

内容分发网络

开发者手册

产品文档



腾讯云

【版权声明】

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

开发者手册

脚本工具

日志合并工具

开发者手册

脚本工具

最近更新时间：2019-04-29 16:16:31

调用准备

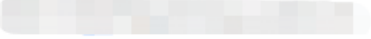
使用命令行脚本调用 API，需要配置 SecretId 和 SecretKey，登录腾讯云官网，进入 [访问管理控制台](#)，在左侧菜单中，单击【API 密钥管理】，进入管理页面后，可以看到调用 API 时需要的 SecretId 和 SecretKey，请妥善保管您的密钥。

API 密钥管理

调用腾讯云 API 时需要签名，云 API 密钥用于生成签名，查看 [生成签名算法](#)。

API 密钥是构建腾讯云 API 请求的重要凭证，使用腾讯云 API 可以操作您名下的所有腾讯云资源，为了您的财产和服务安全，请妥善保管和定期更换密钥，当您更换密钥后

新建密钥

APPID	密钥	创建时间	状态	操作
	SecretId:  SecretKey: ***** 显示	2015-06-12 09:25:04	已启用	禁用

使用说明

[下载](#) 命令行为 Python 脚本。

使用准备

在使用上述 Python 脚本时，您需要安装 requests 库，可使用如下命令：

```
pip install requests
```

您可以直接执行脚本，会提示您目前已经支持的接口清单：

```
Support Actions:
  AddCdnHost
  DeleteCdnHost
  DescribeCdnHostDetailedInfo
  DescribeCdnHostInfo
  DescribeCdnHosts
  GenerateLogList
  GetCdnMiddleSourceList
  GetCdnRefreshLog
  GetCdnStatTop
  GetCdnStatusCode
  GetHostInfoByHost
  GetHostInfoById
  OfflineHost
  OnlineHost
  RefreshCdnDir
  RefreshCdnUrl
  UpdateCache
  UpdateCdnConfig
  UpdateCdnHost
  UpdateCdnProject
```

接口功能说明，请参考 [API 概览](#)。

域名详情查询

查询所有域名详情

1. 使用如下命令，可以调用 [DescribeCdnHosts](#) 接口，查询 APPID 下所有域名详细信息：

```
python QcloudCdnTools_V2.py DescribeCdnHosts -u xxxxxx -p xxxxxxxx
```

2. 参数说明

- -u 表示 SecretId。
- -p 表示 SecretKey。

3. 结果示例

```
{
  u'total': 1,
  u'hosts': [
    {
      u'origin': u'8.8.8.8',
      u'enable_overseas': u'no',
      u'disabled': 0,
      u'create_time': u'2016-07-26 10:34:09',
      u'seo': u'off',
      u'message': u'\u90e8\u7f72\u4e2d',
      u'id': 279950,
```

```
u'cache': [  
  {  
    u'type': 0,  
    u'unit': u'd',  
    u'rule': u'all',  
    u'time': 2592000  
  },  
  {  
    u'type': 1,  
    u'unit': u's',  
    u'rule': u'.php;jsp;asp;aspx',  
    u'time': 0  
  }  
],  
u'middle_resource': -1,  
u'fwd_host_type': u'default',  
u'readonly': 0,  
u'fwd_host': u'www.test.com',  
u'service_type': u'media',  
u'project_id': 0,  
u'refer': {  
  u'list': [  
  ],  
  u'type': 0,  
  u'null_flag': 0  
},  
u'status': 4,  
u'update_time': u'2016-08-12 20: 36: 58',  
u'ssl_expire_time': None,  
u'deleted': u'no',  
u'ssl_type': 0,  
u'ssl_deploy_time': None,  
u'app_id': 1251991073,  
u'host': u'www.test.com',  
u'bucket_name': u'',  
u'host_id': 279950,  
u'pid_config': None,  
u'cache_mode': u'simple',  
u'furl_cache': u'on',  
u'host_type': u'cname',  
u'owner_uin': 78976504,  
u'cname': u'www.test.com.cdn.dns1.com'  
]  
...
```

```
]
}
```

根据域名查询域名详情

1. 使用如下命令，可以调用 [GetHostInfoByHost](#) 接口，查询指定域名对应的详情：

```
python QcloudCdnTools_V2.py GetHostInfoByHost -u xxxxxx -p xxxxxxxx --hosts www.test.com -
-hosts www.test2.com
```

2. 参数说明

- -u 表示SecretId。
- -p 表示SecretKey。
- --hosts 表示要查询的域名，可以一次查询多个，均使用 --hosts，注意，接口参数需要两条横线。

3. 结果示例

```
{
  u'total': 2,
  u'hosts': [
    {
      u'origin': u'8.8.8.8',
      u'enable_overseas': u'no',
      u'disabled': 0,
      u'create_time': u'2016-07-26 10: 34: 09',
      u'seo': u'off',
      u'message': u'\u5df2\u5f00\u542f',
      u'id': 1234,
      u'cache': [
        {
          u'type': 0,
          u'unit': u'd',
          u'rule': u'all',
          u'time': 2592000
        },
        {
          u'type': 1,
          u'unit': u's',
          u'rule': u'.php;.jsp;.asp;.aspx',
          u'time': 0
        }
      ]
    }
  ]
}
```

```
],
  u'middle_resource': -1,
  u'fwd_host_type': u'default',
  u'readonly': 0,
  u'fwd_host': u'www.test.com',
  u'service_type': u'media',
  u'project_id': 0,
  u'refer': {
    u'list': [
    ],
    u'type': 0,
    u'null_flag': 0
  },
  u'status': 5,
  u'update_time': u'2016-08-12 20: 36: 58',
  u'ssl_expire_time': None,
  u'deleted': u'no',
  u'ssl_type': 0,
  u'ssl_deploy_time': None,
  u'app_id': 1251991073,
  u'host': u'www.test.com',
  u'bucket_name': u'',
  u'host_id': 279950,
  u'pid_config': None,
  u'cache_mode': u'simple',
  u'furl_cache': u'on',
  u'host_type': u'cname',
  u'owner_uin': 78976504,
  u'cname': u'www.test.com.cdn.dns.v1.com'
},
...
]
```

根据域名 ID 查询域名详情

1. 使用如下命令，可以调用 [GetHostInfoById](#) 接口，查询ID对应域名详情：

```
python QcloudCdnTools_V2.py GetHostInfoById -u xxxxx -p xxxxxxxx --ids 1234
```

2. 参数说明

- -u 表示SecretId。

- -p 表示SecretKey。
- --ids 表示要查询的域名 ID，可以一次查询多个，均使用 --ids，注意，接口参数需要两条横线。

3. 结果示例

```
{
  u'total': 1,
  u'hosts': [
    {
      u'origin': u'8.8.8.8',
      u'enable_overseas': u'no',
      u'disabled': 0,
      u'create_time': u'2016-07-26 10: 34: 09',
      u'seo': u'off',
      u'message': u'\u5df2\u5f00\u542f',
      u'id': 1234,
      u'cache': [
        {
          u'type': 0,
          u'unit': u'd',
          u'rule': u'all',
          u'time': 2592000
        },
        {
          u'type': 1,
          u'unit': u's',
          u'rule': u'.php;.jsp;.asp;.aspx',
          u'time': 0
        }
      ],
      u'middle_resource': -1,
      u'fwd_host_type': u'default',
      u'readonly': 0,
      u'fwd_host': u'www.test.com',
      u'service_type': u'media',
      u'project_id': 0,
      u'refer': {
        u'list': [
        ],
        u'type': 0,
        u'null_flag': 0
      },
      u'status': 5,
      u'update_time': u'2016-08-12 20: 36: 58',
    }
  ]
}
```

```
u'ssl_expire_time': None,
u'deleted': u'no',
u'ssl_type': 0,
u'ssl_deploy_time': None,
u'app_id': 1251991073,
u'host': u'www.test.com',
u'bucket_name': u'',
u'host_id': 279950,
u'pid_config': None,
u'cache_mode': u'simple',
u'furl_cache': u'on',
u'host_type': u'cname',
u'owner_uin': 78976504,
u'cname': u'www.test.com.cdn.dnsv1.com'
}
]
}
```

刷新预热

刷新 URL

1. 使用如下命令，可以调用[RefreshCdnUrl](#) 接口，刷新指定URL：

```
python QcloudCdnTools_V2.py RefreshCdnUrl -u xxxxx -p xxxxxxxx --urls http://xxxxxxxxtang.sp.oa.com/test.php --urls http://www.test.com/1.html
```

2. 参数说明

- -u 表示SecretId。
- -p 表示SecretKey。
- --urls 表示要刷新的URL，可以一次查询多个，均使用 --urls。
- urls 参数前必须添加http:// 或者 https:// 前缀。
- 每个用户每天仅允许刷新 10000 条 URL，每次仅允许提交最多 1000 条进行刷新。

3. 结果示例

```
[
{
u'log_id': 220332,
u'count': 1
```

```
}  
]
```

其中 log_id 为提交的刷新任务 ID，您可以根据此 ID 查询该刷新任务的执行状态，count 为本次提交的刷新 URL 数目。

刷新目录

1. 使用如下命令，可以调用[RefreshCdnDir](#) 接口，刷新指定目录：

```
python QcloudCdnTools_V2.py RefreshCdnDir -u xxxxx -p xxxxxxxx --dirs http://www.test.com/abc/
```

2. 参数说明

- -u 表示SecretId。
- -p 表示SecretKey。
- --dirs 表示要刷新的 URL，可以一次查询多个，均使用 --dirs。
- dirs 参数前必须添加 http:// 或者 https:// 前缀。
- 每个用户每天仅允许刷新 100 条目录，每次仅允许提交最多 20 条进行刷新。

3. 结果示例

```
request is success.
```

查询刷新记录

1. 您可以使用如下命令查询刷新记录：

```
python QcloudCdnTools_V2.py GetCdnRefreshLog -u xxxxxxxxxxxx -p xxxxxxxxxxxx --startDate 2016-08-15 --endDate 2016-08-16
```

2. 参数说明

- -u 表示 SecretId。
- -p 表示 SecretKey。
- startDate 为开始查询日期，endDate 为结束查询日期，startDate 需要小于 endDate。

3. 结果示例

```
{
  u'total': 2,
  u'logs': [
    {
      u'status': 1,
      u'complete_time': u'2016-08-15 10:39:16',
      u'url_list': [
        u'http://www.test.org/1.html'
      ],
      u'app_id': 1251991073,
      u'datetime': u'2016-08-15 10:39:14',
      u'host': u'www.test.org',
      u'project_id': 0,
      u'type': 0,
      u'id': 1234
    },
    ...
  ]
}
```

其中 status 为刷新状态，1 表示刷新完成。URL 刷新、目录刷新的记录均可以通过此接口查询到。

域名配置

修改缓存配置

1. 使用如下命令，您可以调用 [UpdateCache](#) 接口，修改缓存过期设置：

```
python QcloudCdnTools_V2.py UpdateCache -u xxxxx -p xxxxxxxx --hostId 1234 --cache [[0,\"all\",1000],[1,\".jpg;.js\",2000],[2,\"/www/html\",3000]]
```

2. 参数说明

- -u 表示SecretId。
- -p 表示SecretKey。
- hostId 为您需要修改缓存过期配置的域名 ID。
- cache 为目标缓存配置，注意双引号需要转义。

缓存过期设置

一个域名的缓存过期配置由若干条缓存过期配置组成，每条缓存过期配置分为三个参数，第一个为缓存类型，第二个为匹配规则，第三个为设置的过期时间，单位为秒。CDN为您提供三种类型：

- 0：全部类型，表示匹配的所有文件，为默认缓存配置。
- 1：文件类型，表示按文件后缀匹配，匹配内容示例：.jpg;.png。
- 2：文件夹类型，表示按目录匹配，匹配内容示例：/abc;/def。

3. 结果示例

```
request is success.
```

修改域名所属项目

1. 使用如下命令，您可以调用[UpdateCdnProject](#)接口，修改域名所属项目：

```
python QcloudCdnTools_V2.py UpdateCdnProject -u xxxxx -p xxxxxxxx --hostId 1234 --projectId 0
```

修改域名所属项目时，您需要知道项目对应的 ID，您可以前往[项目管理](#) 进行查看，默认项目的 ID 为 0。

2. 参数说明

- -u 表示SecretId。
- -p 表示SecretKey。
- hostId 表示要修改项目的域名对应的 ID。
- projectId 为目标项目的 ID。

3. 结果示例

```
request is success.
```

修改域名配置

1. 使用如下命令，您可以调用[UpdateCdnConfig](#) 接口修改域名配置，包括缓存过期设置、防盗链、回源 HOST、全路径缓存等：

```
python QcloudCdnTools_V2.py UpdateCdnConfig -u xxxxx -p xxxxxxxx --hostId 1234 --projectId 0 --cacheMode custom --cache [[0,\"all\",1023448]] --refer [1,[\"www.baidu.com\\\", \"www.qq.com \"]] --fwdHost www.test.org --fullUrl off
```

2. 参数说明

- -u 表示 SecretId。
- -p 表示 SecretKey。
- hostId 表示要修改配置的域名 ID。
- projectId 表示要修改的项目 ID。
- cacheMode 表示是否开启高级缓存设置。
- cache 表示缓存过期设置，可参考 UpdateCache 处说明。
- refer 表示防盗链设置。
- fwdHost 表示回源Host设置。
- fullUrl 表示是否开启全路径缓存，开启了全路径缓存，则表明过滤参数开关为关闭状态；若未开启全路径缓存，则表明过滤参数开关为开启状态。

防盗链设置说明

防盗链由两个字段组成，第一个字段标识了 refer 的类型：

- 0：不设置防盗链。
- 1：设置黑名单。
- 2：设置白名单。

第二个字段为具体的名单列表。

3. 结果示例

```
request is success.
```

域名管理

添加域名

1. 使用如下命令，您可以调用[AddCdnHost](#) 接口，新增CDN加速域名：

```
python QcloudCdnTools_V2.py AddCdnHost -u xxxxxx -p xxxxxxxx --host www.test.com --projectId 0 --hostType cname --origin 1.1.1.1
```

2. 参数说明

- -u 表示SecretId。
- -p 表示SecretKey。
- host 表示要增加的加速域名，域名需要通过工信部备案，且尚未接入过腾讯云 CDN。
- projecctId 表示要将域名增加加入的项目 ID，可进入[项目管理](#) 查看项目对应ID。

- hostType 为接入类型，若为 'cname' 则表示自有源接入；若为 'ftp' 则表示 FTP 源接入，此时源站参数会被忽略。
- origin 表示源站配置。

3. 结果示例

```
request is success.
```

下线域名

1. 使用如下命令，您可以调用 [OfflineHost](#) 接口，关闭指定域名的 CDN 加速服务：

```
python QcloudCdnTools_V2.py OfflineHost -u xxxxx -p xxxxxxxx --hostId 1234
```

2. 参数说明

- -u 表示 SecretId。
- -p 表示 SecretKey。
- 处于“已开启”状态的域名调用下线命令才能成功。
- hostId 表示要下线的域名 ID，可通过 GetHostInfoByHost 获取域名对应 ID。

3. 结果示例

```
request is success.
```

上线域名

1. 使用如下命令，您可以调用 [OnlineHost](#) 接口，开启指定域名的 CDN 加速服务：

```
python QcloudCdnTools_V2.py OnlineHost -u xxxxx -p xxxxxxxx --hostId 1234
```

2. 参数说明

- -u 表示 SecretId。
- -p 表示 SecretKey。
- 只有处于“已关闭”状态的域名调用上线命令才能够成功。
- hostId 表示要下线的域名 ID，可通过 GetHostInfoByHost 获取域名对应 ID。

3. 结果示例

```
request is success.
```

删除域名

1. 使用如下命令，您可以调用 [DeleteCdnHost](#) 接口，删除指定域名：

```
python QcloudCdnTools_V2.py DeleteCdnHost -u xxxxx -p xxxxxx -hostId 1234
```

2. 参数说明

- -u 表示SecretId。
- -p 表示SecretKey。
- 只有处于“已关闭”状态的域名调用删除命令才能够成功。
- hostId 表示要下线的域名 ID，可通过 [GetHostInfoByHost](#) 获取域名对应 ID。

日志相关

获取日志下载链接

1. 使用如下命令，您可以调用 [GenerateLogList](#) 接口，获取指定域名的 CDN 日志下载链接：

```
python QcloudCdnTools_V2.py GenerateLogList -u xxxxx -p xxxxxxxx --hostId 1234
```

2. 参数说明

- -u 表示SecretId。
- -p 表示SecretKey。
- hostId 为要查询日志下载链接的域名ID。
- 获取30天内每天的日志下载链接。

3. 结果示例

```
{
  u'now': 1471267882,
  u'list': [
    {
      u'date': u'2016-07-16',
      u'type': 0,
      u'name': u'20160716-test.com'
    },
  ],
}
```

```
{
  u'date': u'2016-07-17',
  u'link': u'http: //log-download.cdn.qcloud.com/20160717/20160717-test.com.gz?st=xYeU1vW6N9U
JlSc3hxM0lg&e=1472131882',
  u'type': 1,
  u'name': u'20160717-test.com'
},
...
]
```

若无 link 字段，则表示当天无日志数据产生。

消耗查询

查询 TOP 100 URL

1. 使用如下命令，您可以调用 [GetCdnStatTop](#) 接口，查询域名或项目的TOP 100 流量 / 带宽消耗 URL：

```
python QcloudCdnTools_V2.py GetCdnStatTop -u xxxxxxxxxxxx -p xxxxxxxxxxxx --startDate 2016-08-15 --endDate 2016-08-15 --statType bandwidth --projects 0 --hosts test.com
```

2. 参数说明

- -u 表示SecretId。
- -p 表示SecretKey。
- startDate 表示查询起始时间，若设置为 2016-08-15，则实际查询开始时间为：2016-08-15 00:00:00。
- endDate 表示查询结束时间，若设置为 2016-08-15，则实际查询结束时间为：2016-08-15 23:55:00。
- projects 表示查询的项目 ID，支持多个。
- hosts 表示查询的域名，域名所属项目必须传参，否则会导致错误，支持多个。
- statType 为排名依据，bandwidth 表示带宽，flux 表示流量。

3. 结果示例

```
{
  u'start_datetime': u'2016-08-15',
  u'url_data': [
    {
      u'name': u'test.com/uploads/20141218/1418891322.jpeg',
      u'value': 877
    },
    ...
  ]
}
```

```
{
  u'name': u'test.com/uploads/20141218/1418891825.jpeg',
  u'value': 796
},
{
  u'name': u'test.com/uploads/20141218/1418896965.jpeg',
  u'value': 706
},
...
]
```

value 为消耗值，flux 的单位为 Byte，bandwidth 的单位为 bps。

查询状态码统计

1. 使用如下命令，您可以调用[GetCdnStatusCode](#)接口，查询域名或项目的状态码统计：

```
python QcloudCdnTools_V2.py GetCdnStatusCode -u xxxxxxxxxxxx -p xxxxxxxxxxxx --startDate 2016-08-15 --endDate 2016-08-15 --projects 0 --hosts test.com
```

2. 参数说明

- -u 表示 SecretId。
- -p 表示 SecretKey。
- startDate 表示查询起始时间，若设置为 2016-08-15，则实际查询开始时间为：2016-08-15 00:00:00。
- endDate 表示查询结束时间，若设置为 2016-08-15，则实际查询结束时间为：2016-08-15 23:55:00。
- projects 表示查询的项目ID，支持多个。
- hosts 表示查询的域名，域名所属项目必须传参（projects），否则会导致错误，支持多个。

3. 结果示例

```
[
{
  u'200': [

  ],
  u'206': [

  ],
  u'304': [
```

```
],  
u'416': [  
  
],  
u'404': [  
69,  
69,  
76,  
69,  
66,  
78,  
73,  
71,  
73,  
76,  
...  
],  
u'500': [  
  
]  
}  
]
```

查询消耗统计明细

1. 使用如下命令，您可以调用[DescribeCdnHostDetailedInfo](#)接口，查询域名或项目的消耗明细：

```
python QcloudCdnTools_V2.py DescribeCdnHostDetailedInfo -u xxxxxxxxxxxx -p xxxxxxxxxxxx --startDate 2016-05-08 --endDate 2016-08-15 --projects 0 --hosts www.test.com --statType bandwidth
```

2. 参数说明

- -u 表示 SecretId。
- -p 表示 SecretKey。
- startDate 表示查询起始时间，若设置为 2016-08-15，则实际查询开始时间为：2016-08-15 00:00:00。
- endDate 表示查询结束时间，若设置为 2016-08-15，则实际查询结束时间为：2016-08-15 23:55:00。
- projects 表示查询的项目 ID，支持多个。
- hosts 表示查询的域名，域名所属项目必须传参（projects），否则会导致错误，支持多个。
- statType 为指定查询的消耗类型，flux 为流量，单位为 Byte；bandwidth 为带宽，单位为 bps。

3. 结果示例

```
{
  u'start_datetime': u'2016-08-1300: 00: 00',
  u'total_data': [
    35216,
    41875,
    42256,
    34333,
    40868,
    40906,
    38505,
    39487,
    39407,
    ...
  ],
  u'stat_type': u'flux',
  u'end_datetime': u'2016-08-1523: 55: 00',
  u'period': 5
}
```

period 为时间粒度，根据所查询时间区间长短不同，返回的时间粒度也不同。1 ~ 3 日，明细时间粒度均为 5 分钟，4 ~ 7 日时间粒度为 1 小时，8 天及以上时间粒度为 1 天。

查询消耗量统计

1. 使用如下命令，您可以调用[DescribeCdnHostInfo](#)接口，查询域名或项目的消耗统计：

```
python QcloudCdnTools_V2.py DescribeCdnHostInfo -u xxxxxxxxxxxx -p xxxxxxxxxxxx --startDate 2016-08-15 --endDate 2016-08-15 --projects 0 --hosts www.test.com --statType bandwidth
```

2. 参数说明

- -u 表示SecretId。
- -p 表示SecretKey。
- startDate 表示查询起始时间，若设置为 2016-08-15，则实际查询开始时间为：2016-08-15 00:00:00。
- endDate 表示查询结束时间，若设置为 2016-08-15，则实际查询结束时间为：2016-08-15 23:55:00。
- projects 表示查询的项目 ID，支持多个。
- hosts 表示查询的域名，域名所属项目必须传参（projects），否则会导致错误，支持多个。
- statType 为指定查询的消耗类型，flux 为流量，单位为 Byte；bandwidth 为带宽，单位为 bps。

3. 结果示例

```
{
  u'start_datetime': u'2016-08-1300: 00: 00',
  u'stat_type': u'bandwidth',
  u'end_datetime': u'2016-08-1523: 55: 00',
  u'detail_data': [
    {
      u'host_id': u'www.test.com',
      u'host_type': u'cname',
      u'host_name': u'www.test.com',
      u'host_value': 2214
    }
  ],
  u'period': 5
}
```

查询结果为指定时间区间的总量，host_type 为域名接入时的类型，cname 表示自有源接入，ftp 表示 FTP 源接入，cos 表示 COS 源接入。

日志合并工具

最近更新时间：2018-05-29 17:10:00

使用说明

此脚本用于获取指定域名、指定日期的日志数据包（日期范围为 30 天以内）。

[下载脚本](#)

运行环境：

- Python2.7
- Linux 系统

使用准备

在使用上述 Python 脚本时，您需要安装 requests 库，可使用如下命令：

```
pip install requests
```

参数说明

```
host 域名
-u SECRET_ID
-p SECRET_KEY
--day 日期
--dstpath 日志下载路径
```

- SecretId 和 SecretKey 在 [云 API 密钥](#) 处获取。
- 仅支持 30 天内日志下载。
- 默认情况下，指定日期日志存放地址为当前路径。

使用示例

```
python GetDayLog.py www.test.com -u XXXXXXXXXXXXXXXX -p XXXXXXXXXXXXXXXX --day 20161130 --dstpath /home/test/
```

使用成功后，可在指定目录下看到所需日志文件，文件名称为：

```
20161130-www.test.com.gz
```