

# vdn 对外接口定义

版本 v1.0.6.2

2018/11/01

## 目录

vdm 对外接口定义.....	1
1. api 接口通用说明.....	3
1.1 接口请求方式.....	3
1.2 公共参数.....	3
1.3 签名方法.....	3
1.4 接口地址.....	5
2. 基础数据查询接口.....	5
2.1 流量接口.....	5
2.2 带宽接口.....	7
2.3 请求数接口.....	9
3. 日志下载接口.....	10
3.1 日志命名规范.....	10
3.2 日志生成规定.....	11
3.3 日志查询接口.....	11
4. 刷新预热接口.....	13
4.1 URL 刷新接口.....	13
4.2 刷新目录接口.....	14
4.3 刷新纪录查询接口.....	15
4.4 URL 预热接口.....	17
4.5 预热记录查询接口.....	18
5. 错误码定义.....	21
5.1. 公共错误码.....	21

# 1.api 接口通用说明

## 1.1 接口请求方式

API 接口使用 POST 请求。请求和返回结果均使用 UTF-8 字符集进行编码。

## 1.2 公共参数

每个接口中都存在的参数为公共参数，包含如下：

名称	类型	描述	示例值
action	String	查询的具体接口名称	trafficquery
timestamp	Uint	当前 unix 时间戳	1516070805
nonce	Uint	随机正整数	1325467
secretId	String	标识身份的 secretId	bW2sxLUY9xjSkBBU
signature	String	请求签名，用来验证此次请求的合法性	

## 1.3 签名方法

签名 signature 通过 secretId 和 secretKey 来生成

假定 secretId= bW2sxLUY9xjSkBBU 时对应的 secretKey 为 AiX2gxbUNRm2hAmf

生成方法如下

以带宽查询接口为例，此接口所有参数如下

```
{
  'action' : 'trafficquery',
  'secretId' : 'bW2sxLUY9xjSkBBU',
  'timestamp' : 1516070805,
  'nonce' : 122324,
  'start_time' : '201801160905',
  'end_time' : '201801160910',
  'interval' : 5,
  'domains' : 'www.a.com,www.b.com',
```

```
'type' : 'all'
}
```

### 1) 对参数排序

首先对所有请求参数按参数名做字典序升序排列,可以借助编程语言中的相关排序函数来实现这一功能,如 php 中的 ksort 函数。上述示例参数的排序结果如下:

```
{
  'action' : 'trafficquery',
  'domains' : 'www.a.com,www.b.com',
  'end_time' : '201801160910',
  'interval' : 5,
  'nonce' : 122324,
  'secretId' : 'bW2sxLUY9xjSkBBU',
  'start_time': '201801160905',
  'timestamp' : 1516070805,
  'type' : 'all'
}
```

### 2) 将排序好的所有参数格式化并拼接为字符串,参数格式化成“参数名称”=“参数值”的形式,如对 action 参数,其参数名称为"action",参数值为" trafficquery",因此格式化后就为 action=trafficquery。然后将格式化后的各个参数用"&"拼接在一起,最终生成的请求字符串为:

```
action=trafficquery&domains=www.a.com,www.b.com&end_time=201801160910&interval=5&nonce=122324&secretId=bW2sxLUY9xjSkBBU&start_time=201801160905&timestamp=1516070805&type=all
```

还需要加入请求方法和请求地址的信息

此例中请求方法为 post,请求地址假定为 http://cdn.api.xxxxx.com/index.php

签名原文串的拼接规则为:

请求方法 + 请求地址 + ? + 请求字符串

则拼接后的结果为

```
posthttp://cdn.api.xxxxx.com/index.php?action=trafficquery&domains=www.a.com,www.b.com&end_time=201801160910&interval=5&nonce=122324&secretId=bW2sxLUY9xjSkBBU&start_time=201801160905&timestamp=1516070805&type=all
```

### 3) 使用 HMAC-SHA1 算法对请求字符串进行签名,然后将生成的签名串使用 Base64 进行编码,即可获得最终的签名串。

```
$secretKey = 'AiX2gxbUNRm2hAmf';
```

```
$srcStr =
```

```
'posthttp://cdn.api.xxxxx.com/index.php?action=trafficquery&domains=www.a.com,www.b.com&end_time=201801160910&interval=5&nonce=122324&secretId=bW2sxLUY9xjSkBBU&start_time=201801160905&timestamp=1516070805&type=all';
```

```
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
```

```
echo $signStr
```

- 4) 假定上面生成的签名串为 ahQ8U0kWSHfYzVODxFVR4c5YJSg=  
 则对签名串做 URL 编码后为 ahQ8U0kWSHfYzVODxFVR4c5YJSg%3D  
 最终得到的签名串请求参数(signature)为: ahQ8U0kWSHfYzVODxFVR4c5YJSg%3D

## 1.4 接口地址

对接将提供接口查询地址

每个接口的地址在每个接口的请求示例中进行了提供

## 2. 基础数据查询接口

基础数据接口提供 5 分钟粒度数据。时间点均为前指向。

- 1) 5 分钟粒度指 2018-01-16 09:05 代表 [2018-01-16 09:00:00, 2018-01-16 09:04:59 ]。
- 2) 所有基础数据查询类接口，均支持多域名同时查询，并且分开返回不同域名的数据（不会合并域名数据）。
- 3) 返回的结果需要使用 json 格式返回

### 2.1 流量接口

说明：查询指定域名在指定时间段的流量信息

请求参数

名称	类型	描述	示例值
action	String	流量查询接口	trafficquery
domains	String	待查询的域名,多个域名用逗号分隔	www.a.com,www.b.com
start_time	String	查询起始时间, 格式 YYYYmmddHHii	201801160905
end_time	String	查询结束时间, 格式 YYYYmmddHHii, 起始和结束时间可以相同	201801160910
interval	int	时间粒度固定为 5	5
type	String	固定为 e	e

## 返回参数

名称	类型	描述	示例值
code	int	公共错误码, 0 表示成功, 其他值表示失败。详见公共错误码	0
message	String	错误信息描述, 与接口相关	
secretId	String	标识身份的 secretId	bW2sxLUY9xjSkBBU
interval	int	时间粒度固定为 5	5
data	object	结果数据, 详见下文说明	

## 流量, 单位 B

## 请求示例

POST 请求, 参数填充在 HTTP Request-body 中, 请求地址:

<http://openapi.vdncloud.com/basicDataQuery/trafficquery>

参数数组如下

```
array (
  'action' => 'trafficquery',
  'secretId' => 'XXXXXXXXXXXX',
  'timestamp' => 1516070805,
  'nonce' : 122324,
  'signature' => 'XXXXXXXX',
  'start_time' => '201801160905',
  'end_time' => '201801160910',
  'interval' => 5,
  'domains' => 'www.a.com,www.b.com',
  'type' => 'e'
)
```

返回结果示例:

```
{
  "code": 0,
  "message": "",
  "secretId": "bW2sxLUY9xjSkBBU",
  "action": "trafficquery",
  "interval": 5,
  "data": {
    "e": {
      "www.a.com": {
        "1516759200": 33664135068,
        "1516765200": 33040029888,

```



带宽, 单位 bit/s

请求示例

POST 请求, 参数填充在 HTTP Request-body 中, 请求地址:

<http://openapi.vdncloud.com/basicDataQuery/bandwidthquery>

参数数组如下

```
array (  
  'action' => 'bandwidthquery',  
  'secretId' => 'XXXXXXXXXXXX',  
  'timestamp' => 1516070805,  
  'nonce' : 122324,  
  'signature' => 'XXXXXXXX',  
  'start_time' => '201801160905',  
  'end_time' => '201801160910',  
  'interval' => 5,  
  'domains' => 'www.a.com,www.b.com',  
  'type' => 'e'  
)
```

返回结果示例:

```
{  
  "code": 0,  
  "message": "",  
  "secretId": "bW2sxLUY9xjSkBBU",  
  "action": "bandwidthquery",  
  "interval": 5,  
  "data": {  
    "e": {  
      "www.a.com": {  
        "1516759200": 33664135068,  
        "1516765200": 33040029888,  
        "1516763400": 33225771458  
      },  
      "www.b.com": {  
        "1516759200": 21135768214,  
        "1516763400": 25372480085,  
        "1516765200": 21522411024  
      }  
    }  
  }  
}
```

## 2.3 请求数接口

说明：查询指定域名在指定时间段的请求数信息

请求参数

名称	类型	描述	示例值
action	String	请求数查询接口	pvquery
domains	String	待查询的域名,多个域名用逗号分隔	www.a.com,www.b.com
start_time	String	查询起始时间, 格式 YYYYmmddHHii	201801160905
end_time	String	查询结束时间, 格式 YYYYmmddHHii, 起始和结束时间可以相同	201801160910
interval	int	时间粒度固定为 5	5
type	String	固定为 e	e

返回参数

名称	类型	描述	示例值
code	int	公共错误码, 0 表示成功, 其他值表示失败。详见公共错误码	0
message	String	错误信息描述, 与接口相关	
secretId	String	标识身份的 secretId	bW2sxLUY9xjSkBBU
interval	int	时间粒度固定为 5	5
data	object	结果数据, 详见下文说明	

请求数, 单位个

请求示例

POST 请求, 参数填充在 HTTP Request-body 中, 请求地址:

<http://openapi.vdncloud.com/basicDataQuery/pvquery>

参数数组如下

```
array (
  'action' => 'pvquery',
  'secretId' => 'XXXXXXXXXXXX',
  'timestamp' => 1516070805,
  'nonce' : 122324,
  'signature' => 'XXXXXXXX',
  'start_time' => '201801160905',
```

```
'end_time' => '201801160910',  
'interval' => 5,  
'domains' => 'www.a.com,www.b.com',  
'type' => 'e'  
)
```

返回结果示例:

```
{  
  "code": 0,  
  "message": "",  
  "secretId": "bW2sxLUY9xjSkBBU",  
  "action": "pvquery",  
  "interval": 5,  
  "data": {  
    "e": {  
      "www.a.com": {  
        "1516759200": 33664135068,  
        "1516765200": 33040029888,  
        "1516763400": 33225771458  
      },  
      "www.b.com": {  
        "1516759200": 21135768214,  
        "1516763400": 25372480085,  
        "1516765200": 21522411024  
      }  
    }  
  }  
}
```

## 3. 日志下载接口

### 3.1 日志命名规范

日志需要按照域名打包

命名如下

域名\_日志日期\_日志起时间\_日志止时间\_分片序号.gz

举例:

www.test.com\_20180115\_1200\_1300\_000.gz

说明:

日志时间:示例中的 20180115\_1200\_1300 部分,是日志文件内包含的访问时间范

围, 20180115\_1200\_1300 代表 2018/1/15 12:00:00 到 2018/1/15 12:59:59 间产生的日志。\_000 代表日志分片的序号, 为三位数字。

超时收集到的数据不再放入正常的日志包, 而是放入每小时一个的超时日志包, 超时时间可由双方商定, 超时日志包命名如下

域名\_日志日期\_日志起时间\_日志止时间\_timeout\_分片序号.gz

注意: 只比正常日志多了 timeout 这个信息, 但是日志内容并不是日志起时间和日志止时间之间产生的日志, 而是在日志起时间和日志止时间之间收集到的超时日志。

## 3.2 日志生成规定

日志生成间隔以域名为单位确定, 一种域名的下载日志生成间隔有以下几种可能

Log 生成时间间隔选项, 24 小时到 1 分钟内的整分钟数均可能, 目前仅支持为 60 分钟, 如果生成间隔为 24 小时, 则日志时间表示示例如下 20180115\_0000\_0000 具体域名下载日志的生成间隔需要提前与我司沟通确定。

超时日志的生成间隔为 1 小时。

## 3.3 日志查询接口

日志查询接口要符合 api 接口通用说明, 但是可能与基础数据查询接口是不同的地址

日志查询接口地址为

<http://openapi.vdncloud.com/basicDataQuery/logquery>

说明: 查询指定域名在指定时间段的日志下载地址

请求参数

名称	类型	描述	示例值
action	String	日志下载查询接口	logquery
domains	String	待查询的域名, 多个域名用逗号分隔	www.a.com,www.b.com
start_time	String	查询起始时间, 格式 YYYYmmddHHii	201801160900
end_time	String	查询结束时间, 格式 YYYYmmddHHii, 起始和结束时间可以相同	201801161000
type	String	查询类型, e 表示边缘数据, et 表示边缘超时数据	e

返回参数

名称	类型	描述	示例值
code	int	公共错误码, 0 表示成功, 其他值表示失败。详见公共错误码	0
message	String	错误信息描述, 与接口相关	
secretId	String	标识身份的 secretId	bW2sxLUY9xjSkBBU
data	object	结果数据, 详见下文说明	

日志下载链接后的值为日志文件的大小

请求示例

POST 请求, 参数填充在 HTTP Request-body 中, 请求地址:

<http://openapi.vdncloud.com/basicDataQuery/logquery>

参数数组如下

```
array (
  'action' => 'logquery',
  'secretId' => 'XXXXXXXXXXXX',
  'timestamp' => 1516070805,
  'nonce' : 122324,
  'signature' => 'XXXXXXXX',
  'start_time' => '201801160900',
  'end_time' => '201801161000',
  'domains' => 'www.a.com,www.b.com',
  ' type' => 'e'
)
```

返回结果示例:

```
{
  "code": 0,
  "message": "",
  "secretId": "bW2sxLUY9xjSkBBU",
  "action": "logquery",
  "data": {
    "e": {
      "www.a.com": {
        "http://cdn.api.xxxxx.com/www.a.com_20180116_0900_1000_000.gz":
          53341343,
        "http:
//cdn.api.xxxxx.com/www.a.com_20180116_0900_1000_000.gz": 5456621
      }
    }
  }
}
```



<http://openapi.vdncloud.com/refreshAndPush/RefreshCdnUrl>

参数数组如下

```
array (
  'action': 'RefreshCdnUrl',
  'secretId' : 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX',
  'timestamp' : 1462864833,
  'nonce' : 1149033341,
  'signature' : 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX',
  'urls' : 'https://www.test.com/1.jpg, https://www.test.com/2.jpg '
)
```

返回结果示例:

```
{
  "code": 0,
  "message": "",
  "data": {
    "count": 1,
    "task_id": "1480069888795584532"
  }
}
```

## 4.2 刷新目录接口

说明: 用于将 CDN 全网节点上指定目录下的资源设置为过期。

提交目录的 URL 必须以 http:// 或 https:// 开头

目录刷新会比对节点内容与源站内容的 Last-Modify, 若未变化, 则不会进行资源刷新  
调用频率限制为 100ms

请求参数

名称	类型	描述
action	String	刷新目录接口
dirs	String	需要刷新的 URL: 支持一个或多个目录刷新, 多个请以逗号分隔

返回参数

名称	类型	描述	示例值
code	int	公共错误码, 0 表示成功, 其他值表示失败。详见公共错误码	0
message	String	错误信息描述, 与接口相关	

请求示例

POST 请求，参数填充在 HTTP Request-body 中，请求地址：

<http://openapi.vdncloud.com/refreshAndPush/RefreshCdnDir>

参数数组如下

```
array (
  'action' : 'RefreshCdnDir',
  'secretId' : 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
  'timestamp' : 1462864833,
  'nonce' : 1149033341,
  'signature' : 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
  'dirs' : 'https://www.test.com/test/'
)
```

返回结果示例：

```
{
  "code": 0,
  "message": ""
}
```

## 4.3 刷新纪录查询接口

说明：查询提交的刷新 URL、刷新目录任务执行情况

刷新纪录保存 30 天

支持精确到秒的时间区间查询

支持根据提交时反馈的任务 taskId 进行查询

支持根据域名查询刷新记录

支持根据 url 查询刷新记录

调用频率限制为 100ms

请求参数

名称	是否必填	类型	描述
action	是	String	刷新纪录查询接口
start_time	否	String	查询开始时间，如 201711290000
end_time	否	String	查询结束时间，如 201711290005
taskId	是	String	根据提交刷新 URL 任务返回的 task_id 查询
host	否	String	需要查询的域名

url	否	String	需要查询的 URL(可为空)
-----	---	--------	----------------

## 返回参数

名称	类型	描述
code	Int	公共错误码, 0 表示成功, 其他值表示失败。详见错误码
message	String	模块错误信息描述, 与接口相关。
data	Object	返回结果数据

## 详细说明

## data

名称	类型	描述
logs	Array	预热任务详情, 具体说明见下文

## logs

名称	类型	描述
id	Int	编号
app_id	Int	用户 APP ID
host	String	域名
type	Int	刷新类型 0: 代表 URL 刷新 1: 代表目录刷新
status	Int	刷新结果 1: 表示刷新成功 0: 表示刷新中 若为负数, 表示刷新失败
url_list	Array	本次刷新时提交的 URL 列表
datetime	String	提交时间

## 请求示例

POST 请求, 参数填充在 HTTP Request-body 中, 请求地址:

<http://openapi.vdmcloud.com/refreshAndPush/GetCdnRefreshLog>

参数数组如下

```
array (
  'action' : 'GetCdnRefreshLog',
  'secretId' : 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX',
  'timestamp' : 1462782282,
  'nonce' : 123456789,
  'signature' : 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX',
  'start_time' : '201702061900',
  "end_time" : '201702061940'
)
```

返回结果示例:

```
{
  "code": 0,
  "message": "",
  "data": {
    "logs": [
      {
        "id": 6182538,
        "app_id": 123456,
        "host": "www.test.com",
        "type": 0,
        "status": 1,
        "url_list": [
          "http://www.test.com/Content/image/test.png"
        ],
        "datetime": "2017-02-06 19:39:59"
      }
    ],
    "total": 1
  }
}
```

## 4.4 URL 预热接口

说明: 将指定 URL 资源主动推送至 CDN 节点。

提交的 URL 必须以 http:// 或 https:// 开头

预热会导致回源带宽较高, 请根据源站带宽来拆分提交预热任务

调用频率限制为 100ms

请求参数

名称	是否必填	类型	描述
action	是	String	URL 预热接口
urls	是	String	需要进行预热的资源 URL 列表, 多个请以逗号分隔

返回参数

名称	类型	描述
code	Int	公共错误码, 0 表示成功, 其他值表示失败。详见错误码
message	String	模块错误信息描述, 与接口相关。
data	Array	返回结果数据

详细说明

data

名称	类型	描述
task_ids	Object	提交的任务 ID 信息

task\_ids

名称	类型	描述
task_id	Int	提交的任务 ID
date	String	提交任务的日期

请求示例

POST 请求，参数填充在 HTTP Request-body 中，请求地址：

<http://openapi.vdmcloud.com/refreshAndPush/CdnPusherV2>

参数数组如下

```
array (
  'action' : 'CdnPusherV2',
  'secretId' : 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
  'timestamp' : 1462782282,
  'nonce' : 123456789,
  'signature' : 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
  'urls' : 'http://www.test.com/1.jpg'
)
```

返回结果示例：

```
{
  "code": 0,
  "message": "",
  "data": {
    "task_ids": [
      {
        "task_id": 20860,
        "date": "20161013"
      }
    ]
  }
}
```

## 4.5 预热记录查询接口

说明：查询提交的预热任务执行状态。

预热记录保存 30 天

支持指定时间区间查询  
 支持根据提交时反馈的任务 taskId 进行查询  
 支持指定多个域名查询  
 支持指定任务状态、指定关键字查询  
 支持分页查询  
 调用频率限制为 100ms

## 请求参数

名称	是否必填	类型	描述
start_time	否	String	指定查询起始时间, 如 201711290000
end_time	否	String	指定查询结束时间, 如 201711292000
taskId	是	Int	预热任务提交时返回的 ID
hosts	否	String	指定域名查询, 多个以逗号间隔
status	否	String	预热任务状态 "init": 表示状态为排队中, 尚未开始 "process": 表示预热任务执行中 "done": 表示预热任务已经完成 可指定上述状态中一个或多个进行查询, 多个以逗号间隔
keyword	否	String	关键字查询
offset	否	Int	查询偏移量, 用于分页
limit	否	Int	返回记录数目, 用于分页

## 返回参数

名称	类型	描述
code	Int	公共错误码, 0 表示成功, 其他值表示失败。详见错误码
message	String	模块错误信息描述, 与接口相关。
data	Array	返回结果数据

## 详细说明

## data

名称	类型	描述
logs	Object	预热任务详情, 具体说明见下文

total	Int	查询到的预热 URL 总次数
-------	-----	----------------

## task\_ids

名称	类型	描述
log_id	Int	该条操作记录对应的 ID
task_id	Int	预热任务提交时返回的 ID
host	String	域名, 预热 URL 对应的域名
datetime	String	预热任务提交的时间
status	String	预热任务当前状态 "inprocess": 预热任务执行中 "done": 预热任务已经完成 "fail": 表示预热任务失败
url	String	预热的 URL

请求示例

POST 请求, 参数填充在 HTTP Request-body 中, 请求地址:

<http://openapi.vdncloud.com/refreshAndPush/GetPushLogs>

参数数组如下

```
array (
  'action' : 'GetPushLogs',
  'secretId' : 'XXXXXXXXXXXXXXXXXXXXXXXXXXXX',
  'timestamp' : 1462782282,
  'nonce' : 123456789,
  'signature' : 'XXXXXXXXXXXXXXXXXXXXXXXXXX',
  'start_time' : '201711290000',
  'end_time' : '201711292100'
)
```

返回结果示例:

```
{
  "code": 0,
  "message": "",
  "data": {
    "logs": [
      {
        "log_id": 85291,
        "task_id": 67396,
        "host": "www.test.com",
        "datetime": "2017-11-29 20:40:13",
        "status": "process",
        "url": "http://www.test.com/1.jpg"
      }
    ],
    "total": 1
  }
}
```

```
}  
}
```

## 5. 错误码定义

### 5.1. 公共错误码

返回结果中的错误码表示了用户调用 API 的结果。其中，code 为公共错误码，其适用于所有模块的 API 接口。若 code 为 0，表示调用成功，否则，表示调用失败。当调用失败后，用户可以根据下表确定错误原因并采取相应措施。

错误代码	错误类型	描述
1000	请求参数非法	缺少必要参数，或者参数值格式不正确，具体错误信息请查看错误描述 message 字段。
1100	鉴权失败	签名鉴权失败，请参考文档中鉴权部分。
1200	请求过期	请求已经过期，请参考文档中请求有效期部分。
1300	拒绝访问	帐号被封禁，或者不在接口针对的用户范围内等。
2000	服务器内部错误	服务器内部出现错误，请稍后重试或者联系客服人员帮忙解决。