

天翼云分布式缓存 Redis 产品 OpenAPI 标准

第一部分

OpenAPI Standard for Redis of China Telecom Cloud

-Part One

2022-04 发布

2022-04 实施

目 录

天翼云分布式缓存 REDIS 产品 OPENAPI 标准	2
1 范围	2
2 规范性引用文件	2
3 缩略语、术语和定义	2
3.1 缩略语	2
4 调用前必知	2
4.1 概述	2
4.2 终端节点	2
4.3 分页查询	2
4.4 请求状态码	2
5 API 概览	3
6 如何调用 API	3
6.1 构造请求	3
7 API	4
7.1 查询实例	4
7.2 重置实例密码	6
7.3 清空实例数据	7
7.4 重启实例	7
8 更新历史	8
9 其他	8

天翼云分布式缓存 redis 产品 OpenAPI 标准

1 范围

本说明提供了天翼云分布式缓存redis产品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
DCS	Distributed Cache Service	分布式缓存服务

（按缩略语字母顺序依次排）

4 调用前必知

4.1 概述

本说明提供了天翼云分布式缓存redis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。

4.3 分页查询

4.4 请求状态码

正常状态码	描述
800	成功
900	请求处理失败

5 API 概览

天翼云分布式缓存redis产品API说明

类型	描述
功能类型 1 (锚点链接至具体的 API 中)	包含了查实例、实例重启、实例数据清空等 API

6 如何调用 API

6.1 构造请求

请求头说明

参数	值(说明)
Content-Type	application/json
ctyun-eop-request-id	用户请求 id，通过 uuid 生成，形如 33dfa732-b27b-464f-b15a-21ed6845afd5
Eop-Authorization	由天翼云官网 accessKey 和 securityKey 经签名后生成，签名逻辑详见后续说明
Eop-Date	请求时间，形如 yyyyMMdd'T'HHmmss'Z'，例如 20211109T104641Z

请求代码示例

Bash:

```
Curl https://itpaas.[资源池编码].ctapi.ctyun.cn/ctcloud/ctapi/接口地址  
-H "Content-Type:application/json;charset=UTF-8" -H "eop-authorization:xxxxxxxxxxxxxxxxxxxx" -H "Eop-date:20211014T163109Z" -H "ctyun-eop-request-id:e5d0d445-29a5-4bfc-a4e8-a5b70f889cc3"
```

签名逻辑

基本签名流程

1、构造规范请求。将待发送的请求内容按照与 EOP 约定的规则组装，确保客户端签名、EOP 认证时使用

用的请求内容一致。

2、使用规范请求和其他信息创建待签字符串。

3、使用 HEADER、ctyun-eop-sk、ctyun-eop-ak 来创建 Hmac 算法的密钥

4、使用第三步的密钥和待签字符串在通过 hmacsha256 来计算签名。

5、将生成的签名信息作为请求消息头添加到 HTTP 请求的 header 中。

创建待签字符串

待签字符串的构造规则如下：

需要进行签名的 Header 排序后的组合列表(键和值)+ "\n" + "排序的 query" + "\n" + toHex(sha256(原封的 body))

1、header 以 header_name:header_value 来一个一个通过\n 拼接起来，并非是把所有 header 都加进来做待签字符串，而是你需要把哪些 Header 作为待签字符串，EOP 是强制要求 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。先将需要进行签名算法的 Header 提取出来，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。假设你需要将 eop-date，host 都要签名，则待签名的 header 构造出来是：eop-date:20210531T100101Z\nhost:1.1.1.1:9080\n；eop-date 和 host 的排序就是这个顺序，如果你加入一个 ccda 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 eop-date:20210531T100101Z\nhost:1.1.1.1:9080\n。

2、query 和 header 是一样的，只是它不以\n 为拼接，而是以&作为拼接，key 和值不是:连接而是以=连接，排序规则一致。Query 参数全部都需要进行签名。

3、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。

4、经过如上三步分别得出：排序的 header，排序的 query，原始的 body；这三个再以\n 连接起来，得出待签名的字符串。

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；Ktime = hmacSha256(sk, eop-date)

2、拿 ktime 作为密钥，你申请来的 ctyun-eop-ak 数据，算出 kAk；kAk = hmacsha256(ktime,ctyun-eop-ak)

3、拿 kAk 作为密钥，eop-date 的年月日值作为数据；算出 kdate

签名的使用

由“构造动态秘钥”和“创建待签字符串”分别的出来的待签字符串 string_sigtire、kdate；再根据 hmacsha256(kdate,string_sigtire)得出的结果，再将结果进行 base64 编码得出 Sigtire；

然后将数据整合成 HEADER 放在 http_client 内，发出即可。

HEADER 名称为 Eop-Authorization；

值是你的 ctyun-eop-ak Header=你构造待签字符串时的 header 排序 Signature=你算来的 Sigtire 值。

例子所述：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

Eop-Authorization: {ctyun-eop-ak} Header=eop-date;host Signature=xad01/ada

7 API

7.1 查询实例

POST /DCS2/services/open/restart

请求参数

参数名	是否必填	位置	参数类型	说明
prodInstId	非必填	header	String	查询指定实例信息的实例 ID

返回值

data 参数说明

参数名	类型	说明
instanceName	String	实例名称。
prodInstId	String	实例 ID。
archType	Integer	架构类型 1-集群版 2-标准版
nodeType	Integer	节点类型 1-双副本 2-单副本
engineVersion	String	缓存版本
capacity	String	实例规格大小（G）
netName	String	专有网络
securityGroup	String	安全组
subnet	String	子网
createTime	String	实例创建时间。格式为：YYYY-MM-DD HH:mm:ss
createTime	String	实例到期时间。格式为：YYYY-MM-DD HH:mm:ss
ip	String	实例连接 VPC 地址
port	Integer	实例连接端口
status	Integer	实例状态 0-有效 1-实例创建中 2-实例暂停服务 3-变更扩容中

成功:

```
{
  "returnObj": {
    "total": 1,
    "data": [
      {
        "engineVersion": "2.8",
        "subnet": "subnet-0916",
        "instanceName": "cache-BS8G20",
        "netName": "vpc-39b0",
```

```

        "expTime": "2019-10-20 16:12:35",
        "ip": "192.168.32.131",
        "securityGroup": "default",
        "nodeType": "1",
        "capacity": "8G",
        "createTime": "2019-09-20 16:08:52",
        "port": "22001",
        "archType": "2",
        "prodInstId": "627d149fe9c7438493edce765cd893b9",
        "status": "0"
    }
],
{
    "prodInstId": "627d149fe9c7438493edce765cd893b9",
    "message": "success",
    "statusCode": 800
}

```

失败:

```

{
    "statusCode": 900,
    "message": "XXXXXXX"
}

```

7.2 重置实例密码

POST /DCS2/services/open/updateInstancePassword

请求参数

参数名	是否必填	位置	参数类型	说明
prodInstId	*必填	body	String	实例 id
oldPassword	*必填	body	String	原密码
newPassword	*必填	body	String	新密码

返回值

成功:

```

{
    "statusCode": 800
}

```

失败:

```
{
  "statusCode":900,
  "message":"XXXXXXXX"
}
```

7.3 清空实例数据

POST /DCS2/services/open/flushData

请求参数

参数名	是否必填	位置	参数类型	说明
prodInstId	*必填	body	String	实例 id

返回值

成功:

```
{
  "statusCode":800
}
```

失败:

```
{
  "statusCode":900,
  "message":"XXXXXXXX"
}
```

7.4 重启实例

POST /DCS2/services/open/restart

请求参数

参数名	是否必填	位置	参数类型	说明
prodInstId	*必填	body	String	实例 id

返回值

成功:

```
{  
  "statusCode":800  
}
```

失败:

```
{  
  "statusCode":900,  
  "message":"XXXXXXXX"  
}
```

8 更新历史

9 其他