

# 信鸽 REST API 用户手册

## 1 协议整体描述

请求 URL 结构为：

```
http://接口域名/v2/class/method?params
```

表 1 url 结构字段解释

字段名	用途	备注
接口域名	接口域名	统一使用 openapi.xg.qq.com
v2	表示当前 api 的版本号	
class	提供的接口类别	
method	每个接口大类提供的具体操作接口	如查询、设置、删除等
params	以 GET 方式调用接口时传递的参数	包括通用参数和 api 相关特定参数。所有的参数都必须为 utf8 编码，params 字符串应进行 url encode

注：以 POST 方式调用接口时，参数应以 POST 参数形式传递，内容要求同 params 字段。HTTP HEADER 中 “Content-type” 字段要设置为 “application/x-www-form-urlencoded”

### 1.1 通用参数

各接口 url 结构的 params 字段有共同参数，参数列表如下

表 2 通用参数表

参数名	类型	是否必要	参数描述
access_id	uint	是	应用的唯一标识符，在提交应用时管理系统返回

timestamp	uint	是	本请求的 unix 时间戳 ,用于确认请求的有效期。默认情况下 , 请求时间戳与服务器时间 ( 北京时间 ) 偏差大于 600 秒则会被拒绝
valid_time	uint	否	配合 timestamp 确定请求的有效期 ,单位为秒 ,最大值为 600。 若不设置此参数或参数值非法 , 则按默认值 600 秒计算有效期
sign	string	是	<p>内容签名。生成规则 :</p> <p>A ) 提取请求方法 method ( GET 或 POST );</p> <p>B ) 提取请求 url 信息 ,包括 Host 字段的 IP 或域名和 URI 的 path 部分 , 注意不包括 Host 的端口和 Path 的 querystring。请在请求中带上 Host 字段 ,否则将视为无效请求。比如</p> <p>openapi.xg.qq.com/v2/push/single_device 或者 10.198.18.239/v2/push/single_device;</p> <p>C ) 将请求参数 ( 不包括 sign 参数 ) 格式化成 K=V 方式 , 注意 : 计算 sign 时所有参数不应进行 urlencode ;</p> <p>D ) 将格式化后的参数以 K 的字典序升序排列 ,拼接在一起 , <b>注意字典序中大写字母在前 ;</b></p> <p>E ) 拼接请求方法、url、排序后格式化的字符串以及应用的 secret_key ;</p> <p>F ) 将 E 形成字符串计算 MD5 值 , 形成一个 32 位的十六进制 ( 字母小写 ) 字符串 ,即为本次请求 sign ( 签名 ) 的值 ;</p> <p>Sign=MD5 (\$http_method\$url\$kl=\$v1\$k2=\$v2\$sec</p>

		<p>ret_key);</p> <p>该签名值基本可以保证请求是合法者发送且参数没有被修改，但无法保证不被偷窥。</p> <p>例如：</p> <p>POST 请求到接口</p> <p>http://openapi.xg.qq.com/v2/push/single_device，有四个参数，access_id=123，</p> <p>timestamp=1386691200，Param1=Value1，</p> <p>Param2=Value2，secret_key 为 abcde。则上述 E 步骤</p> <p>拼接出的字符串为</p> <p>POSTopenapi.xg.qq.com/v2/push/single_device</p> <p>Param1=Value1Param2=Value2access_id=123timestamp=1386691200abcde，注意字典序中大写在前。计算出该字符串的 MD5 为</p> <p>ccaefecaef6be07493cfe75ebc43b7d53，以此作为</p> <p>sign 参数的值</p>
--	--	---

## 1.2 通用返回结果

返回结果的例子如下：

<pre>{   "ret_code":0,   "err_msg":"ok",   "result":{"status":0} }</pre>
--

返回结果字段定义如下：

表 3 返回结果字段表

参数名	类型	是否必要	参数描述
ret_code	int	是	返回码
err_msg	string	否	请求出错时的错误信息
result	json	否	请求正确时，若有额外数据要返回，则结果封装在该字段的 json 中。若无额外数据，则可能无此字段

特别注意 :1 )参数和值都是大小写敏感 ,如没有特别注明 ,都是小写 2 )所有的 K 和 V 须 urlencode ,  
避免里面有 "&" 或 "=" 之类字符影响解析

## 1.3 通用返回码描述

通用返回码含义如下：

值	含义
0	调用成功
-1	参数错误，请对照错误提示和文档检查请求参数
-2	请求时间戳不在有效期内
-3	sign 校验无效，检查 access id 和 secret key（注意不是 access key）
2	参数错误，请对照文档检查请求参数
40	推送的 token 没有在信鸽中注册，请检查终端注册是否成功
48	推送的账号没有在信鸽中注册，请检查终端注册是否成功
73	消息字符数超限，请减少消息内容再试
76	请求过于频繁，请稍后再试
78	循环任务参数错误

100	APNS 证书错误。请重新提交正确的证书
其他	内部错误

## 1.4 推送消息定义

所有推送接口中的 message 参数定义是相同的，故提出在本节描述。

### 1.4.1 推送目标平台为 Android

message 参数值应为如下所述的 json 字符串，其总长度不能超过 4096 字节。

#### 1.4.1.1 通知定义

一个简单的例子：

```
{"content":"this is content","title":"this is title", "vibrate":1}
```

完整定义：

```
{
  "title ":"xxx", // 标题，必填

  "content ":"xxxxxxxxx", // 内容，必填

  "accept_time": //表示消息将在哪些时间段允许推送给用户，选填
  [
    {
      "start":{"hour":"13","min":"00"},
      "end": {"hour":"14","min":"00"}
    },
    {
      "start":{"hour":"00","min":"00"},
      "end": {"hour":"09","min":"00"}
    }
  ],
}
```

```

"n_id":0, //通知 id, 选填。若大于 0, 则会覆盖先前弹出的相同 id 通知; 若为 0, 展示
本条通知且不影响其他通知; 若为-1, 将清除先前弹出的所有通知, 仅展示本条通知。默认为 0

"builder_id":0, // 本地通知样式, 必填

"ring":1, // 是否响铃, 0 否, 1 是, 下同。选填, 默认 1

"ring_raw":"ring", // 指定应用内的声音 ( ring.mp3 ), 选填

"vibrate":1, // 是否振动, 选填, 默认 1

"lights":1// 是否呼吸灯, 0 否, 1 是, 选填, 默认 1

"clearable":1, // 通知栏是否可清除, 选填, 默认 1

"icon_type":0 //默认 0, 通知栏图标是应用内图标还是上传图标, 0 是应用内图标, 1 是上
传图标, 选填

"icon_res":"xg", // 应用内图标文件名 ( xg.png ) 或者下载图标的 url 地址, 选填

"style_id":1 //Web 端设置是否覆盖编号的通知样式, 默认 1, 0 否, 1 是, 选填

"small_icon":"xg"指定状态栏的小图片 (xg.png), 选填

"action":{ // 动作, 选填。默认为打开 app

    "action_type ": 1, // 动作类型, 1 打开 activity 或 app 本身, 2 打开浏览
器, 3 打开 Intent

    "activity ": "xxx"

    "aty_attr ": // activity 属性, 只针对 action_type=1 的情况
    {

        "if":0, // 创建通知时, intent 的属性, 如 :
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);

        "pf":0, // PendingIntent 的属性, 如 :
PendingIntent.FLAG_UPDATE_CURRENT

```

```

    }

    "browser": {"url": "xxxx ", "confirm": 1}, // url : 打开的 url ,
confirm 是否需要用户确认

    "intent": "xxx"
  },

  "custom_content": { // 用户自定义的 key-value , 选填

    "key1": "value1",
    "key2": "value2"
  }
}

```

### 1.4.1.2透传消息定义

一个简单的例子：

```

{"content": "this is content", "title": "this is title"}

```

完整定义：

```

{

  "title": "xxx", // 标题, 选填

  "content": "xxxxxxxxxx", // 内容, 选填

  "accept_time": // 表示消息将在哪些时间段允许推送给用户, 选填
  [
    {
      "start": {"hour": "13", "min": "00"},
      "end": {"hour": "14", "min": "00"}
    },
    {
      "start": {"hour": "00", "min": "00"},
      "end": {"hour": "09", "min": "00"}
    }
  ],
}

```

```
"custom_content":{ // 用户自定义的 key-value , 选填
    "key1": "value1",
    "key2": "value2"
}
}
```

## 1.4.2 推送目标平台为 iOS

message 参数应为 APNS 规定的 payload ( 也是一个 json 字符串 ), 详细定义参考 APNS 官方手册 [https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html#//apple\\_ref/doc/uid/TP40008194-CH100-SW1](https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html#//apple_ref/doc/uid/TP40008194-CH100-SW1) , 信鸽在其基础上仅增添了两保留字段 xg 和 accept\_time。payload 不能超过 256 字节。需要注意的是 accept\_time 字段不会传递给 APNS , 因此不占用 payload 容量。

**一个简单的例子：**

```
{"aps":{"alert":"gogogo"}}
```

**一个复杂点的例子：**

```
{
    "aps" : { // apns 规定的 key-value
        "alert" : {
            "body" : "Bob wants to play poker",
            "action-loc-key" : "PLAY"
        },
        "badge" : 5,
    },
    "accept_time":[ //允许推送给用户的时段 , 选填。accept_time 不会占用 payload
容量
        {
            "start":{"hour":"13","min":"00"},
            "end": { "hour":"14","min":"00"}
        },
    ],
}
```



```
{
  "start":{"hour":"00","min":"00"},
    "end": {"hour":"09","min":"00"}
}

] // 仅 0~9 点和 13~14 点这两个时段可推送

"custom1" : "bar",    // 合法的自定义 key-value , 会传递给 app

"custom2" : [ "bang",  "whiz" ],  // 合法的自定义 key-value , 会传递给 app

"xg" : "oops"  // 错误! xg 为信鸽系统保留 key , 其 value 会被信鸽系统覆盖, 应避免使用

}
```

## 2 各接口定义

### 2.1 Push 消息（包括通知和透传消息）给单个设备

**url 路径：**

```
http://接口域名/v2/push/single_device?params
```

**请求参数：**除了通用参数外，还包括如下参数

表 4 push 消息给单个设备请求参数列表

参数名	类型	必需	默认值	参数描述
device_token	string	是	无	针对某一设备推送
message_type	uint	是	无	消息类型：1：通知 2：透传消息。iOS 平台请填写 0
message	string	是	无	参见“推送消息定义”一节
expire_time	uint	否	不存储	消息离线存储多久，单位为秒，最长存储时间 3 天。设为 0，则不存储

send_time	string	否	立即	指定推送时间，格式为 year-mon-day hour:min:sec 若小于服务器当前时间，则会立即推送；若为循环任务，则该字段标识任务的起始时间
multi_pkg	uint	否	0	0 表示按注册时提供的包名分发消息；1 表示按 access id 分发消息，所有以该 access id 成功注册推送的 app 均可收到消息。本字段对 iOS 平台无效
environment	uint	仅 iOS 必需	无	向 iOS 设备推送时必须填，1 表示推送生产环境；2 表示推送开发环境。推送 Android 平台不填或填 0

**注意**

**响应结果：**在通用返回结果参数中，result 字段的 json 为空。本接口不返回 push id

**2.2 Push 消息（包括通知和透传消息）给单个账户或别名**

设备的账户或别名由终端 SDK 在调用推送注册接口时设置，详情参考终端 SDK 文档。

**url 路径：**

http://接口域名/v2/push/single\_account?params

**请求参数：**除了通用参数外，还包括如下参数

表 5 push 消息给单个账户或别名请求参数列表

参数名	类型	必需	默认值	参数描述
account	string	是	无	针对某一账号推送
message_type	uint	是	无	消息类型：1：通知 2：透传消息
message	string	是	NA	参见“推送消息定义”一节
expire_time	uint	否	不存储	消息离线存储多久，单位为秒，最长存储时间 3 天。

				设为 0，则不存储
send_time	string	否	立即	指定推送时间 格式为 year-mon-day hour:min:sec 若小于服务器当前时间，则会立即推送
multi_pkg	uint	否	0	0 表示按注册时提供的包名分发消息；1 表示按 access id 分发消息，所有以该 access id 成功注册推送的 app 均可收到消息。
environment	uint	仅 iOS 必需	无	向 iOS 设备推送时必须填，1 表示推送生产环境；2 表示推送开发环境。推送 Android 平台不填或填 0

**响应结果：**在通用返回结果参数中，result 字段的 json 为空。本接口不返回 push id

## 2.3 Push 消息（包括通知和透传消息）给多个账户或别名

如果推送目标帐号数量很大（比如 $\geq 10000$ ），推荐使用 [account\\_list\\_multiple](#) 接口，用户请自行比较异同。

设备的账户或别名由终端 SDK 在调用推送注册接口时设置，详情参考终端 SDK 文档。

**url 路径：**

http://接口域名/v2/push/account\_list?params

**请求参数：**除了通用参数外，还包括如下参数

表 6 push 消息给多个账户或别名请求参数列表

参数名	类型	必需	默认值	参数描述
account_list	string	是	无	由 account 组成的 json 数组，单次发送 account 不超过 100 个。

message_type	uint	是	无	消息类型：1：通知 2：透传消息
message	string	是	NA	参见“推送消息定义”一节
expire_time	uint	否	不存储	消息离线存储多久，单位为秒，最长存储时间 3 天。 设为 0，则不存储
multi_pkg	uint	否	0	0 表示按注册时提供的包名分发消息；1 表示按 access id 分发消息，所有以该 access id 成功注册推 送的 app 均可收到消息。
environment	uint	仅 iOS 必需	无	向 iOS 设备推送时必填，1 表示推送生产环境；2 表 示推送开发环境。推送 Android 平台不填或填 0

**响应结果：**在通用返回结果参数中，result 字段的 json 为每个 account 发送返回码。本接口不返回 push id

## 2.4 Push 消息（包括通知和透传消息）给 app 的所有设备

后台对本接口的调用频率有限制，两次调用之间的时间间隔不能小于 3 秒。

**url 路径：**

http://接口域名/v2/push/all\_device?params

**请求参数：**除了通用参数外，还包括如下参数

表 6 push 消息给应用所有用户的请求参数列表

参数名	类型	必需	默认值	参数描述
message_type	uint	是	无	消息类型：1：通知 2：透传消息。iOS 平台请填写 0
message	string	是	NA	参见“推送消息定义”一节

expire_time	uint	否	不存储	消息离线存储多久，单位为秒，最长存储时间 3 天。设为 0，则不存储
send_time	string	否	立即	指定推送时间,格式为 year-mon-day hour:min:sec 若小于服务器当前时间,则会立即推送;若为循环任务,则该字段标识任务的起始时间
multi_pkg	uint	否	0	0 表示按注册时提供的包名分发消息;1 表示按 access id 分发消息,所有以该 access id 成功注册推送的 app 均可收到消息。本字段对 iOS 平台无效
environment	uint	仅 iOS 必需	无	向 iOS 设备推送时必填,1 表示推送生产环境;2 表示推送开发环境。推送 Android 平台不填或填 0
loop_times	uint	否	无	循环任务执行的次数,取值[1, 15]
loop_interval	uint	否	无	循环任务的执行间隔,以天为单位,取值[1, 14]。 loop_times 和 loop_interval 一起表示任务的生命周期,不可超过 14 天

**响应结果：**在通用返回结果参数中，result 字段的 json 示例如下

```
{
  "push_id":string (表示给 app 下发的任务 id，如果是循环任务，返回的是循环父任务 id)
}
```

## 2.5 Push 消息（包括通知和透传消息）给 tags 指定的设备

**url 路径：**

http://接口域名/v2/push/tags\_device?params

**请求参数：**除了通用参数外，还包括如下参数

表 7 push 消息给应用所有用户的请求参数列表

参数名	类型	必需	默认值	参数描述
message	string	是	无	参见“推送消息定义”一节
message_type	uint	是	无	消息类型：1：通知 2：透传消息。iOS 平台请填写 0
tags_list	json	是	无	["tag1","tag2","tag3"]
tags_op	string	是	无	取值为 AND 或 OR
expire_time	uint	否	不存储	消息离线存储多久，单位为秒，最长存储时间 3 天。设为 0，则不存储
send_time	string	否	立即	指定推送时间,格式为 year-mon-day hour:min:sec 若小于服务器当前时间，则会立即推送
multi_pkg	uint	否	0	0 表示按注册时提供的包名分发消息；1 表示按 access id 分发消息，所有以该 access id 成功注册推送的 app 均可收到消息。本字段对 iOS 平台无效
environment	uint	仅 iOS 必需	无	向 iOS 设备推送时必填，1 表示推送生产环境；2 表示推送开发环境。推送 Android 平台不填或填 0
loop_times	uint	否	无	循环任务执行的次数，取值[1, 15]
loop_interval	uint	否	无	循环任务的执行间隔，以天为单位，取值[1, 14]。 loop_times 和 loop_interval 一起表示任务的生命周期，不可超过 14 天

**响应结果：**在通用返回结果参数中，result 字段的 json 示例如下

```
{
  "push_id":string (表示给 app 下发的任务 id，如果是循环任务，返回的是循环父任务 id)
}
```

## 2.6 查询群发消息发送状态

url 路径：

http://接口域名/v2/push/get\_msg\_status?params

请求参数：除了通用参数外，还包括如下参数

表 8 查询消息发送状态请求参数列表

参数名	类型	必需	默认值	参数描述
push_ids	json	是	[]	[ {“push_id”: string}, {“push_id”: “xxxx” }, ]

响应结果：在通用返回结果参数中，result 字段的 json 形式为：

```
{
  "list": [
    {
      "push_id": "27ABC5486977"

      "status": 0（未处理）/1（推送中）/2（推送完成）/3（推送失败）

      "start_time": " year-mon-day hour:min:sec "

      "finished":xxxx （已发送）

      "total":xxxxx （共需要发送）
    },
  ]
}
```

## 2.7 查询应用覆盖的设备数

url 路径：

http://接口域名/v2/application/get\_app\_device\_num?params

**请求参数：**本接口仅包括公共参数

**响应结果：**在通用返回结果参数中，result 字段的 json 形式为：

```
{
  "device_num": 34567(设备数)
}
```

若请求应用列表中某个应用信息非法，则不会在 result 中返回结果

## 2.8 查询应用设置的标签

**url 路径：**

http://接口域名/v2/tags/query\_app\_tags?params

**请求参数：**除了通用参数外，还包括如下参数

表 9 查询消息发送状态请求参数列表

参数名	类型	必需	默认值	参数描述
start	uint	否	0	开始值
limit	uint	否	100	限制数量

**响应结果：**在通用返回结果参数中，result 字段的 json 格式如下

```
{
  "total": 2, //应用的 tag 总数，注意不是本次查询返回的 tag 数
  "tags":["tag1","tag2"]
}
```



## 2.9 取消尚未触发的定时群发任务

url 路径：

http://接口域名/v2/push/cancel\_timing\_task?params

请求参数：除了通用参数外，还包括如下参数

参数名	类型	必需	默认值	参数描述
push_id	string	是		要取消的任务号。

响应结果：在通用返回结果参数中，result 字段的 json 格式如下

```
{
    "status": 0, //0 为成功，其余为失败
}
```

## 2.10 批量设置标签

url 路径：

http://接口域名/v2/tags/batch\_set

请求参数：除了通用参数外，还包括如下参数

表 10 批量设置标签请求参数列表

参数名	类型	必需	默认值	参数描述
tag_token_list	string	是	无	json 字符串，包含若干标签-token 对，后台将把每一对里面的 token 打上对应的标签。每次调用最多允许设置 20 对，每个对里面标签在前，token 在后。注意标签最长 50 字节，不可包含空格；真实 token 长度至少 40 字节。

				示例如下（其中 token 值仅为示意）： [[{"tag1","token1"}, {"tag2","token2"}]]
--	--	--	--	---

**响应结果：**在通用返回结果参数中，result 字段的 json 为空

## 2.11 批量删除标签

**url 路径：**

http://接口域名/v2/tags/batch\_del

**请求参数：**除了通用参数外，还包括如下参数

表 11 批量删除标签请求参数列表

参数名	类型	必需	默认值	参数描述
tag_token_list	string	是	无	json 字符串，包含若干标签-token 对，后台将为每一对里面的 token 删除对应的标签。每次调用最多允许设置 20 对，每个对里面标签在前，token 在后。注意标签最长 50 字节，不可包含空格；真实 token 长度至少 40 字节。  示例如下（其中 token 值仅为示意）： [[{"tag1","token1"}, {"tag2","token2"}]]

**响应结果：**在通用返回结果参数中，result 字段的 json 为空

## 2.12 查询应用某 token 设置的标签

**url 路径：**

http://接口域名/v2/tags/query\_token\_tags?params

**请求参数：**除了通用参数外，还包括如下参数

表 12 查询 token 所设置的 tags

参数名	类型	必需	默认值	参数描述
device_token	string	是	无	

**响应结果：**在通用返回结果参数中，result 字段的 json 格式如下

```
{
  "tags":["tag1","tag2"]
}
```

## 2.13 查询应用某标签关联的设备数量

**url 路径：**

http://接口域名/v2/tags/ query\_tag\_token\_num?params

**请求参数：**除了通用参数外，还包括如下参数

表 13 查询应用 tag 的 deviceToken 数量

参数名	类型	必需	默认值	参数描述
tag	string	是	无	

**响应结果：**在通用返回结果参数中，result 字段的 json 格式如下

```
{
  "device_num":1
}
```

## 2.14 删除群发推送任务的离线消息

**url 路径：**

http://接口域名/v2/push/delete\_offline\_msg?params

**请求参数：**除了通用参数外，还包括如下参数

**请求参数：**除了通用参数外，还包括如下参数

参数名	类型	必需	默认值	参数描述
push_id	string	是		待删除离线消息的任务号。

**响应结果：**在通用返回结果参数中，result 字段的 json 为空

## 2.15 创建批量推送消息（包括通知和透传消息）

**url 路径：**

http://接口域名/v2/push/create\_multipush?params

**请求参数：**除了通用参数外，还包括如下参数

表 15 创建批量推送消息

参数名	类型	必需	默认值	参数描述
message_type	uint	是	无	消息类型：1：通知 2：透传消息
message	string	是	NA	参见“推送消息定义”一节
expire_time	uint	否	0	消息离线存储多久，单位为秒，最长存储时间 3 天。 在超时时间内，可以发起此消息的批量推送。
multi_pkg	uint	否	0	0 表示按注册时提供的包名分发消息；1 表示按 access id 分发消息，所有以该 access id 成功注册推送的 app 均可收到消息。
environment	uint	仅 iOS 必需	无	向 iOS 设备推送时必填，1 表示推送生产环境；2 表示推送开发环境。推送 Android 平台不填或填 0

**响应结果：**在通用返回结果参数中，result 字段的 json 示例如下

```
{
```

```
    "push_id":string (表示给 app 下发的任务 id)
}
```

## 2.16 按照已创建的批量推送消息推送给多个账户

设备的账户或别名由终端 SDK 在调用推送注册接口时设置，详情参考终端 SDK 文档。

**url 路径：**

```
http://接口域名/v2/push/account_list_multiple?params
```

**请求参数：**除了通用参数外，还包括如下参数

表 16 按照已创建的批量推送消息推送给多个账户

参数名	类型	必需	默认值	参数描述
account_list	string	是	无	由 account 组成的 json 数组，单次发送 account 不超过 1000 个。
push_id	uint	是	无	<a href="#">创建批量推送消息</a> 接口的返回值中的 push_id

**响应结果：**在通用返回结果参数中，result 字段的 json 为空。

## 2.17 按照已创建的批量推送消息推送给多个设备

设备的账户或别名由终端 SDK 在调用推送注册接口时设置，详情参考终端 SDK 文档。

**url 路径：**

```
http://接口域名/v2/push/device_list_multiple?params
```

**请求参数：**除了通用参数外，还包括如下参数

表 17 按照已创建的批量推送消息推送给多个设备

参数名	类型	必需	默认值	参数描述
device_list	string	是	无	由 token 组成的 json 数组，单次发送 token 不超过

				1000 个。
push_id	uint	是	无	<a href="#">创建批量推送消息</a> 接口的返回值中的 push_id

**响应结果：**在通用返回结果参数中，result 字段的 json 为空。

## 2.18 查询应用某个 token 的信息

**url 路径：**

http://接口域名/v2/application/get\_app\_token\_info?params

**请求参数：**除了通用参数外，还包括如下参数

表 18 查询应用 token 信息

参数名	类型	必需	默认值	参数描述
device_token	string	是	无	

**响应结果：**在通用返回结果参数中，result 字段的 json 格式如下

```
{
  "isReg":1, ( 1 为 token 已注册，0 为未注册 )
  "connTimestamp":1426493097, ( 最新活跃时间戳 )
  "msgsNum":2 ( 该应用的离线消息数 )
}
```

## 2.19 查询应用某账号映射的 tokens

**url 路径：**

http://接口域名/v2/application/get\_app\_account\_tokens?params

**请求参数：**除了通用参数外，还包括如下参数

表 19 查询 token 所设置的 tags

参数名	类型	必需	默认值	参数描述
account	string	是	无	账号

**响应结果：**在通用返回结果参数中，result 字段的 json 格式如下

```
{
  "tokens":["token1","token2"]
}
```