	String	否	无	
createdBy	创建人ID	long	是	无	
createdTime	创建时间	Datetime	是	无	格式 2020-12-12 12:12:12
modifiedBy	修改人ID	String	是	无	
modifiedTime	修改时间	String	是	无	格式 2020-12-12 12:12:12

7.4 创建函数

URL：CF/openapi/v1/cf/function

Method：POST

请求：

```
{
  "appId": 3,
  "funcName": "func-1",
  "type": 1,
  "funcDesc": "function",
  "runtime": "python3",
  "handler": "handler.index",
```

```
    "sourceIndex": "29fb80cda03f4aeb8cc6902a618b6108.1619660724754",
    "memorySize": 128,
    "timeout": 10,
    "config": "{\"authorize\":[]}"
}
```

字段	名称	类型	必填	默认值	描述
appId	应用Id	long	是	无	
funcName	函数名称	String	是	无	
type	函数类型	int	是	无	1-非HTTP函数
funcDesc	函数描述	String	否	无	
runtime	运行时	String	是	无	nodejs
handler	函数入口	String	是	无	
sourceIndex	源码索引	String	是	无	
memorySize	内存	int	是	无	128,256,512,1024,2048
timeout	超时时间	long	是	无	
config	配置	String	否	无	

响应：

```
{
    "code": 0,
    "data": 10000
}
```

字段	名称	类型	必填	默认值	描述
code	编码	int	是	无	0-表示成功
data	函数ID	long	否	无	

8 更新历史

9 其他