



# 推送平台接口文档



Huawei Technologies Co., Ltd.

华为技术有限公司

All rights reserved

版权所有 侵权必究



## Table of Contents 目 录

1	PUSH开放平台HCTT概述.....	4
1.1	系统概述 .....	4
1.2	典型应用交互流程.....	5
1.3	关键注意事项 .....	5
2	应用接入流程.....	5
2.1	应用接入准备 .....	5
2.2	应用接入关键参数.....	6
3	客户端开发(sdk模式).....	6
4	服务端开发接口 .....	6
4.1	接口定义 .....	6
4.2	对象类型 .....	7
4.2.1	android结构体.....	7
4.2.2	IOS结构体 .....	10
4.2.2.1	aps结构体 .....	12
4.2.2.2	Alert JSON OBJECT .....	13
4.3	推送请求接口 .....	14
4.3.1	输入参数说明.....	15
4.3.2	响应参数说明 .....	16
4.4	群发消息接口 .....	17
4.4.1	输入参数说明.....	17
4.4.2	响应参数说明.....	18
4.5	地理围栏消息接口.....	19
4.5.1	输入参数说明.....	19
4.5.2	响应参数描述.....	21
4.6	通知栏消息接口 .....	22
4.6.1	输入参数说明.....	22
4.6.2	响应参数描述.....	24
4.7	设置用户标签接口.....	25
4.7.1	输入参数说明.....	26
4.7.2	响应参数说明.....	26
4.8	查询应用标签接口.....	27

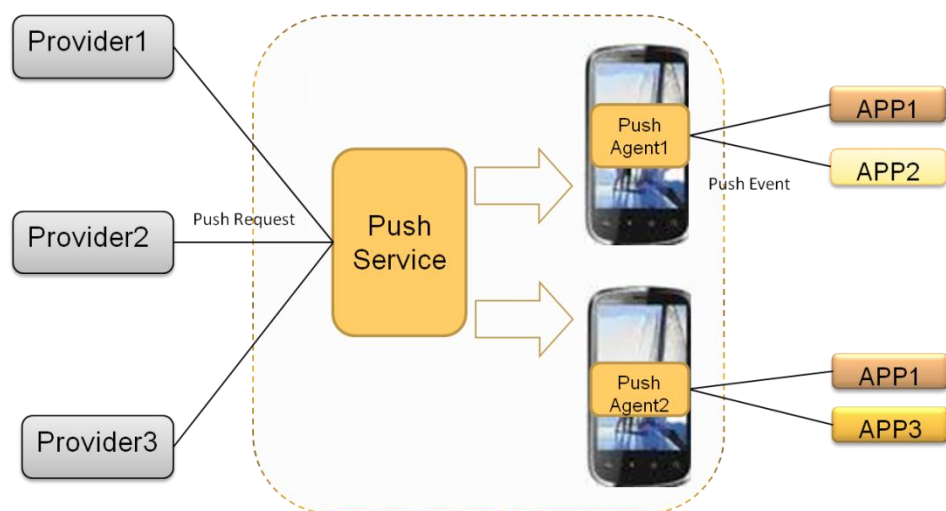


4.8.1	输入参数说明.....	27
4.8.2	响应参数说明.....	27
4.9	删除用户标签接口.....	28
4.9.1	输入参数说明.....	28
4.9.2	响应参数说明.....	28
4.10	查询用户标签接口.....	29
4.10.1	输入参数说明.....	29
4.10.2	响应参数说明.....	29
4.11	查询消息发送结果接口 .....	30
4.11.1	输入参数说明.....	30
4.11.2	响应参数说明.....	31
4.12	根据日期获取新生成PUSH用户标识接口 .....	32
4.12.1	输入参数说明.....	32
4.12.2	响应参数说明.....	32
4.13	返回代码表.....	33
4.13.1	HTTP返回码.....	33
4.13.2	错误代码对照表.....	34
5	常见问题.....	36

# 1 PUSH 开放平台 HCTT 概述

## 1.1 系统概述

华为PUSH开放平台HCTT（Huawei Cloud Terminal Touch）为合作伙伴提供一条从云端（应用侧）向移动终端无线推送消息的通道，通过此通道，应用可为用户带来更为即时、丰富的业务体验。



注：

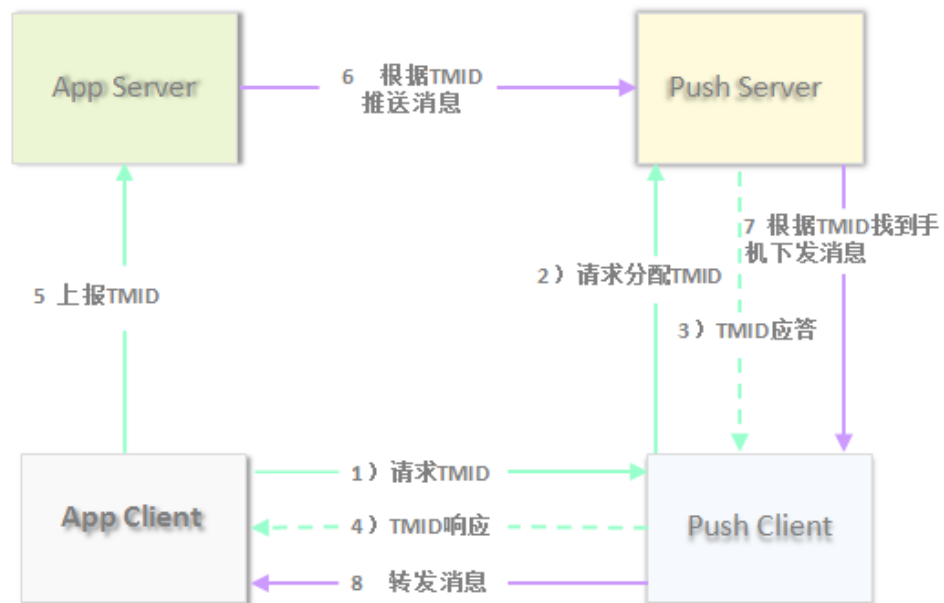
Provider：指PUSH消息的提供方，一般为应用的服务器端App server。

Push Service：指PUSH服务器

Push Agent：指PUSH服务在手机端的代理，即APP嵌入的SDK包

App：即应用的客户端软件。

## 1.2 典型应用交互流程



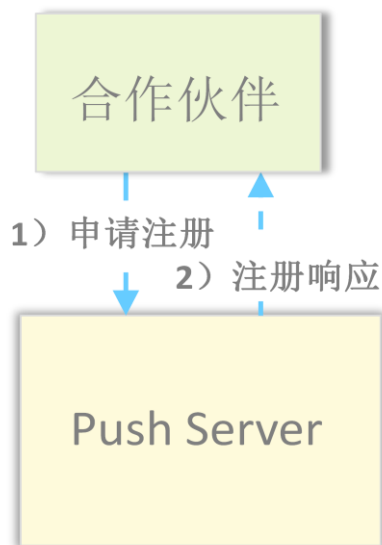
## 1.3 关键注意事项

华为PUSH通道目前基于IP连接,通过无线互联网网络传送,由于网络的不完全可靠性,在用户不在线时,不保证消息的实时送达。

当用户因为各种原因不在线时, PUSH提供消息缓存机制,在48小时内,当用户在此联网时即时推送到用户终端。超过48小时,消息可能被删除。

# 2 应用接入流程

## 2.1 应用接入准备



应用接入PUSH平台前，应先通过开发者社区注册，然后创建应用，应用审核通过后将获取应用ID：APP ID，应用安全码：APP Secre和APP TYPE。

应用提供者基于获取的应用ID和接口文档开发完成后可和PUSH开放平台对接测试，验证完成后通过上线审核即可上线。

## 2.2 应用接入关键参数

应用在注册和接入过程中，将根据应用场景获取以下关键参数：

- 1) APP ID：应用的标识，全局唯一。用以区分是接入方。
- 2) APP TYPE：应用类型。用以区分可调用的能力及约束。

应用在发送消息时，需要传送接受消息的用户标识：

- 1) TMID：用户标识唯一标识一部用户终端。此标识在应用通过开放接口注册到PUSH平台时会获得，需要应用客户端将此标识传递给应用服务端。

## 3 客户端开发(sdk 模式)

客户端开发请参照《华为PushSDK集成说明.pdf》

## 4 服务端开发接口

### 4.1 接口定义

本接口基于HTTPS协议，采用HTTPS POST方式，每个调用参数分为两个部分组成：系统级别参数和应用级别参数。



所有的请求和响应数据编码皆为utf-8格式。

## 4.2 对象类型

### 4.2.1 android 结构体

参数描述

参数名称	类型	是否必须	描述
notification_title	String	M	Notification bar上显示的标题
notification_content	String	M	Notification bar上显示的内容
notification_status_icon	String	O	系统小图标名称 该图标预置在客户端，在通知栏顶部展示
content_file_url	String	O	仅富媒体消息需要填写该字段  富媒体消息的附件为.zip格式，客户端解压后默认打开index.html。.zip文件的大小不能超过100K哦，否则会拒绝发送的  只要在开发者联盟上实名认证，都会赠送5G的云存储空间，可以在开发者联盟portal给每个应用申请云存储空间。富媒体附件一定要保存到云存储空间的哦。云存储文件地址格式： <a href="http://cs.dbank.com/dl/appName/path/filename">http://cs.dbank.com/dl/appName/path/filename</a> ，其中 “ <a href="http://cs.dbank.com/dl/">http://cs.dbank.com/dl/</a> ”为写死的部分， “appName/path/file”应用可以根据自身实际情况填写。如何上传文件请参考：资料中心->云存储服务->SDK下载 中的样例代码

doings	Integer	M	<p>1: 直接打开应用</p> <p>2: 通过自定义动作打开应用</p> <p>3: 打开URL</p> <p>4: 富媒体消息</p> <p>5: 短信收件箱广告</p> <p>6: 彩信收件箱广告</p> <p>注意: 当手机收到短信、彩信收件箱广告后, 在收件人一栏显示的是应用在联盟上注册的名字哦~</p>
smsContent	String	O	<p>短信收件箱广告内容</p> <p>当doings的取值为5时, 该字段必须填写</p>
mmsUrl	String	O	<p>彩信收件箱广告附件链接</p> <p>当doings的取值为6时, 该字段必须填写</p> <p>彩信附件为.zip格式, 压缩包中必须要有.smil描述文件, 并且符合标准smil语法哦。 .zip文件的大小不能超过100K哦, 否则会拒绝发送的</p> <p>只要在开发者联盟上实名认证, 都会赠送5G的云存储空间, 可以在开发者联盟portal给每个应用申请云存储空间。富媒体附件一定要保存到云存储空间的哦。云存储文件地址格式:</p> <p><a href="http://cs.dbank.com/dl/appName/path/filename">http://cs.dbank.com/dl/appName/path/filename</a>, 其中</p> <p>“<a href="http://cs.dbank.com/dl/">http://cs.dbank.com/dl/</a>” 为写死的部分,</p> <p>“appName/path/file” 应用可以根据自身实际情况填写。如何上传文件请参考: 资料中心-&gt;云存储服务-&gt;SDK下载 中的样例代码</p>



url	String	O	链接  当doings的取值为3时，必须携带该字段
intent	String	O	自定义打开应用动作  当doings的取值为2时，必须携带该字段
extras	JSON array	O	用户自定义键值对  "extras":[{"season":"Spring"}, {"weather":"raining"}]

直接打开应用样例：

```
{"notification_title":"hello","notification_content ":"hello","doings":1}
```

通过自定义动作打开应用样例：

```
{"notification_title":"hello","notification_content ":"hello","doings":2,"intent":"com.huawei.pushagent"}
```

打开URL样例：

```
{" notification_title":"the good news!","notification_content ":"Price reduction!","doings":3,"url":"vmall.com" }
```

富媒体消息样例：

```
{"notification_title":"hello","notification_content ":"hello","doings":4,"content_file_url":"http://xxxx/xx.zip"}
```

短信收件广告样例：

```
{"notification_title":"hello","notification_content ":"hello","doings":5, "smsContent":"Free phone!"}
```

彩信收件箱广告样例：

```
 {"notification_title":"hello","notification_content ":"hello","doings":6,"mmsUrl":"http://xxxx/xx" }
```

## 4.2.2 IOS 结构体

参数描述

参数名称	类型	是否必须	描述
aps	JSON OBJECT	M	详见4.2.2.1章节
doings	JSON Integer	M	1: 直接打开应用 3: 打开 UR
url	JSON String	O	链接 当 doings 的取值为 3 时，必须携带该字段

样例 1: The following payload has an aps dictionary with a simple, recommended form for alert messages with the default alert buttons (Close and View). It uses a string as the value of alert rather than a dictionary. This payload also has a custom array property.

```
{
  "aps" : { "alert" : "Message received from Bob" },
  "acme2" : [ "bang", "whiz" ]
}
```

样例 2. The payload in the example uses an aps dictionary to request that the device display an alert message with an Close button on the left and a localized title for the "action" button on the right side of the alert. In this case, “PLAY” is used as a key into the Localizable.strings file for the currently selected language to get the localized equivalent of “Play”. The aps dictionary also requests that the application icon be badged with the number 5.

```
{
  "aps" : {
    "alert" : {
      "body" : "Bob wants to play poker",
      "action-loc-key" : "PLAY"
    },
    "badge" : 5,
  }
}
```

```
},  
"acme1" : "bar",  
"acme2" : [ "bang", "whiz" ]  
}
```

样例 3. The payload in this example specifies that device should display an alert message with both Close and View buttons. It also request that the application icon be badged with the number 9 and that a bundled alert sound be played when the notification is delivered.

```
{  
  "aps" : {  
    "alert" : "You got your emails.",  
    "badge" : 9,  
    "sound" : "bingbong.aiff"  
  },  
  "acme1" : "bar",  
  "acme2" : 42  
}
```

样例 4. The payload in this example uses the loc-key and loc-args child properties of the alert dictionary to fetch a formatted localized string from the application's bundle and substitute the variable string values (loc-args) in the appropriate places. It also specifies a custom sound and includes a custom property.

```
{  
  "aps" : {  
    "alert" : {  
      "loc-key" : "GAME_PLAY_REQUEST_FORMAT",  
      "loc-args" : [ "Jenna", "Frank"]  
    },  
    "sound" : "chime"  
  },  
  "acme" : "foo"  
}
```



## 4.2.2.1 aps结构体

## 参数描述

参数名称	类型	是否必须	描述
alert	JSON OBJECT OR JSON STRING	M	<p>If this property is included, the system displays a standard alert. You may specify a string as the value of alert or a dictionary as its value. If you specify a string, it becomes the message text of an alert with two buttons: Close and View. If the user taps View, the application is launched.</p> <p>Alternatively, you can specify a dictionary as the value of alert. See chapter 2.4.1 for descriptions of the keys of this dictionary</p>
badge	JSON INT	O	<p>The number to display as the badge of the application icon.</p> <p>If this property is absent, the badge is not changed. To remove the badge, set the value of this property to 0</p>
sound	JSON STRING	O	<p>The name of a sound file in the application bundle. The sound in this file is played as an alert. If the sound file</p>

			doesn't exist  or default is specified  as the value, the  default alert sound is  played. The audio  must be in one of the  audio data formats  that are compatible  with system sounds
--	--	--	--

#### 4.2.2.2 Alert JSON OBJECT

##### 参数描述

参数名称	类型	是否必须	描述	备注
body	JSON STRING	O	The text of the alert message	
action-loc-key	JSON STRING	O	If a string is specified, the system displays an alert with two buttons, whose behavior is described in chapter 2.4.  The string is used as a key to get a localized string in the current localization to use for the right button's title instead of "View". If the value is null, the system displays an alert with a single OK button that simply dismisses the alert when tapped	

loc-key	JSON STRING	O	A key to an alert-message string in a Localizable.strings file for the current localization (which is set by the user's language preference). The key string can be formatted with %@ and %n\$@specifiers to take the variables specified in loc-args	成对出现
loc-args	JSON ARRAY	O	Variable string values to appear in place of the format specifiers in loc-key	
launch-image	JSON STRING	O	<p>The filename of an image file in the application bundle; it may include the extension or omit it. The image is used as the launch image when users tap the action button or move the action slider. If this property is not specified, the system either uses the previous snapshot, uses the image identified by the UILaunchImageFile key in the application's Info.plist file, or falls back to Default.png.</p> <p>This property was added in iOS 4.0</p>	

### 4.3 推送请求接口

#### 4.3.1 输入参数说明

方法名称: openpush.message.single\_send

方法描述: 单条发送Push消息。

请求消息参数描述

参数名称	类型	是否必须	描述
deviceToken	String	M	32 字节长度, 由系统分配的合法TMID
message	String	M	发送到设备上的消息, 最长为4096 字节
priority	Integer	M	0: 高优先级 1: 普通优先级 缺省值为1
cacheMode	Integer	M	消息是否需要缓存 0: 不缓存 1: 缓存 缺省值为0
msgType	Integer	M	标识消息类型(缓存机制), 由调用端赋值, 取值范围(1~100)。当TMID+msgType的值一样时, 仅缓存最新的一条消息
requestID	String	O	如果开发者填写了该

			<p>字段，则需要保证该字段唯一</p> <p>如果开发者没有携带该字段，则推送平台会生成一个全局唯一的ID</p>
expire_time	String	O	<p>unix 时间戳</p> <p>格式：2013-08-29 19:55</p> <p>消息过期删除时间</p> <p>如果不填写，默认超时时间为当前时间后48小时</p>

#### 4.3.2 响应参数说明

##### 响应消息参数描述

参数名称	类型	是否必须	描述
resultcode	String	必选	<p>结果码</p> <p>0: success</p> <p>100: Invalid unknown params.</p> <p>20203: No permission to send message to these tmIDs.</p> <p>20204: Invalid expire</p>





			time.  20205: Not enough quota.
message	String	可选	对结果码的描述
requestID	String	必选	消息跟踪 ID  如果请求消息中填写 了该字段，则将请求 消息中的值返回

#### 4.4 群发消息接口

方法名称: openpush.message.batch\_send

方法描述: 群发Push消息

##### 4.4.1 输入参数说明

请求消息参数描述

参数名称	类型	是否必须	描述
deviceTokenList	String[]	M	Device token列表，最 多填1000个
message	String	M	发送到设备上的消 息，最长为4096 字节
cacheMode	Integer	M	消息是否需要缓存  0: 不缓存  1: 缓存

			缺省值为0
msgType	Integer	M	标识消息类型（缓存机制），由调用端赋值，取值范围（1~100）。当TMID+msgType的值一样时，仅缓存最新的一条消息
expire_time	String	O	unix 时间戳 格式：2013-08-29 19:55 消息过期删除时间 如果不填写，默认超时时间为当前时间后48小时

#### 4.4.2 响应参数说明

##### 响应消息参数描述

参数名称	类型	是否必须	描述
resultcode	String	必选	结果码  0: success  100: Invalid unknown params.  20203: No permission to send message to

			these tmIDs.  20204: Invalid expire time.  20205: Not enough quota.
message	String	可选	对结果码的描述
requestID	String	必选	消息跟踪 ID  如果请求消息中填写了该字段，则将请求消息中的值返回

## 4.5 地理围栏消息接口

方法名称: openpush.openapi.lbs\_send

方法描述: 该接口用户创建地理围栏消息，如果有用户进入地理围栏，则给用户发送通知栏消息

### 4.5.1 输入参数说明

请求消息参数描述

参数名称	类型	是否必须	描述
location	String	M	经纬度信息  样例:  { "location": { "type": "1", "coordinates": [ { "lat": "32.57844", "lng": "117.502087" }, { "lat": "30.848528", "lng": "118.458746" }, { "lat": "31.33934", "lng": "120.31" } ] } }

			<p>6873"},{"lat":"32.656293","lng":"119.580981"}]}}</p> <p>Latlng表示经纬度列表，多个经纬度围城一个封闭区域，必须按照顺序填写，否则会影响定位准确度；</p> <p>纬度在前，经度在后，纬度、经度之间用英文“,”分隔。</p> <p>Type表示封闭区域类型，目前仅支持1，表示多边形</p>
tokens	String	O	<p>用户标识</p> <p>样例： xxx, ddd</p> <p>多个token用 “,” 分隔</p> <p>注意：该字段暂不生效</p>
tags	String	O	<p>用户标签</p> <p>样例：</p> <pre>{"tags":[{"location":["ShangHai","GuangZhou"]}, {"age":["20","30"]}]}</pre> <p>含义：在广州或者上海，并且年龄为20或30岁的用户</p>
exclude_tags	String	O	<p>需要剔除的用户的标签</p> <p>样例：</p> <pre>{"exclude_tags":[{"music":["blue"]}, {"fruit":["apple"]}]}</pre> <p>含义：不给喜欢蓝调音乐的用户，或者喜欢苹果的用户发送消息</p>
android	android结	M	消息内容

	构体		样例: {"notification_title":"the good news!","notification_content":"Price reduction!","doings":3,"url":"vmall.com"}
send_time	String	O	消息生效时间。如果不携带该字段,则表示消息实时生效。实际使用时,该字段精确到分  消息发送时间戳,timestamp格式ISO 8601: 2013-06-03T17:30:08+08:00
expire_time	String	M	消息过期删除时间  消息过期时间,timestamp格式ISO 8601: 2013-06-03T17:30:08+08:00
allow_periods	String	O	消息允许展示时间段,时间精确到半小时,24小时制,可以填写一个或者多个时间段  消息到达客户端后,由客户端决定是否将消息弹出或展示  时间段样例: [[09:30,12:00],[15:00,16:00]],表示上午9点30到12点之间和下午3点到4点之间可以展示消息  各时间段之间不允许重叠

#### 4.5.2 响应参数描述

##### 响应消息参数描述

参数名称	类型	是否必须	描述
result_code	String	M	返回码  0: 成功
request_id	String	M	由服务器生成,方便用户问题追查与定位
result_desc	String	O	失败原因



			返回码备注，失败原因，成功时无需填写  80100001: Failed to get userId or devAppId from context  80200001: Illegal parameter  80200005: Attachment is too big  80300002: The app is not exist  80500001: Failure
--	--	--	--

响应消息样例：

```
{"result_code": "0", "request_id": "123456"}
```

## 4.6 通知栏消息接口

方法名称：openpush.openapi.notification\_send

方法描述：该方法用于发送通知栏消息

### 4.6.1 输入参数说明

请求消息参数描述

参数名称	类型	是否必须	描述
push_type	Integer	M	推送范围  1：指定用户，必须指定tokens字段

			<p>2: 所有人, 无需指定tokens, tags, exclude_tags</p> <p>3: 一群人, 必须指定tags或者exclude_tags字段</p>
tokens	String	O	<p>用户标识</p> <p>样例: xxx, ddd</p> <p>多个token用 “,” 分隔</p>
tags	String	O	<p>用户标签</p> <p>样例:</p> <pre>{ "tags": [{ "location": ["ShangHai", "GuangZhou"] }, { "age": ["20", "30"] } ] }</pre> <p>含义: 在广州或者上海, 并且年龄为20活30岁的用户</p>
exclude_tags	String	O	<p>需要剔除的用户的标签</p> <p>样例:</p> <pre>{ "exclude_tags": [{ "music": ["blue"] }, { "fruit": ["apple"] } ] }</pre> <p>含义: 不给喜欢蓝调音乐的用户, 或者喜欢苹果的用户发送消息</p>
android	android 结构体	M	<p>消息内容</p> <p>样例: { " notification_title": "the good news!", "notification_content": "Price reduction!", "doings": 3, "url": "vmall.com" }</p>
send_time	String	O	<p>消息生效时间。如果不携带该字段, 则表示消息实时生效。实际使用时, 该字段精确到分</p> <p>消息发送时间戳, timestamp格式ISO 8601:</p> <p>2013-06-03T17:30:08+08:00</p>

expire_time	String	O	消息过期删除时间。如果不带该字段，消息默认删除时间是48小时后  消息发送时间戳，timestamp格式ISO 8601： 2013-06-03T17:30:08+08:00
device_type	Integer	O	目标设备类型 1: android  2: ios  默认为 android
message	String	O	消息结构体 发送给非 android 设备的消息内容  样例：  <pre>{"ios":{"aps":{"alert":"Message received from Bob"},"acme2":["bang","whiz"]}}</pre> ios 结构体参照 2.3 章节  现在除 android 设备外，只有 IOS 设备，后续如果还要支持 WP 设备的话，只需要在 message 结构体中，再增加 wp 结构体。这样接口字段就不用再调整了
target_user_type	Integer	O	1: IOS 开发用户  2: IOS 生产用户  目前仅给 IOS 设备发送消息需要填写该字段
allow_periods	String	O	消息允许展示时间段，时间精确到半小时，24 小时制，可以填写一个或者多个时间段  消息到达客户端后，由客户端决定是否将消息弹出或展示  时间段样例：[[09:30,12:00],[15:00,16:00]]，表示上午 9 点 30 到 12 点之间和下午 3 点到 4 点之间可以展示消息  各时间段之间不允许重叠

#### 4.6.2 响应参数描述

##### 响应消息参数描述



参数名称	类型	是否必须	描述
result_code	String	M	返回码  0: 成功
request_id	String	M	由服务器生成，方便 用户问题追查与定位
result_desc	String	O	失败原因  返回码备注，失败原因，成功时无需填写  80100001: Failed to get userId or devAppId from context  80200001: Illegal parameter  80200005: Attachment is too big  80300002: The app is not exist  80500001: Failure

响应消息样例：

```
{"result_code":"0","request_id":"123456"}
```

## 4.7 设置用户标签接口

方法名称：openpush.openapi.set\_user\_tag

#### 4.7.1 输入参数说明

请求消息参数描述

参数名称	类型	是否必须	描述
token	String	M	用户标识
tag_key	String	M	标签类型 比如：地区
tag_value	String	M	标签值 比如：上海

#### 4.7.2 响应参数说明

响应消息参数描述

参数名称	类型	是否必须	描述
result_code	String	M	0：成功
request_id	String	M	由服务器生成，方便 用户问题追查与定位
result_desc	String	O	失败原因  80100001: Failed to get userId or devAppId from context  80200001: Illegal parameter  80200003: Illegal tokens



			80300002: The app is not exist  80500001: Failure
--	--	--	---

响应消息样例:

响应消息为JSON格式

```
{"result_code":"0","request_id":"1370749118"}
```

## 4.8 查询应用标签接口

方法名称: openpush.openapi.query\_app\_tags

### 4.8.1 输入参数说明

该方法不需要输入参数

参数名称	类型	是否必须	描述
------	----	------	----

### 4.8.2 响应参数说明

响应消息参数描述

参数名称	类型	是否必须	描述
request_id	String	M	由服务器生成,方便用户问题追查与定位
tags	String	O	标签列表  样例: <pre>{"tags":[{"location":["ShangHai","GuangZhou"]}, {"age":["20","30"]}]}</pre>

响应消息样例:



响应消息为JSON格式

```
{"request_id":"1370749118","tags":[{"location":["ShangHai","GuangZhou"]}, {"age":["20","30"]}]}
```

## 4.9 删除用户标签接口

方法名称: openpush.openapi.delete\_user\_tag

### 4.9.1 输入参数说明

请求消息参数描述

参数名称	类型	是否必须	描述
token	String	M	用户标识
tag_key	String	M	标签类型 比如: 地区

### 4.9.2 响应参数说明

响应消息参数描述

参数名称	类型	是否必须	描述
result_code	String	M	0: 成功
request_id	String	M	由服务器生成, 方便用户问题追查与定位
result_desc	String	O	失败原因  80100001: Failed to get userId or devAppId from context



			80200001: Illegal parameter  80200003: Illegal tokens  80300002: The app is not exist  80500001: Failure
--	--	--	--

响应消息样例:

响应消息为JSON格式

```
{"result_code": "0", "request_id": "1370749118"}
```

#### 4.10 查询用户标签接口

方法名称: openpush.openapi.query\_user\_tag

##### 4.10.1 输入参数说明

请求消息参数描述

参数名称	类型	是否必须	描述
token	String	M	用户标识

##### 4.10.2 响应参数说明

响应消息参数描述

参数名称	类型	是否必须	描述
request_id	String	M	由服务器生成,方便用户问题追查与定位
tags	String	O	标签列表



			样例:  {"tags":[{"location":"ShangHai"}, {"age":"20"}]}
--	--	--	---

响应消息样例:

响应消息为JSON格式

```
{"request_id":"1370749118","tags":[{"location":"ShangHai"}, {"age":"20"}]}
```

#### 4.11 查询消息发送结果接口

该接口仅能查询single\_send和batch\_send接口发送的消息

方法名称: openpush.openapi.query\_msg\_result

##### 4.11.1 输入参数说明

请求参数描述:

参数名称	类型	是否必须	描述
request_id	String	M	开发者调用 single_send和 batch_send接口时返 回的requestID字段值
token	String	O	用户标识  如果携带该字段, 则 表示查询request_id 中的token对应的消 息结果; 如果不携带 该字段, 则查询 request_id对应的所 有token的消息结果



#### 4.11.2 响应参数说明

响应消息参数描述

参数名称	类型	是否必须	描述
request_id	String	M	消息跟踪ID  请求消息中的 request_id
result	JSON ARRAY	M	消息状态

result结构体:

参数名称	类型	是否必须	描述
token	String	M	用户标识
status	Int	M	消息状态  0: 成功送达  1: 待发送（没送到， 但又没过期、没被覆盖 的消息，还在等待 补发的）  2: 被覆盖  3: 过期丢弃

响应消息样例:

```
{"result":[{"status":0,"token":"00000000000000000000000000000000"},  
{"request_id":"123456"}]}
```



## 4.12 根据日期获取新生成 PUSH 用户标识接口

该接口用于开发者根据时间获取，指定时间段内新生成的PUSH用户标识

PUSH用户标识根据时间，每天生成一个文件

### 4.12.1 输入参数说明

请求参数描述：

参数名称	类型	是否必须	描述
date	String	M	格式：2014-04-25 表示获取 2014-04-25 这一天新产生的 token  不能填写当天时间，可 以填写当前之前的时间

### 4.12.2 响应参数说明

响应消息参数描述：

参数名称	类型	是否必须	描述
result_code	String	M	响应码  0: success  80500001: Failure  80100001: Failed to get userId or devAppld from context  80200001: Illegal parameter  80200017: There is no token file  80300002:The app is not exist
result_desc	String	O	响应码描述  0: success





			80500001: Failure  80100001: Failed to get userId or devAppld from context  80200001: Illegal parameter  80200017: There is no token file  80300002:The app is not exist  当没有获取到文件下载链接时必须填写该字段，用于高速开发者为什么没有获取到文件
request_id	String	M	每次请求唯一标识
tokenFile_url	String	O	文件存在的时候，则该字段填写文件下载链接，每个链接的有效时间为 12 小时，12 小时过后则链接失效  文件为.zip 格式
unzip_password	String	O	文件解压密码  获取文件之后用这个密码解压缩

响应消息样例：

```
{"request_id":"1399551889637254","result_code":"0","tokenFile_url":"http://xxx","unzip_password":"xxx"}
```

## 4.13 返回代码表

### 4.13.1 HTTP 返回码

代码	错误信息	详细描述
200	成功	
500	服务器内部错误	



503	服务端暂时不可达	
-----	----------	--

#### 4.13.2 错误代码对照表

➤ 系统级错误代码

错误代码	错误信息	详细描述
0	成功	
1	一个未知的错误发生	
2	服务临时不可用	
3	未知的方法	
4	应用已达到设定的请求上限	
5	请求来自未经授权的IP地址	
7	调用次数超过了限制	
8	调用太频繁	
100	无效未知参数	
101	无效的API_KEY	
102	无效的SESSION_KEY	
103	必须是POST提交	
104	无效的签名	
105	缺少系统参数： app_id,app_key等	
106	app没有调用当前服务的权限	
107	app_id和secret需要重新获取 (如算法升级等)	
108	路由失败，无法识别服务投递地	



	址	
--	---	--

## ➤ 业务级错误代码

错误代码	错误信息	详细描述
20001	tmID is null	tmID参数为空
20002	tmID is too many	tmID参数太长了
20003	Message is null	消息内容为空
20004	Data too long, please send data less than 1024 bytes	消息内容不得超过1024字节
20005	Illegal cache Mode	非法的缓存模式
20006	Illegal msg Type	非法的消息类型
20007	Device Quota Exceeded	发送给某个tmID终端的Push消息太多
20101	message Unreachable	消息不可达
20102	Send message Timeout	发送消息超时
20202	Illegal file Url	非法的文件url地址
20203	No permission to send message to these tmIDs	无权给这些tmID发送消息。
20204	Invalid expire time	无效的过期时间
80100001	Failed to get userId or devAppId from context	userId或者devAppId未能获取到
80200001	Illegal parameter	参数非法



80200003	request_id is not exist	requestId不存在
80200004	Illegal token	Token非法
80200005	Attachment is too big	附件过大
80200006	Analyse message into json failed	消息无法解析为json结构体
80200007	Illegal priority	消息非同步非异步
80200008	DeviceTokenList contains more than 1000 items	DeviceTokenList包含超过1000个token
80300002	The app is not exist	应用不存在
80300009	Not enough quota	配额不足
80500001	Failure	操作失败

## 5 常见问题