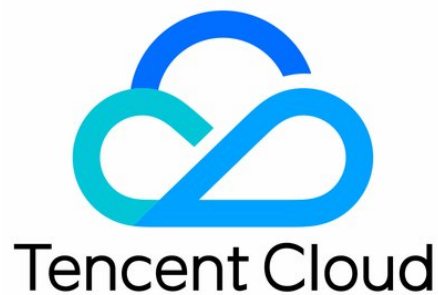


Tencent Push Notification Service

API Documentation

Product Documentation



Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

API Documentation

- Overview

- API Overview

- Calling Method

 - Signature Verification (recommended)

 - Basic Auth Verification

 - Request Structure

- Push APIs

 - Push API

 - Creating Push Plan

 - Token Package Upload API

 - Account Package Upload API

- Tag APIs

 - Tag Binding and Unbinding

 - Deleting All Devices Under Tag

- Account APIs

 - Account Binding and Unbinding

 - Querying Account and Device Binding

- Statistics APIs

 - Querying Daily Aggregated Push Statistics

 - Querying Device Statistics

 - Querying All Push Information in Specific Period

 - Querying Push Information for One Task

 - Querying Push Statistics Information for One Task

 - Querying Aggregated Statistics for Multiple Tasks

- Server-Side Error Codes

- API (java)

- User Attribute APIs

API Documentation

Overview

Last updated : 2020-07-01 16:40:40

RESTful APIs are HTTP management APIs that provide the application backend with a management entry at the backend for you to remotely call the push service.

RESTful APIs support both HTTP and HTTPS protocols. For the sake of security, the HTTPS protocol is recommended.

Prerequisites

Before you call a RESTful API, make sure that you have created the application in the TPNS Console. For more information, please see [Creating Products and Applications](#).

API Overview

Last updated : 2020-10-19 10:18:25

Push APIs

API name	Feature
v3/push/app	Pushes notification and in-app messages
v3/push/package/upload	Uploads number package

Tag APIs

API name	Feature
v3/device/tag	Binds and unbinds tag
v3/device/tag/delete_all_device	Deletes all devices under tag

Account APIs

API name	Feature
v3/device/account/batchoperate	Binds and unbinds account
v3/device/account/query	Queries the account device binding

Statistics APIs

API name	Feature
v3/statistics/get_push_stat_overview	Queries daily aggregated push statistics
v3/statistics/get_device_stat_overview	Queries device statistics
v3/statistics/get_push_record	Queries all push information in a specific period
v3/statistics/get_push_record	Queries information of a single task
v3/statistics/get_push_task_stat_channel	Queries statistics information for a single task
v3/statistics/get_push_group_stat	Aggregates queries on statistics data for multiple tasks

Calling Method

Signature Verification (recommended)

Last updated : 2020-12-31 17:43:48

Overview

This document describes the signature authentication methods of TPNS.

The HMAC-SHA256 algorithm is used to produce signing information according to `SecretKey`. The authentication is performed by verifying the signature, which ensures higher security and is recommended.

Parameter description

Parameter	Description
AccessId	Application ID assigned by the TPNS backend, which can be obtained in Configuration Management > Basic Configuration in the **TPNS console
SecretKey	<code>SecretKey</code> assigned by the TPNS backend, which corresponds to <code>AccessId</code> and can be obtained in Configuration Management > Basic Configuration in the **TPNS console
Sign	API signature method
timeStamp	Request timestamp

Signature Generation Method

1. Splice the request timestamp + `accessId` + request body to get the original string to be signed:

```
String to be signed = ${timeStamp} + ${accessId} + ${request body}
```

2. Use `secretKey` as the key to sign the original string to be signed to generate a signature:

```
sign = Base64(HMAC_SHA256(string to be signed, secretKey))
```

HTTP Protocol Assembly Method

In addition to the general header protocol, the HTTP protocol header also needs to carry the current request timestamp, `AccessId`, and the signature's `Sign` information. The specific parameters are as follows:

Parameter Key in Header	Description	Required
Sign	Request signature	Yes
AccessId	Application ID	Yes
TimeStamp	Request timestamp	Yes

The specific HTTP request packet is as follows:

```
POST /v3/push/app HTTP/1.1
Host: api.tps.tencent.com
Content-Type: application/json
AccessId: 1500001048
TimeStamp: 1565314789
Sign: Y2QyMDc3NDY4MmJmNzhiZmRiNDNlMTdkMWQ1ZDU2YjNlNWl3ODlhMTY3MGZjMTUyN2VmNTRjNjVzMmQ3Yjc2ZA==
{"audience_type": "account", "platform": "android", "message": {"title": "test title", "content": "test content", "android": {"action": {"action_type": 3, "intent": "xgscheme://com.xg.push/notify_detail?param1=xg"}}}, "message_type": "notify", "account_list": ["5822f0eee44c3625ef0000bb"] }
```

Signature Generation Sample

1. The generated string to be signed is as follows:

```
String to be encrypted =15653147891500001048{"audience_type": "account", "platform": "android", "message": {"title": "test title", "content": "test content", "android": {"action": {"action_type": 3, "intent": "xgscheme://com.xg.push/notify_detail?param1=xg"}}}, "message_type": "notify", "account_list": ["5822f0eee44c3625ef0000bb"] }
```

2. Generate a hexadecimal hash based on the key through the HMAC-SHA256 algorithm, i.e., `secretKey` =1452fcb9e9f3115ba794fb0fff2fd73 in the sample.

```
hashcode= hmac-sha256(string to be signed, secretKey)
Get hashcode="cd20774682bf78bdfb43e17d1d5d56b3e5b789a1670fc1527ef54c65d2d7b76d"
```

3. Base64-encode the hashcode to get the following signature string:

```
Get Sign=Base64(hashcode)
Sign="Y2QyMDc3NDY4MmJmNzhiZmRiNDNlMTdkMWQ1ZDU2YjNlNWl3ODlhMTY3MGZjMTUyN2VmNTRjNjVzMmQ3Yjc2ZA=="
```

Sample signature for Python

```
#!/usr/bin/env python
import hmac
import base64
from hashlib import sha256

s = '15653147891500001048{"audience_type": "account", "platform": "android", "message": {"title": "test title", "content": "test content", "android": {"action": {"action_type": 3, "intent": "xgscheme://com.xg.push/notify_detail?param1=xg"}}}, "message_type": "notify", "account_list": ["5822f0eee44c3625ef0000bb"] }'
key = '1452fcb9e9f3115ba794fb0fff2fd73'
hashcode = hmac.new(key, s, digestmod=sha256).hexdigest()
print base64.b64encode(hashcode)
```

Sample signature for Java

```

package com.tencent.xg;

import java.io.UnsupportedEncodingException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import org.apache.commons.codec.binary.Base64;
import org.apache.commons.codec.binary.Hex;

public class SignTest {
public static void main(String[] args) {
try{
String stringToSign = "15653147891500001048{"audience_type": "account", "platform": "android", "message": {"title": "test title", "content": "test content", "android": { "action": { "action_type": 3, "intent": "xgscheme://com.xg.push/notify_detail?param1=xg"}}, "message_type": "notify", "account_list": ["5822f0eee44c3625ef0000bb"] } }";
String appSecret = "1452fcbae9f3115ba794fb0fff2fd73";

Mac mac;
mac = Mac.getInstance("HmacSHA256");
mac.init(new SecretKeySpec(appSecret.getBytes("UTF-8"), "HmacSHA256"));
byte[] signatureBytes = mac.doFinal(stringToSign.getBytes("UTF-8"));

String hexStr = Hex.encodeHexString(signatureBytes);
String signature = Base64.encodeBase64String(hexStr.getBytes());

System.out.println(signature);
} catch (NoSuchAlgorithmException | InvalidKeyException | UnsupportedEncodingException e) {
e.printStackTrace();
}
}
}

```

Sample signature for Go

```

import(
    "crypto/hmac"
    "crypto/sha256"
    "encoding/base64"
    "encoding/hex"
    "testing"
)

func TestSign(t *testing.T) {
requestBody := "15653147891500001048{"audience_type": "account", "platform": "android", "message": {"title": "test title", "content": "test content", "android": { "action": { "action_type": 3, "intent": "xgscheme://com.xg.push/notify_detail?param1=xg"}}, "message_type": "notify", "account_list": ["5822f0eee44c3625ef0000bb"] } }"
secretKey := "1452fcbae9f3115ba794fb0fff2fd73"

h := hmac.New(sha256.New, []byte(secretKey))

```



```

h.Write([]byte(requestBody))
sha := hex.EncodeToString(h.Sum(nil))
sign := base64.StdEncoding.EncodeToString([]byte(sha))
println(sign)
}

```

Sample signature for C#

```

using System;
using System.Security.Cryptography;
using System.Text;

namespace tpns_server_sdk_cs
{
    class GenSign {

        // Main Method
        // static public void Main(String[] args)
        // {
        // string reqBody =
        // "{¥"audience_type¥": ¥"account¥",¥"platform¥": ¥"android¥",¥"message¥": {¥"title¥": ¥"test title¥",¥"content¥": ¥"test content¥",¥"android¥": { ¥"action¥": {¥"action_type¥": 3,¥"intent¥": ¥"xgscheme://com.xg.push/notify_detail?param1=xg¥"}},¥"message_type¥": ¥"notify¥",¥"account_list¥": [¥"5822f0eee44c3625ef0000bb¥" ] }";
        // string genSign = GenSign.genSign("1565314789", "1500001048", "reqBody", "1452fceb9f3115ba794fb0fff2fd73");
        // Console.WriteLine(genSign);
        // }

        public static string HmacSHA256(string key, string data)
        {
            string hash;
            ASCIIEncoding encoder = new ASCIIEncoding();
            Byte[] code = encoder.GetBytes(key);
            using (HMACSHA256 hmac = new HMACSHA256(code))
            {
                Byte[] hmBytes = hmac.ComputeHash(encoder.GetBytes(data));
                hash = ToHexString(hmBytes);
            }
            return hash;
        }

        public static string ToHexString(byte[] array)
        {
            StringBuilder hex = new StringBuilder(array.Length * 2);
            foreach (byte b in array)
            {
                hex.AppendFormat("{0:x2}", b);
            }
            return hex.ToString();
        }

        public static string genSign(string timeStampStr, string accessId, string requestBody, string keySecret)
        {
            string data = timeStampStr + accessId + requestBody;

```

```
string hash = HmacSHA256(keySecret, data);
string sign = Base64Encode(hash);
Console.WriteLine("timestampStr: " + timestampStr + " accessId:" + accessId + " requestBody" + requestBody + "
keySecret:" + keySecret);
return sign;
}

public static string Base64Encode(string plainText) {
var plainTextBytes = System.Text.Encoding.UTF8.GetBytes(plainText);
return System.Convert.ToBase64String(plainTextBytes);
}
}
}
```

Basic Auth Verification

Last updated : 2020-05-13 18:13:08

This document describes the Basic Auth authentication methods of TPNS.

For Basic Auth authentication with `AccessId` and `SecretKey`, the key is easy to be compromised and the security is not high. You are recommended to use [signature authentication](#).

Obtaining Key

1. Log in to the [TPNS Console](#) and select **Configuration Management** on the left sidebar.
2. Get `Access ID` and `SECRET KEY` from the Application Information column (only visible to root account).

Generating Signature

1. Generate a Request String

This step generates a request string.

The generation algorithm is: `base64(Access ID:SECRETKEY)`, which adds a colon after `Access ID` followed by `SECRETKEY` to form a string and then perform `base64` encoding on the string.

For example:

```
base64(150000****:cf43dac624820****c1fe5fc993)
```

2. Perform Basic Auth verification and authentication

Basic authentication is used, i.e., adding a field (`key-value` pair) to the HTTP Header:

```
Authorization: Basic base64_auth_string
```

For example:

```
Authorization:Basic base64(150000****:cf43dac624820****c1fe5fc993)
```

Request Structure

Last updated : 2020-11-03 11:34:47

Service URL

TPNS currently provides four service access points in Guangzhou, Shanghai, Hong Kong (China), and Singapore (Southeast Asia). The data between different service access points is completely isolated. Based on the service access point you selected when creating a product, a service URL is obtained.

Test	Guangzhou
Platform	Application Name
Android	Test-Android-long-name-test
iOS	Test-iOS

The table below lists the service URLs currently available:

Service Access Point	Service URL
Service access point in Guangzhou, China	https://api.tpns.tencent.com/
Service access point in Shanghai, China	https://api.tpns.sh.tencent.com/
Service access point in Hong Kong, China	https://api.tpns.hk.tencent.com/
Service access point in Singapore	https://api.tpns.sgp.tencent.com/

Communication Protocol

All TencentCloud APIs communicate over HTTPS, which provides high security.

Request Method

The supported HTTP request methods are as follows:

- POST

POST requests support the following content types:

1. application/json.
2. multipart/form-data (only supported by certain APIs).

Character Encoding

Only UTF-8 encoding is used.

Push APIs

Push API

Last updated : 2020-12-23 15:06:44

API Description

Request method: POST.

```
Service address/v3/push/app
```

The API service address corresponds to the service access point one by one; therefore, please select the service address corresponding to your application [service access point](#).

Feature: `Push` API is the general term for all push APIs. Push API has many types of push targets, which are described as follows.

All request parameters are JSON encapsulated and uploaded to the backend, which differentiates push targets based on the request parameters. If an error code returns, please see [Server Error Codes](#).

Required Parameters

The required push parameters refer to the parameters that must be carried in a push message.

Parameter Name	Type	Required	Description
audience_type	String	Yes	Push target. Valid values: <ul style="list-style-type: none">all: full pushtag: tag pushtoken: single-device pushtoken_list: device list pushaccount: single-account pushaccount_list: account list pushpackage_account_push: account package pushpackage_token_push: token package push
message	Object	Yes	Message body. For more information, please see message body type
message_type	String	Yes	Message type. Valid values: <ul style="list-style-type: none">notify: notificationmessage: in-app message/silent message

environment	String	Yes (only for iOS)	Push environment (only available for pushes on iOS). Valid values: <ul style="list-style-type: none"> product: production environment dev: development environment
upload_id	Integer	Yes (only available for account package push/token package push)	Account/token package upload ID

audience_type: push target

Push target indicates which devices a push can be delivered to.

Push API supports a variety of push targets, such as full push, tag push, single device push, device list push, single account push, and account list push.

Push Target	Description	Required Parameters and Instructions
all	Full push	Empty
tag	Tag push	<ol style="list-style-type: none"> tag_list : <ul style="list-style-type: none"> Push to devices with tag1 and tag2 {"tags": ["tag1", "tag2"], "op": "AND"} Push to devices with tag1 or tag2 {"tags": ["tag1", "tag2"], "op": "OR"} Tag list cannot exceed 512 characters tag_rules : <ul style="list-style-type: none"> Tag combination push. You can set 'AND', 'OR' and 'NOT' rules Note: if both tag_rules and tag_list are specified, tag_list becomes invalid automatically. For parameter descriptions, please see Tag combination rules.
token	Single-device push	<p>token_list</p> <ul style="list-style-type: none"> If the parameter contains multiple tokens, messages are only pushed to the first token Format example: ["token1"] A token string cannot exceed 36 characters
token_list	Device list push	<p>token_list</p> <ul style="list-style-type: none"> A maximum of 1,000 tokens Format example: ["token1", "token2"] A token string cannot exceed 36 characters
account	Single-account push	<p>account_list</p> <ul style="list-style-type: none"> If the parameter contains multiple accounts, messages are only pushed to the first account Format example: ["account1"]
account_list	Account list push	<p>account_list</p> <ul style="list-style-type: none"> A maximum of 1,000 accounts Format example: ["account1", "account2"]
package_account_push	Account package	Required for uploading the account package to push

	push	
package_token_push	Token package push	Required for token package push

Tag combination rules

Field	Type	Required	Description
tag_rules	Array	Yes	Tag rule set. See tag_rules description

tag_rules description

Field	Type	Parent Item	Required	Description
tag_items	Array	tag_rules	Yes	Tag rule. See tag_items description
operator	String	tag_rules	Yes	Operator between elements in the <code>tag_rules</code> array. The operator of the first <code>tag_rules</code> element is invalid data. The operator of the second <code>tag_rules</code> element is the operator between the first and second <code>tag_rules</code> elements, and so on. Valid values: <ul style="list-style-type: none"> OR: OR operation AND: AND operation
is_not	Boolean	tag_rules	Yes	Whether to perform "NOT" operation on the calculation result of the <code>tag_items</code> array. <ul style="list-style-type: none"> true: yes false: no

tag_items description

Field	Type	Parent Item	Required	Description
tags	Array	tag_items	Yes	Specific tag value in string type, such as tag1 and guangdong.
is_not	Boolean	tag_items	Yes	Whether to perform "NOT" operation on the calculation result of the tag array. Valid values: <ul style="list-style-type: none"> true: yes false: no
tags_operator	String	tag_items	Yes	Operator for tag in <code>tags</code> . Valid values: <ul style="list-style-type: none"> OR: OR operation AND: AND operation
items_operator	String	tag_items	Yes	Operator between elements in the <code>tag_items</code> array. The <code>items_operator</code> of the first <code>tag_items</code> element is invalid. The <code>items_operator</code> of the second

				<p><code>tag_items</code> element is the operator between the first and second <code>tag_items</code> elements, and so on. Valid values:</p> <ul style="list-style-type: none"> OR: OR operation AND: AND operation
<code>tag_type</code>	String	<code>tag_items</code>	Yes	See tag_type value table

tag_type value table

Tag Name	tag_type Value	Sample Tag
Custom tag	<code>xg_user_define</code>	tag1, tag2
Application version	<code>xg_auto_version</code>	1.1.0, 1.2.0.1
Device district information	<code>xg_auto_province</code>	guangdong, shanghai
Active information	<code>xg_auto_active</code>	20200131, 20200201
XG SDK version	<code>xg_auto_sdkversion</code>	1.1.5.2, 1.1.5.3
System language	<code>xg_auto_systemlanguage</code>	zh, en
Mobile phone brand	<code>xg_auto_devicebrand</code>	Mi, Vivo
Mobile phone model	<code>xg_auto_deviceversion</code>	MI 9 SE, Vivo X9Plus
Country/Region	<code>xg_auto_country</code>	CN, SG

Note :

For more information on the usage, please see [Sample tag push](#).

- Full push: push to all devices

```
{
  "audience_type": "all"
}
```

- Tag push (using `tag_rules`): push to male users who were active on April 8, 2020 in Guangdong or Hunan province

```
{
  "audience_type": "tag",
  "tag_rules": [
    {
      "tag_items": [
        {
          "tags": [
```

```

"guangdong",
"hunan"
],
"is_not": false,
"tags_operator": "OR",
"items_operator": "OR",
"tag_type": "xg_auto_province"
},
{
"tags": [
"20200408"
],
"is_not": false,
"tags_operator": "OR",
"items_operator": "AND",
"tag_type": "xg_auto_active"
},
{
"tags": [
"male"
],
"is_not": false,
"tags_operator": "OR",
"items_operator": "AND",
"tag_type": "xg_user_define"
}
],
"operator": "OR",
"is_not": false
}
]
}

```

- Single-device push: push to the device with token1 as the token

```

{
"audience_type": "token",
"token_list": [
"token1"
]
}

```

- Device list push: push to devices with token1 and token2 as their tokens

```

{
"audience_type": "token_list",
"token_list": [
"token1",
"token2"
]
}

```

- Single-account push: push to device(s) bound to account1

```
{
  "audience_type": "account",
  "account_list": [
    "account1"
  ]
}
```

- Account list push: push to devices bound to account1 and account2

```
{
  "audience_type": "account_list",
  "account_list": [
    "account1",
    "account2"
  ]
}
```

message_type: message body type

The message types of different platforms may vary slightly. For more information, please see the table below:

Message Type	Description	Supported Platform	Feature Description
notify	Notification bar message	Android and iOS	Message is displayed in the notification bar
message	In-app message or silent message	Android (in-app message) iOS (silent message)	Notifications are not displayed in the notification bar. Due to vendor restrictions, Android in-app messages can only be delivered through the TPNS channel and cannot be delivered through vendor channels

message: message body

The message body is the message delivered to the client.

Push API handles messages on iOS and Android differently, so you need to implement message pushes for the two platforms separately. The push message body is in JSON format.

General message on Android

The table below specifies fields for the Android platform:

Field Name	Type	Parent Item	Default Value	Required	Description
title	String	message	Empty	Yes	Message title.

content	String	message	Empty	Yes	Message content.
accept_time	Array	message	Empty	No	<p>The time period at which the message push is allowed to users.</p> <ul style="list-style-type: none"> A single element is formed by a "start" time and an "end" time. "start" and "end" are indicated by hour and minute. For more information, please see the samples. Note: this is only valid for the TPNS channel due to vendor restrictions.
thread_id	String	message	Empty	No	<p>Thread ID for collapsed notification in threaded display.</p> <ul style="list-style-type: none"> Note: this is only valid for the TPNS channel due to vendor restrictions.
thread_sumtext	String	message	Empty	No	<p>Summary displayed after the notification is collapsed in a thread, which is valid if <code>thread_id</code> is not empty.</p> <ul style="list-style-type: none"> Note: this is only valid for the TPNS channel due to vendor restrictions.
xg_media_resources	String	message	Empty	No	<p>URL of big image in notification bar, which takes effect only for the TPNS and Mi channels.</p> <ul style="list-style-type: none"> Note: to use the big image notification feature of the Mi channel, you need to call the Mi image uploading API to upload an image file, get the <code>pic_url</code> image address specified by Mi, and enter it in the parameter <code>xg_media_resources</code> of TPNS. For more information, please see the image uploading API section in Rich Text Message in Mi Push.
xg_media_audio_resources	String	message	Empty	No	<p>URL of audio rich media elements.</p> <ul style="list-style-type: none"> It supports audio in MP3 format and the suggested size is within 5 MB. Note: this parameter is only valid for TPNS channel, which is

					unsupported by other channels.
android	Object	message	Empty	No	Structure of advanced settings for Android notification. For more information, please see Android structure description

Android structure description

Field Name	Type	Parent Item	Default Value	Required	Description
n_ch_id	String	Android	Empty	No	Notification channel ID (only valid for the TPNS channel). For more information, please see Creating Notification Channel .
n_ch_name	String	Android	Empty	No	Notification channel name (only valid for the TPNS channel). For more information, please see Creating Notification Channel .
xm_ch_id	String	Android	Empty	No	Mi channel ID (only valid for the Mi channel).
hw_ch_id	String	Android	Empty	No	Huawei channel ID (only valid for the Huawei channel).
oppo_ch_id	String	Android	Empty	No	OPPO channel ID (only valid for the OPPO channel).
vivo_ch_id	String	Android	0	No	Vivo channel ID (only valid for the Vivo channel). Valid values: 0: operation message; 1: system message
builder_id	Integer	Android	0	No	Local notification style identifier.
badge_type	Integer	android	-1	No	Notification badge, which takes effect only for Huawei devices. Valid values: <ul style="list-style-type: none"> -2: automatically increased by 1; -1: unchanged
ring	Integer	Android	1	No	Whether there is a ringtone. Valid values: <ul style="list-style-type: none"> 0: no 1: yes
ring_raw	String	Android	Empty	No	This specifies the name of the ringtone file in the <code>raw</code> directory of the Android project; no extension is needed.
vibrate	Integer	Android	1	No	Whether the device vibrates. Valid values: <ul style="list-style-type: none"> 0: no 1: yes
lights	Integer	Android	1	No	Whether the breathing light is used. Valid

					<p>values:</p> <ul style="list-style-type: none"> • 0: no • 1: yes
clearable	Integer	Android	1	No	Whether the notification bar can be cleared.
icon_type	Integer	Android	0	No	<p>Whether the notification bar thumbnail is an in-app icon or an uploaded icon. Valid values:</p> <ul style="list-style-type: none"> • 0: in-app icon • 1: uploaded icon <p>This parameter is supported only for the TPNS and Huawei channels.</p>
icon_res	String	Android	Empty	No	<p>URL of the uploaded notification thumbnail. This parameter is supported only for the TPNS and Huawei channels. For more information on the thumbnail formats, please see Rich Media Notification.</p>
style_id	Integer	Android	1	No	Whether the notification style with the specified number will be overwritten.
small_icon	String	Android	Empty	No	The icon that the message displays in the status bar. If not set, the application icon will be displayed.
icon_color	Integer	Android	0	No	Color of the icon in the notification bar, which is only valid for Google devices. A hexadecimal color value should be used, such as 0xFF00FFFF.
action	Object	Android	Yes	No	The action after the notification bar is clicked; the default action is to open the application.
action_type	Integer	Action	Yes	No	<p>One-click actions. Valid values:</p> <ul style="list-style-type: none"> • 1: opens activity or the application • 2: opens the browser • 3: opens Intent (recommended; for more information, please see Guide for redirection through Intent).
custom_content	String	Android	Empty	No	<p>User-defined parameter (required to be serialized to a JSON string).</p> <p>Note:</p> <ul style="list-style-type: none"> • To adapt to advanced vendor features, TPNS upgraded the Huawei push protocol to V4 on July 15, 2020, which does not support carrying custom parameters through the <code>custom_content</code> field.

					<ul style="list-style-type: none"> If you have integrated the Huawei channel, we recommend you use intent to carry custom parameters; otherwise, custom parameters cannot be delivered through the Huawei channel. If you need to continue using the original V2 protocol, please click Submit a Ticket in the top-right corner of the console, and you will be added to the allowlist.
show_type	Integer	Android	2	No	<p>Whether to display notification when the application is running in the foreground, which is displayed by default. This takes effect only for TPNS and FCM channels. Valid values:</p> <ul style="list-style-type: none"> 1: no 2: yes <p>Note: if the value is 1 and the application is running in the foreground, this push is imperceptible to end users, but arrival data will be reported.</p>

Below is an example of a complete message:

```
{
  "title": "xxx",
  "content": "xxxxxxxxx",
  "xg_media_resources": "xxx1", // Enter the URL of rich media elements, such as https://www.xx.com/img/bd_logo
  1.png?qua=high
  "xg_media_audio_resources": "xxx", // Enter the URL of the audio rich media elements, such as http://sc1.111tt
  t.cn/2018/1/03/13/396131227447.mp3
  "thread_id": "Activity_id",
  "thread_sumtext": "Operational activity",
  "accept_time": [
    {
      "start": { // Period start time,
        "hour": "13", // Start time in hour. Value range: [0:24)
        "min": "00" // Start time in minute. Value range: [0:60)
      },
      "end": { // Period end time
        "hour": "14", // End time in hour. Value range: [0:24)
        "min": "00" // End time in minute. Value range: [0:60)
      }
    }
  ],
  "start": {
    "hour": "00",
    "min": "00"
  },
  "end": {
    "hour": "09",
```

```

"min": "00"
}
}
],
"android": {
  "n_ch_id": "default_message",
  "n_ch_name": "default notification",
  "n_id": 0,
  "builder_id": 0,
  "ring": 1,
  "ring_raw": "ring",
  "badge_type": -1,
  "vibrate": 1,
  "lights": 1,
  "clearable": 1,
  "icon_type": 0,
  "icon_res": "xg",
  "style_id": 1,
  "small_icon": "xg",
  "action": {
    "action_type": 1, // Action type; 1. Open activity or application; 2. Open browser; 3. Open Intent
    "activity": "xxx",
    "aty_attr": { // activity attribute, only for action_type=1
      "if": 0, // Intent's Flag attribute
      "pf": 0 // PendingIntent's Flag attribute
    },
    "browser": {
      "url": "xxx ", // Only http and https are supported
      "confirm": 1 // Whether user's confirmation is required
    },
    "intent": "xxx" // The SDK must be version 1.0.9 or later. Configure the data tag in the client's intent and set the scheme attribute
  },
  "custom_content": "{¥"key¥":¥"value¥"}"
}
}

```

General message on iOS

The table below specifies fields for the iOS platform.

Field Name	Type	Parent Item	Default Value	Required	Description
title	String	message	Empty	Yes	Message title, which will overwrite the content in <code>title</code> under <code>alert</code> .
content	String	message	Empty	Yes	Message content, which will overwrite the content in <code>body</code> under <code>alert</code> .
thread_id	String	message	Empty	No	Thread ID for collapsed notification in threaded display.
ios	Object	message	Empty	Yes	iOS message structure. See iOS field

					description for more information.
xg_media_resources	String	message	Empty	No	URL of rich media elements such as image, audio, and video

iOS field description

Field Name	Type	Parent Item	Default Value	Required	Description
aps	Object	ios	Empty	Yes	APNs-specific field. For more information, please see Payload .
alert	Object	aps	Empty	Yes	It contains the title and message content.
badge_type	Integer	aps	Empty	No	The user-configured badge number. Valid values: <ul style="list-style-type: none"> -1: badge number does not change. -2: badge number automatically increases by 1. >=0: a custom badge number is configured.
category	String	aps	Empty	No	The action identifier displayed when the message is pulled down.
mutable-content	Integer	aps	Empty	No	This is an additional notification parameter that carries "mutable-content" during push. <ul style="list-style-type: none"> 1 means that the Service Extension supports iOS 10. Once enabled, the push details will include the arrival data report. Before using this feature, please see Notification Service Extension Use Instructions to implement the Service Extension API. If <code>mutable-content</code> is not carried, arrival data will not be reported.
sound	String	aps	Empty	No	The sound field is used as follows. <ol style="list-style-type: none"> To play the system default ringtone, use "sound":"default" To play local custom ringtone, use "sound":"chime.aiff" To mute, use "sound":"" or remove the <code>sound</code> field. Note: if you want to use a custom ringtone, the ringtone must be in Linear PCM, MA4 (IMA/ADPCM), alaw, or μLaw format, saved in the <code>bundle</code> directory of the project, and of no more than 30 sec in length; otherwise, the system default ringtone will be used.
custom_content	String	ios	Empty	No	Parameters for custom delivery, which must

					be serialized to a JSON string.
xg	String	ios	Empty	No	Reserved key, which should not be used.

Below is an example of a complete message:

```
{
  "title": "xxx",
  "content": "xxxxxxxx",
  "thread_id": "Activity_id",
  "xg_media_resources": "https://www.xx.com/img/bd_logo1.png",
  "ios": {
    "aps": {
      "alert": {
        "subtitle": "my subtitle"
      },
      "badge_type": 5,
      "category": "INVITE_CATEGORY",
      "sound": "default",
      "mutable-content": 1
    },
    "custom_content": "{¥"key¥":¥"value¥"}",
    "xg": "oops"
  }
}
```

In-app message on Android

In-app message is unique to the Android platform and not displayed in the notification bar of the mobile phone. It can be used to deliver controlled messages to users imperceptibly.

Note :

Due to vendor restrictions, Android in-app messages can only be delivered through the TPNS channel and cannot be delivered through vendor channels.

The table below specifies fields for the Android platform:

Field Name	Type	Parent Item	Default Value	Required	Description
title	String	message	Empty	Yes	Command description.
content	String	message	Empty	Yes	Command content.
android	Object	message	Empty	No	Android message structure.
custom_content	String	android	Empty	No	This must be serialized to a JSON string.
accept_time	Array	message	Empty	No	The time period at which the message push is allowed to users.

- A single element is formed by a "start" time and an "end" time.
- "start" and "end" are indicated by hour and minute. For more information, please see the samples.
- **Note:** this is only valid for the TPNS channel due to vendor restrictions.

Complete example:

```
{
  "title": "this is title",
  "content": "this is content",
  "android": {
    "custom_content": "{¥key¥}:¥value¥}"
  },
  "accept_time": [
    {
      "start": {
        "hour": "13",
        "min": "00"
      },
      "end": {
        "hour": "14",
        "min": "00"
      }
    },
    {
      "start": {
        "hour": "00",
        "min": "00"
      },
      "end": {
        "hour": "09",
        "min": "00"
      }
    }
  ]
}
```

Silent messages on iOS

Similar to in-app message on Android, silent message is unique to the iOS platform and not displayed. When the message arrives at the device, iOS will wake up the application for a period of time (less than 30 seconds) in the background to let the application handle the message logic.

The table below specifies the fields:

Field Name	Type	Parent Item	Default Value	Required	Description
aps	Object	ios	Empty	Yes	APNs-specific field, where the most

					important key-value pair is as follows: content-available: it identifies the message type (which must be 1) and cannot contain <code>alert</code> , <code>sound</code> , or <code>badge_type</code> fields. For more information, please see Payload .
ios	Object	message	Empty	Yes	iOS message structure.
custom_content	String	ios	Empty	No	This must be serialized to a JSON string.
xg	String	ios	Empty	No	Reserved key, which should not be used.

Complete example:

```
{
  "ios": {
    "aps": {
      "content-available": 1
    },
    "custom_content": "{¥"key¥":¥"value¥"}",
    "xg": "oops"
  }
}
```

Optional Parameters

Optional Push API parameters refer to advanced parameters that can be carried in a push message, except `audience_type`, `message_type`, and `message` fields.

Parameter Name	Type	Parent Item	Required	Default Value	Description
expire_time	Integer	None	No	259200 (72 hours)	Offline message retention duration (in seconds), up to 72 hours. <ul style="list-style-type: none"> If <code>expire_time</code> is equal to 0, it indicates a real-time message. If <code>expire_time</code> is greater than 0 and less than 800 seconds, the system will reset it to 800 seconds. If <code>expire_time</code> is greater than or equal to 800 seconds, the message will be retained according to the set value, up to 72 hours.

					<ul style="list-style-type: none"> The value set cannot exceed 2147483647; otherwise, the push will fail.
send_time	String	None	No	Current system time	<p>Push time. You can specify a push time in the next 90 days.</p> <ul style="list-style-type: none"> The format is yyyy-MM-DD HH:MM:SS. If the push time specified is earlier than the current server time, the push starts immediately. This field is supported only for full push and tag push.
multi_pkg	Boolean	None	No	false	<p>Multi-package name push: for an application that has multiple packages (such as for MyApp and Wandoujia), if you want the application in all channels to receive the push message, you can set this value to <code>true</code>.</p> <p>Note: this parameter controls the multi-package name push of the TPNS channel by default. To implement multi-package name push on vendor channels, please see Configuring vendor channel for multi-package name.</p>
loop_param	Object	None	No	0	<p>Loop push (related to full push and tag push). For more information on loop push, please see the loop_param field description below.</p>
group_id	String	None	No	tpns_yyyymmdd, where <code>yyymmdd</code> indicates the push date	<p>This field has been disused and will be deactivated subsequently. If you need to use the aggregate statistics feature, please use the push plan field (<code>plan_id</code>).</p>
plan_id	String	None	No	Empty	<p>Push plan ID. For more information on how to</p>

					create and use the push plan, click here .
tag_list	Object	None	Only required for tag push	Empty	<ul style="list-style-type: none"> Push to devices with tag1 and tag2: <code>{"tags": ["tag1", "tag2"], "op": "AND"}</code> Push to devices with tag1 or tag2: <code>{"tags": ["tag1", "tag2"], "op": "OR"}</code>
account_list	Array	None	Only required for single-account push and account list push	Empty	<p>For single-account push:</p> <ul style="list-style-type: none"> <code>audience_type</code> must be <code>account</code> Parameter format: <code>["account1"]</code> <p>For account list push:</p> <ul style="list-style-type: none"> Parameter format: <code>["account1", "account2"]</code> Up to 1,000 accounts
account_push_type	Integer	None	Optional for account push	0	<p>Push type. Valid values:</p> <ul style="list-style-type: none"> 0: push messages to the latest device of the account 1: push messages to all devices associated with the account.
token_list	array	None	Required by single-device push and device list push	Empty	<p>For single-device push:</p> <ul style="list-style-type: none"> <code>audience_type</code> must be <code>token</code> Parameter format: <code>["token1"]</code> <p>For device list push:</p> <ul style="list-style-type: none"> Parameter format: <code>["token1", "token2"]</code> Up to 1,000 tokens
push_speed	Integer	None	Only valid for full push and tag push	Empty	Push speed limit to X pushes per second. Value range of X: 1000-50000
collapse_id	Integer	None	No	System-assigned <code>collapse_id</code>	<ul style="list-style-type: none"> Message overwriting parameter. After the first push task has been scheduled and delivered, if the second push task carries the same <code>collapse_id</code>, it will stop the TPNS channel data in

					<p>the first push task that has not been delivered yet and will also overwrite the message in the first push task.</p> <ul style="list-style-type: none"> The <code>collapse_id</code> of a completed task can be obtained through the Querying Push Information for One Task API. Currently, this is supported only for full push, tag push, and account package push.
channel_rules	Array	None	No	Empty	<p>Push channel selection policy.</p> <ul style="list-style-type: none"> You can select the channels through which a push can be delivered. Messages are pushed through all channels by default. For more information on the push policy, please see Channel Policy. For the data structure of single elements in the <code>channel_rules</code> array, please see channel_rules parameter description below.
tpns_online_push_type	Integer	None	No	0	<p>Whether to push the message to online devices through the TPNS channel. Valid values:</p> <ul style="list-style-type: none"> 0: the message will be pushed to online devices through the TPNS channel by default 1: the message will not be pushed to online devices preferably through the TPNS channel.
force_collapse	Boolean	None	No	false	<p>Whether to deliver messages to OPPO or Vivo devices that do not support message overwriting. Valid values:</p> <ul style="list-style-type: none"> false: no

- true: yes.

Note :

The `collapse_id` parameter is subject to the following restrictions:

- This parameter cannot be customized, which is generated by TPNS.
- This parameter is supported only for the TPNS channel, APNs channel, Mi channel, Meizu channel, and Huawei devices on version EMUI 10 and later.
- For the Huawei channel, you must use [intent](#) to carry custom parameters to implement message overwriting. If you use `custom_content`, the API layer will block custom parameters.
- Currently, the OPPO and Vivo channels do not support message overwriting. When an overwriting message is created, delivery to the OPPO and Vivo channels can be disabled by setting the `force_collapse` field to `false`.

channel_rules parameter description

Field Name	Type	Required	Description
channel	String	Yes	Delivery push channel. Valid values: <ul style="list-style-type: none"> • hw: Huawei channel • xm: Mi channel • mz: Meizu channel • vivo: Vivo channel • oppo: OPPO channel • apns: channel
disable	Boolean	No	Whether to disable the channel specified in <code>channel</code> , which is enabled by default. Valid values: <ul style="list-style-type: none"> • true: disable • false: enable

loop_param parameter description

Field Name	Type	Required	Description
startDate	string	Yes	Loop interval start date in YYYY-MM-DD format, such as 2019-07-01. You can choose a date in the next 90 days.
endDate	String	Yes	Loop interval end date in YYYY-MM-DD format, such as 2019-07-07. You can choose a date in the next 90 days.
loopType	Integer	Yes	Loop type. Valid values: <ul style="list-style-type: none"> • 1: daily • 2: weekly • 3: monthly
loopDayIndexs	Array	Yes	For weekly loop, enter days [0-6] in a week. For daily loop, enter 0. If the value is [0, 1, 2], it indicates that the push will be done on

			Monday, Tuesday, and Wednesday every week.
dayTimes	Array	Yes	Specific push time in HH:MM:SS format. If the value is ["19:00:00", "20:00:00"], it indicates that the push will be done at 19:00 and 20:00 every day.

Response Parameters

Field Name	Type	Required	Description
seq	Integer	Yes	The same as the request packet (if the request packet is invalid JSON, this field is 0).
push_id	String	Yes	Push ID.
ret_code	Integer	Yes	Error code. For more information, please see the error codes table.
environment	String	Yes	The push environment specified by the user (only for iOS). Valid values: <ul style="list-style-type: none"> product: production environment dev: development environment
err_msg	String	No	Error message when an error occurs in the request
result	String	No	When the request is correct: <ul style="list-style-type: none"> If there is extra data to be returned, the result will be encapsulated in the field in JSON format. If there is no extra data, this field may not exist.

Samples

Android tag push request message (using tag_list)

```
{
  "audience_type": "tag",
  "tag_list": {
    "tags": [
      "tag1",
      "tag2"
    ],
    "op": "AND"
  },
  "multi_pkg": true,
  "push_speed": 50000,
  "channel_rules": [
    {
      "channel": "mz",
      "disable": true
    }
  ],
}
```

```
{
  "channel": "xm",
  "disable": false
},
],
"message_type": "notify",
"message": {
  "title": "Test title",
  "content": "Test content",
  "xg_media_resources": "xxx1", // Enter the URL of rich media elements, such as https://www.xx.com/img/bd_logo
  1.png?qua=high
  "xg_media_audio_resources": "xxx", // Enter the URL of the audio rich media elements, such as http://sc1.111tt
  t.cn/2018/1/03/13/396131227447.mp3
  "accept_time": [
    {
      "start": { // Period start time,
        "hour": "13", // Start time in hour. Value range: [0:24)
        "min": "00" // Start time in minute. Value range: [0:60)
      },
      "end": { // Period end time
        "hour": "14", // End time in hour. Value range: [0:24)
        "min": "00" // End time in minute. Value range: [0:60)
      }
    }
  ],
  {
    "start": {
      "hour": "00",
      "min": "00"
    },
    "end": {
      "hour": "09",
      "min": "00"
    }
  }
],
"android": {
  "n_ch_id": "default_message",
  "n_ch_name": "default notification",
  "n_id": 0,
  "builder_id": 0,
  "ring": 1,
  "ring_raw": "ring",
  "badge_type": -1,
  "vibrate": 1,
  "lights": 1,
  "clearable": 1,
  "icon_type": 0,
  "icon_res": "xg",
  "style_id": 1,
  "small_icon": "xg",
  "action": {
    "action_type": 1, // Action type; 1. Open activity or application; 2. Open browser; 3. Open Intent
    "activity": "xxx",
```

```

"aty_attr": {// activity attribute, only for action_type=1
"if": 0, // Intent's Flag attribute
"pf": 0 // PendingIntent's Flag attribute
},
"browser": {
"url": "xxx", // Only http and https are supported
"confirm": 1 // Whether user's confirmation is required
},
"intent": "xxx" // The SDK must be version 1.0.9 or later. Configure the data tag in the client's intent and set the scheme attribute
},
"custom_content": "{¥"key¥":¥"value¥"}"
}
}
}

```

Tag push response message

```

{
"seq": 0,
"environment": "product",
"ret_code": 0,
"push_id": "3895624686"
}

```

iOS single-device push request message

```

{
"audience_type": "token",
"environment": "dev",
"token_list": [ "05da87c0ae*****fa9e08d884aada5bb2"],
"message_type": "notify",
"message": {
"title": "Push title",
"content": "Push content",
"ios": {
"aps": {
>alert": {
"subtitle": "Push subtitle"
},
"badge_type": -2,
"sound": "Tassel.wav",
"category": "INVITE_CATEGORY"
},
"custom_content": "{¥"key¥":¥"value¥"}",
"xg": "oops"
}
}
}

```

Single-device push response message

```
{
  "seq": 0,
  "push_id": "427184209",
  "ret_code": 0,
  "environment": "dev",
  "err_msg": "",
  "result": "[0]"
}
```

Sample tag push (using tag_rules)

Sample 1: push a message to male users who were active on April 8, 2020 in Guangdong or Hunan province

Expression: (xg_auto_province.guangdong OR xg_auto_province.hunan) AND xg_auto_active.20200408 AND xg_user_define.male

```
{
  "audience_type": "tag",
  "tag_rules": [
    {
      "tag_items": [
        {
          "tags": [
            "guangdong",
            "hunan"
          ],
          "is_not": false, //Whether to perform "NOT" operation on the calculation result of the tags in `tags`. Valid values: true: yes; false: no
          "tags_operator": "OR", //Operator for tags in `tags`
          "items_operator": "OR", //Operator between elements in `tag_items`. The `items_operator` of the first element is invalid data. The `items_operator` of the second element is the operator between the first and second elements, and so on.
          "tag_type": "xg_auto_province" //Type of tags in `tags`
        },
        {
          "tags": [
            "20200408"
          ],
          "is_not": false,
          "tags_operator": "OR",
          "items_operator": "AND",
          "tag_type": "xg_auto_active"
        },
        {
          "tags": [
            "male"
          ],
          "is_not": false,
          "tags_operator": "OR",
          "items_operator": "AND",

```

```

"tag_type": "xg_user_define"
}
],
"operator": "OR",
"is_not": false
}
]
}

```

Sample 2: push a message to Huawei users who were active in the last three days on application version other than 1.0.2

Expression: (xg_auto_active.20200406 OR xg_auto_active.20200407 OR xg_auto_active.20200408) AND (NOT xg_auto_version.1.0.2) AND xg_auto_devicebrand.huawei

```

{
"audience_type": "tag",
"tag_rules": [
{
"tag_items": [
{
"tags": [
"20200406",
"20200407",
"20200408"
],
"is_not": false, //Whether to perform "NOT" operation on the calculation result of the tags in `tags`. Valid values: true: yes; false: no
"tags_operator": "OR", //Operator for tags in `tags`
"items_operator": "OR", //Operator between elements in `tag_items`. The `items_operator` of the first element is invalid data. The `items_operator` of the second element is the operator between the first and second elements, and so on.
"tag_type": "xg_auto_active" //Type of tags in `tags`
},
{
"tags": [
"1.0.2"
],
"is_not": true,
"tags_operator": "OR",
"items_operator": "AND",
"tag_type": "xg_auto_version"
},
{
"tags": [
"huawei"
],
"is_not": false,
"tags_operator": "OR",
"items_operator": "AND",
"tag_type": "xg_auto_devicebrand"
}
],
"operator": "OR",

```

```
"is_not": false  
}  
]  
}
```

Creating Push Plan

Last updated : 2020-12-31 17:43:48

API Description

Request method: POST

Request address:

```
Service address/v3/push/plan/add_plan_push
```

The API service address corresponds to the service access point one by one; therefore, please select the service address corresponding to your application [service access point](#).

Note :

Call rate limit: 200 calls/day.

Request Parameters

Parameter	Type	Required	Description
planName	string	Yes	Push plan name, which can contain up to 23 letters
planDescribe	string	Yes	Push plan description, which can contain up to 300 characters

Response Parameters

Parameter	Type	Required	Description
retCode	string	Yes	Return status code
errMsg	string	Yes	Error message
result	object	Yes	Please see the description of <code>PlanRecordData</code> in the table below

PlanRecordData

Parameter	Type	Required	Description
planId	string	Yes	Push plan ID generated by TPNS
planName	string	Yes	Push plan name, which can contain up to 23 letters
planDescribe	string	Yes	Push plan description, which can contain up to 300 characters

planPushSource	string	Yes	-
createdAt	uint32	Yes	Creation time in timestamp format
updatedAt	uint32	Yes	Update time in timestamp format

Samples

Sample request

```
{
  "planName": "TPNS_TEST123",
  "planDescribe": "plan_test"
}
```

Sample response

```
{
  "retCode": 0,
  "ErrMsg": "save success",
  "result": {
    "planId": "48",
    "planName": "TPNS_TEST123",
    "planDescribe": "plan_test",
    "planPushSource": "api",
    "createdAt": 1593314408,
    "updatedAt": 1593314408
  }
}
```


Token Package Upload API

Last updated : 2020-12-18 19:59:28

API Description

Request method: POST.

Service address/v3/push/package/upload

The API service address corresponds to the service access point one by one; therefore, please select the service address corresponding to your application [service access point](#).

Feature: you can upload a token package file for a batch of device tokens, and messages will be pushed by the tokens in it.

Request Parameters

Parameter Name	Type	Required	Description
file	form-data	Yes	<ul style="list-style-type: none"> Token package format and size: .zip file within 100 MB The zipped package can contain a single .txt or .csv file but not folders .txt file requirements: (1) encoded in UTF-8; (2) one Token (36 bits) per row .csv file requirements: (1) one column only; (2) one Token (36 bits) per row

Response Parameters

Parameter Name	Type	Required	Description
retCode	Integer	Yes	Error code
errMsg	String	Yes	Error message when an error occurs in the request
uploadId	Integer	Yes	When a file is uploaded, the uploadId is returned as a positive integer, which represents the ID of the uploaded file. It is provided to the push API for Token package pushes

Sample Request

Curl sample

```
curl -X POST
https://api.tpsns.tencent.com/v3/push/package/upload

-H 'Authorization: basic application authorization information'
-F 'file=@C:¥Absolute path of the uploaded file'
```

Python sample

```
import requests

url = "https://api.tpsns.tencent.com/v3/push/package/upload"

files = {'file': open('file name', 'rb')}
upload_data={"fileName": "absolute file address"}

headers = {
'Authorization': "Basic application authorization information",
}

response = requests.request("POST", url, data=upload_data, headers=headers, files=files, verify=False)
print(response.text.encode('utf-8'))
```

Note :

For application authorization information, please see [Basic Auth Verification](#).

Account Package Upload API

Last updated : 2020-10-22 18:45:17

API Description

Request method: POST

```
request address/v3/push/package/upload
```

The API request address is corresponding to the service access point. Select the [request address](#) corresponding to the service access point of your application.

Feature: this API is used to batch upload account package files.

Note :

- The account package is in `.zip` format and less than 100 MB in size.
- The account package contains only one `.txt` or `.csv` file. Nested folders are not allowed.
- The `.txt` file is encoded using UTF-8. Each row contains only one account, and the account length is limited to [2, 100].
- The `.csv` file can contain only one column. Each row contains only one account, and the account length is limited to [2, 100].

Parameters

Request parameters

Parameter	Type	Required	Description
file	form-data	Yes	<ul style="list-style-type: none"> • The uploaded account package must be in <code>.zip</code> format and less than 100 MB in size. • The account package contains only one <code>.txt</code> or <code>.csv</code> file. Nested folders are not allowed. • The <code>.txt</code> file is encoded using UTF-8. Each row contains only one account, and the account length is limited to [2, 100]. • The <code>.csv</code> file can contain only one column. Each row contains only one account, and the account length is limited to [2, 100].

Response parameters

Parameter	Type	Required	Description
retCode	Integer	Yes	Error code.

errMsg	String	Yes	Error message when an error occurs in the request.
uploadId	Integer	Yes	When a file is uploaded, the uploadId is provided as a positive integer, which represents the ID of the uploaded file. It is provided to the account package API for subsequent pushes.

Sample

Curl sample

```
curl -X POST
https://api.tpsns.tencent.com/v3/push/package/upload

-H 'Authorization: Basic application authorization information'
-F 'file=@C:¥Absolute path of the uploaded file'
```

Python sample

```
import requests

url = "https://api.tpsns.tencent.com/v3/push/package/upload"

files = {'file': open('file name', 'rb')}
upload_data={"fileName": "absolute file address"}

headers = {
'Authorization': "Basic application authorization information",
}

response = requests.request("POST", url, data=upload_data, headers=headers, files=files, verify=False)
print(response.text.encode('utf-8'))
```

Note :

For more information on the application authorization, please see [Basic Auth Verification](#).

Tag APIs

Tag Binding and Unbinding

Last updated : 2020-10-22 18:45:17

API Description

Request method: POST.

```
request address/v3/device/tag
```

The API request address is corresponding to the service access point. Select the [request address](#) corresponding to the service access point of your application.

API feature

Tag API is the general term for all tag APIs. It includes various APIs for setting, updating, and deleting which are described as below:

For use cases of the tag feature, please see [Tag Feature Use Instructions](#).

Parameter Description

Request parameters

Parameter Name	Type	Required	Description
operator_type	Integer	Yes	Operation type: 1: add a single tag to a single token. 2: delete a single tag from a single token. 3: add multiple tags to a single token. 4: delete multiple tags from a single token. 5: delete all tags from a single token. 6: add one or more tags or custom tags to a single token in an overwriting manner. (This API only starts to set up new tags after the tag history is cleared, so calling it for the same token should be made at a certain interval (more than 1s recommended); otherwise, it may cause updating failure.) 7: add a single tag to multiple tokens. 8: delete a single tag from multiple tokens. 9: batch add tags (up to 20 pairs are allowed per call; in each tag-token pair, the tag is before the token). 10: batch delete tags (up to 20 pairs are allowed per call; in each tag-token pair, the tag is before the token).
platform	String	Yes	Client platform type:

			<ul style="list-style-type: none"> Android: Android iOS: iOS
token_list	Array	No	<p>Device list:</p> <ul style="list-style-type: none"> Required if <code>operator_type</code> is 1, 2, 3, 4, 5, 6, 7, or 8. If <code>operator_type</code> is 1, 2, 3, 4, 5, or 6, if this parameter contains multiple tokens, only the first one will be set. Format example: ["token1", "token2"] The list can contain up to 20 values. A token string cannot exceed 36 characters.
tag_list	Array	No	<p>Tag list:</p> <ul style="list-style-type: none"> Required if <code>operator_type</code> is 1, 2, 3, 4, 6, 7, or 8; ignored if <code>operator_type</code> is 5. If <code>operator_type</code> is 1, 2, 7, or 8, if this parameter contains multiple tags, only the first tag will be set if for a single tag. Format example: ["tag1", "tag2"] The list can contain up to 20 values. A tag string cannot exceed 50 characters.
tag_token_list	Array	No	<p>Tag-device list:</p> <ul style="list-style-type: none"> Required if <code>operator_type</code> is 9 or 10. Format example: [{"tag": "tag123", "token": "token123"}] In each pair, the tag must be before the token. The list can contain up to 20 values. A tag string cannot exceed 50 characters. A token string cannot exceed 64 characters.

- One application can have up to 10,000 custom `tags`, one device `token` can be bound to up to 100 customized `tags` (if you want to increase this limit, please submit a ticket), and one custom `tag` can be bound to an unlimited number of device tokens.
- If the API is being called just one time without setting up continuous tags, then there are no restrictions on the API call method.
- If the API is called by setting up continuous tags, then the following must be noted:
 - If there is a need to set up API calling by using continuous tags with more than 10 tags or more than 10 tokens, it is recommended that batch APIs be used. However, it is recommended that the interval between API calls be no less than 1s to guarantee correct tag operations.
 - If you are not using batch APIs, then the API for the next non-batched tag should be called after the returned value of the TPNS server is clearly obtained. It is not recommended that multi-thread async be used at the same time as performing tag API calling.
- Colon ":" is a keyword used by the TPNS backend for tag categorization. If ":" is included when a tag is set, then the string before the first ":" will be used as the category name of the tag. This is only used when `operator_type` is 6 (overwriting tags by category). For example, if the existing tags of a device are `level:1`, `level:2`, and `male`, you can directly use tag `level:3` to overwrite tags `level:1` and `level:2` without

having to unbind `level:1` and `level:2` one by one, and then bind tag `level:4` . For more information, please see the following sample request where `operator_type` is 6.

Response parameters

Field name	Type	Required	Comments
<code>ret_code</code>	Integer	Yes	Error code. For more information, please see the error codes table
<code>err_msg</code>	String	No	Error message when an error occurs in the request
<code>result</code>	String	No	When the request is correct: <ul style="list-style-type: none">If there is extra data to be returned, the result will be encapsulated in the JSON of this field. If there is no extra data,there may be no such field

Samples

Sample tag binding and unbinding requests

- Add a single tag (tag1) to a single token (token1)

```
{
  "operator_type": 1,
  "platform": "android",
  "tag_list": [
    "tag1"
  ],
  "token_list": [
    "token1"
  ]
}
```

- Delete a single tag (tag1) from a single token (token1)

```
{
  "operator_type": 2,
  "platform": "android",
  "tag_list": [
    "tag1"
  ],
  "token_list": [
    "token1"
  ]
}
```

- Add multiple tags (tag1 and tag2) to a single token (token1)

```
{
  "operator_type": 3,
  "platform": "android",
  "tag_list": [
    "tag1",
    "tag2"
  ],
  "token_list": [
    "token1"
  ]
}
```

- Delete multiple tags (tag1 and tag2) from a single token (token1)

```
{
  "operator_type": 4,
  "platform": "android",
  "tag_list": [
    "tag1",
    "tag2"
  ],
  "token_list": [
    "token1"
  ]
}
```

- Delete all tags from a single token (token1)

```
{
  "operator_type": 5,
  "platform": "android",
  "tag_list": [
    "tag1",
    "tag2"
  ],
  "token_list": [
    "token1"
  ]
}
```

- Tag overwrites custom tags of token1.

```
{
  "operator_type": 6,
  "platform": "android",
  "tag_list": [
    "test:2",
    "level"
  ],
  "token_list": [

```



```

"token1"
]
}

```

If one or more of the tags do not contain ":", then the `test:2` and `level` tags overwrite all custom tags of token1.

- Batch overwrite custom tags of a single token (this API only overwrites custom but not all tags)

```

{
  "operator_type": 6,
  "platform": "android",
  "tag_list": [
    "test:2",
    "level:2"
  ],
  "token_list": [
    "token1"
  ]
}

```

If all the tags have ":", as the string before the first ":" is the tag category, only tags in the same category corresponding to the device will be overwritten; for example, `test:2` will overwrite `test:*`, and `level:2` will overwrite `level:*`, without affecting other tags.

- Add a single tag (tag1) to multiple tokens (token1 and token2)

```

{
  "operator_type": 7,
  "platform": "android",
  "tag_list": [
    "tag1"
  ],
  "token_list": [
    "token1",
    "token2"
  ]
}

```

- Delete a single tag (tag1) from multiple tokens (token1 and token2)

```

{
  "operator_type": 8,

```

```
"platform": "android",
"tag_list": [
  "tag1"
],
"token_list": [
  "token1",
  "token2"
]
}
```

- Batch set tags

```
{
  "operator_type": 9,
  "platform": "android",
  "tag_token_list": [
    {
      "tag": "tag1",
      "token": "token1"
    }
  ]
}
```

- Batch delete tags (tag1, tag2, and tag3)

```
{
  "operator_type": 10,
  "platform": "android",
  "tag_token_list": [
    {
      "tag": "tag1",
      "token": "token1"
    },
    {
      "tag": "tag2",
      "token": "token2"
    },
    {
      "tag": "tag3",
      "token": "token3"
    }
  ]
}
```

Sample tag setting request

```
POST /v3/device/tag HTTP/1.1
Host: api.tpns.tencent.com
Content-Type: application/json
Authorization: Basic YTViNWYwNzFmZjc3YTplYTUxMmViNzZjc3Y2FmYTcwZTc3MQ==
Cache-Control: no-cache
```

```
Postman-Token: 4b82a159-afdd-4f5c-b459-de978d845d2f
{
  "operator_type": 1,
  "platform": "android",
  "tag_list": [
    "tag1"
  ],
  "token_list": [
    "token1"
  ]
}
```

Sample tag setting response

```
{
  "seq": 0,
  "ret_code": 0
}
```

Deleting All Devices Under Tag

Last updated : 2020-10-22 18:45:17

API Description

```
request address/v3/device/tag/delete_all_device
```

The API request address is corresponding to the service access point. Select the [request address](#) corresponding to the service access point of your application.

Feature: this API is used to delete all device APIs under a certain tag.

Parameter Description

Request parameters

Parameter Name	Type	Required	Description
tag_list	Array	Yes	List of tags to be deleted: <code>"tag_list": ["test_tag_3_Ik0N0", "test_tag_2_Ik0N0"]</code>

Response parameters

Field name	Type	Required	Comments
seq	Integer	Yes	The same as the request packet (if the request packet is invalid JSON, this field is 0)
ret_code	Integer	Yes	Error code. For more information, please see the error codes table
err_msg	String	No	Error message when an error occurs in the request
result	String	No	When the request is correct: <ul style="list-style-type: none"> If there is extra data to be returned, the result will be encapsulated in the json of this field. If there is no extra data, there may be no such field

Samples

Sample request

```
{
  "tag_list": ["test_tag_3_Ik0N0", "test_tag_2_Ik0N0"]
}
```

```
}
```

Sample response

```
{  
  "ret_code": 0,  
  "err_msg": "",  
  "seq": 0,  
  "result": {"": ""}  
}
```

Account APIs

Account Binding and Unbinding

Last updated : 2020-10-22 18:45:17

API Description

Request method: POST.

```
request address/v3/device/account/batchoperate
```

The API request address is corresponding to the service access point. Select the [request address](#) corresponding to the service access point of your application.

Feature: this is an async API that is only responsible for task issuance, with real-time operation currently unsupported.

Parameter Description

Request parameters

Parameter Name	Type	Required	Description
operator_type	Integer	Yes	Operation type. Value range: [1,5] as detailed below: <ul style="list-style-type: none"> 1: bind token to an account. 2: bind token to an account in an overwriting manner. 3: unbind token from multiple accounts. 4: unbind token from all accounts. 5: unbind account from all tokens.
platform	String	Yes	Client platform type <ul style="list-style-type: none"> Android: Android iOS: iOS
account_list	Array	No	The collection of account IDs, which is valid if <code>operator_type</code> is 5, with each element being required, including <code>account</code> and <code>account_type</code> fields. For example: [{"account": "926@126.com"}, {"account": "1527000000"}]
token_list	Array	No	The collection of device IDs, which is valid if <code>operator_type</code> is 4 and is required
token_accounts	Array	No	It is valid if <code>operator_type</code> is 1, 2, or 3 and must be called each time. Up to 20 tokens are allowed, and each <code>token_account</code> consists of 1 token and 1 <code>account_list</code> . Below is an example:

```
[{"token": "token1", "account_list": [{"account": "926@126.com"}, {"account": "1527000000"}]}, {"token": "token2", "account_list": [{"account": "926@163.com"}, {"account": "1527000001"}]}]
```

Response parameters

Parameter Name	Type	Description
ret_code	Integer	Return code
err_msg	String	Error message
result	Array	Operation status code corresponding to each token or account ["0","1008006"]

Samples

Sample request

- Binds token to an account

```
{
  "operator_type": 1,
  "platform": "android",
  "token_accounts": [
    {
      "token": "token1",
      "account_list": [
        {
          "account": "926@126.com"
        },
        {
          "account": "1527000000"
        }
      ]
    },
    {
      "token": "token2",
      "account_list": [
        {
          "account": "926@163.com"
        },
        {
          "account": "1527000001"
        }
      ]
    }
  ]
}
```

- Binds token to an account in an overwriting manner

```
{
  "operator_type": 2,
  "platform": "android",
  "token_accounts": [
    {
      "token": "token1",
      "account_list": [
        {
          "account": "926@126.com"
        },
        {
          "account": "1527000000"
        }
      ]
    },
    {
      "token": "token2",
      "account_list": [
        {
          "account": "926@163.com"
        },
        {
          "account": "1527000001"
        }
      ]
    }
  ]
}
```

- Unbind token from an account

```
{
  "operator_type": 3,
  "platform": "android",
  "token_accounts": [
    {
      "token": "token1",
      "account_list": [
        {
          "account": "926@126.com"
        },
        {
          "account": "1527000000"
        }
      ]
    },
    {
      "token": "token2",
      "account_list": [
        {
          "account": "926@163.com"
        },
      ]
    }
  ]
}
```



```
{
  "account": "1527000001"
}
]
}
]
}
```

- Unbinds token from all accounts

```
{
  "operator_type": 4,
  "platform": "android",
  "token_list": [
    "token1",
    "token2",
    "token3"
  ]
}
```

- Unbinds account from all tokens

```
{
  "operator_type": 5,
  "platform": "android",
  "account_list": [
    {
      "account": "926@126.com"
    },
    {
      "account": "1527000000"
    }
  ]
}
```

Sample return

Binds token to an account

```
{
  "result": [
    "ok",
    "ok"
  ],
  "ret_code": 0,
  "err_msg": "ok"
}
```

Querying Account and Device Binding

Last updated : 2020-10-22 18:45:17

API Description

Request method: POST.

```
request address/v3/device/account/query
```

The API request address is corresponding to the service access point. Select the [request address](#) corresponding to the service access point of your application.

Feature: this API is used to query the binding relationships between accounts and tokens.

Parameter Description

Request parameters

Parameter Name	Type	Required	Description
operator_type	Integer	Yes	Operation type: <ul style="list-style-type: none"> Batch queries the corresponding token according to account Queries the account according to token
account_list	Array	No	The account list to be queried, which is valid and required if <code>operator_type</code> is 1, with each element containing one set of accounts. Below is an example: <code>[{"account": "account1"}, {"account": "account2"}]</code>
token_list	Array	No	The token list to be queried, which is valid and required if <code>operator_type</code> is 2.

Response parameters

Parameter Name	Type	Description
retCode	Integer	Return code
errMsg	String	Error message
account_tokens	Array	Array of mappings from account to token, for example: <code>[{"account": "account1", "token_list": ["token1", "token2"]}, {"account": "account2", "token_list": ["token2", "token3"]}]</code>
token_accounts	Array	Array of mappings from token to account, for example: <code>[{"token": "token1", "account_list": [{"account": "926@126.com"}]}</code>

```
{
  "account": "1527000000",
},
{
  "token": "token2",
  "account_list": [
    {
      "account": "926@163.com",
    },
    {
      "account": "1527000001"
    }
  ]
}
```

Samples

Sample request

- Batch query the relationships of tokens bound to accounts

```
{
  "operator_type": 1,
  "account_list": [
    {
      "account": "account1"
    },
    {
      "account": "account2"
    }
  ]
}
```

- Batch query the relationships of accounts bound to tokens

```
{
  "operator_type": 2,
  "token_list": [
    "token1",
    "token2"
  ]
}
```

Sample return

- Batch query the tokens bound to accounts

```
{
  "retCode": 0,
  "errMsg": "ok",
  "result": [
    "0",
    "0"
  ],
  "account_tokens": [
    {
      "account": "account1",
      "token_list": [
        "token1",
        "token2"
      ]
    },
    {

```

```
"account": "account2",
"token_list": [
  "token2",
  "token3"
]
}
]
}
```

- Batch query the accounts bound to tokens

```
{
  "retCode": 0,
  "errMsg": "ok",
  "result": [
    "0",
    "0"
  ],
  "token_accounts": [
    {
      "token": "token1",
      "account_list": [
        {
          "account": "926@126.com"
        },
        {
          "account": "1527000000"
        }
      ]
    },
    {
      "token": "token2",
      "account_list": [
        {
          "account": "926@163.com"
        },
        {
          "account": "1527000001"
        }
      ]
    }
  ]
}
```

Statistics APIs

Querying Daily Aggregated Push Statistics

Last updated : 2020-10-22 18:45:17

API Description

Request method: POST.

Calling frequency limit: 200 times/hour.

```
request address/v3/statistics/get_push_channel_stat_overview
```

The API request address is corresponding to the service access point. Select the [request address](#) corresponding to the service access point of your application.

Feature: this API is used to query the daily push conversion summary within a certain time period by push channel.

Parameter Description

Request parameters

Parameter Name	Required	Type	Description
startDate	Yes	String	Query start date <ul style="list-style-type: none"> Format: YYYYMMDD Limit: only data in the last 30 days can be queried
endDate	Yes	String	Query end date. Format: YYYYMMDD

Response parameters

Parameter Name	Type	Description
retCode	Integer	Returned status code
errMsg	String	Error message
pushDateChannelStat	Array	Returned result: result array by day. Individual elements in the array represent daily statistics by channel, formed by <code>date</code> and <code>channelDatass</code> . The element structure of <code>channelDatass</code> is as follows

channelDatass

Parameter Name	Type	Description

channel	String	Push channel name. xg: TPNS channel hw: Huawei channel xm: Mi channel mz: Meizu channel oppo: OPPO channel vivo: Vivo channel apns: APNs channel fcm: FCM channel rog: ROG channel
pushState	Object	Push funnel statistics. For the data structures, please see <code>pushState</code> below

pushState (Android)

Parameter Name	Type	Description
pushActiveUv	Integer	Scheduled Number of available devices connected to the Internet within 90 days that meet the target push conditions and on which the notification bar is enabled.
pushOnlineUv	Integer	Sent Actual number of available devices in the scheduled devices that have been delivered to vendor channels or to process online terminal using TPNS channel.
arrivalUv	Integer	Number of reached devices (including arrival receipts for the TPNS and vendor channels. For the Huawei and Meizu channels, you need to add a configuration item for arrival receipt manually. For more information, please see Acquisition of Vendor Channel Arrival Receipt)
verifySvcUv	Integer	Number of reached devices (only valid for TPNS, ROG, and FCM channels. For other vendor channels, the <code>pushOnlineUv</code> metric of actual deliveries by TPNS will be used). Note: this field will be disused subsequently. You are recommended to check the <code>arrivalUv</code> field for arrival data
callbackVerifySvcUv	Integer	Arrival receipt for vendor channel (for the Huawei and Meizu channels, you need to add a configuration item for arrival receipt manually. For more information, please see Acquisition of Vendor Channel Arrival Receipt). Note: this field will be disused subsequently. You are recommended to check the <code>arrivalUv</code> field for arrival data
clickUv	Integer	Clicked
cleanupUv	Integer	Cleared

Note :

The "all" channel in the array corresponds to the aggregated statistics.

- In the aggregated statistics, the `verifySvcUv` (reached devices), `verifyUv` (displayed), `clickUv` (clicked), and `cleanupUv` (cleared) metrics only aggregates the data of the TPNS, ROG, and FCM channels.
- In the aggregated statistics, `pushActiveUv` (attempted) and `pushOnlineUv` (sent) aggregates the data of the TPNS channel and vendor channels.
- In the aggregated statistics, `callbackVerifySvcUv` (arrival receipt of vendor channel) aggregates the data of vendor channel's `callbackVerifySvcUv` (arrival receipt of vendor channel) + TPNS channel's `verifySvcUv` (reached devices) + ROG channel's `verifySvcUv` (reached devices) + FCM channel's `verifySvcUv` (device reached).

pushState (iOS)

Parameter Name	Type	Description
<code>pushActiveUv</code>	Integer	Attempted
<code>pushOnlineUv</code>	Integer	Successfully received by APNs
<code>verifySvcUv</code>	Integer	Reached
<code>clickUv</code>	Integer	Clicked

Samples

Sample request

```
{
  "startDate": "20200216",
  "endDate": "20200216"
}
```

Sample response

```
{
  "retCode": 0,
  "errMsg": "NO_ERROR",
  "pushDateChannelStat": [
    {
      "date": "20200216",
      "channelDatas": [
        {
          "channel": "xm",
          "pushState": {
            "pushActiveUv": 1000,
            "pushOnlineUv": 1000,
            "verifySvcUv": 1000,
            "callbackVerifySvcUv": 800,
            "arrivalUv": 1000,
            "verifyUv": 1000,

```

```
"clickUv": 0,
"cleanupUv": 0
},
{
"channel": "hw",
"pushState": {
"pushActiveUv": 1000,
"pushOnlineUv": 1000,
"verifySvcUv": 1000,
"callbackVerifySvcUv": 800,
"arrivalUv": 1000,
"verifyUv": 1000,
"clickUv": 0,
"cleanupUv": 0
}
},
{
"channel": "oppo",
"pushState": {
"pushActiveUv": 1000,
"pushOnlineUv": 1000,
"verifySvcUv": 1000,
"callbackVerifySvcUv": 800,
"arrivalUv": 1000,
"verifyUv": 1000,
"clickUv": 0,
"cleanupUv": 0
}
},
{
"channel": "xg",
"pushState": {
"pushActiveUv": 1000,
"pushOnlineUv": 800,
"verifySvcUv": 800,
"callbackVerifySvcUv": 0,
"arrivalUv": 1000,
"verifyUv": 800,
"clickUv": 300,
"cleanupUv": 500
}
},
{
"channel": "all",
"pushState": {
"pushActiveUv": 4000,
"pushOnlineUv": 3800,
"verifySvcUv": 3800,
"callbackVerifySvcUv": 2400,
"arrivalUv": 3800,
"verifyUv": 3800,
"clickUv": 300,
"cleanupUv": 500
}
```



```
}  
}  
]  
}  
]  
}
```

Querying Device Statistics

Last updated : 2020-10-22 18:45:17

API Description

Request method: POST.

Calling frequency limit: 200 times/hour.

```
request address/v3/statistics/get_device_stat_overview
```

The API request address is corresponding to the service access point. Select the [request address](#) corresponding to the service access point of your application.

Feature: this API is used to query "daily new devices", "daily connected devices", and "historically accumulated devices" of the application within a certain time period.

Parameter Description

Request parameters

Parameter Name	Required	Type	Description
startDate	Yes	String	Query start date <ul style="list-style-type: none">Format: YYYYMMDDLimit: only data in the last 3 months can be queried
endDate	Yes	String	Query end date. Format: YYYYMMDD

Response parameters

Parameter Name	Type	Description
retCode	Integer	Returned status code
errMsg	String	Error message
getDeviceStatOverviewData	Array	Returned result, with <code>getDeviceStatOverviewData</code> structure variables shown in following table

getDeviceStatOverviewData

Parameter Name	Type	Description
date	Integer	Data date
accuUv	Integer	Accumulated devices

newUv	Integer	Daily new devices
activeUv	Integer	Daily peak of online devices

Samples

Sample request

```
{
  "startDate": "20190724",
  "endDate": "20190726"
}
```

Sample response

```
{
  "retCode": 0,
  "errMsg": "",
  "getDeviceStatOverviewData": [
    {
      "date": 20190724,
      "accuUv": 0,
      "newUv": 0,
      "activeUv": 0
    },
    {
      "date": 20190725,
      "accuUv": 0,
      "newUv": 5,
      "activeUv": 0
    },
    {
      "date": 20190726,
      "accuUv": 5,
      "newUv": 0,
      "activeUv": 2
    }
  ]
}
```

Querying All Push Information in Specific Period

Last updated : 2020-10-22 18:45:18

API Description

Request method: POST

Calling frequency limit: 200 times/hour.

```
request address/v3/statistics/get_push_record
```

The API request address is corresponding to the service access point. Select the [request address](#) corresponding to the service access point of your application.

Feature: this API is used to query the basic information and settings of all tasks within a specified time range.

Parameter Description

Request parameters

Parameter Name	Required	Type	Description
startDate	Yes	String	Query start date, <ul style="list-style-type: none"> Format: YYYY-MM-DD Query limit: within the last 1 months
endDate	Yes	String	Query end date. Format: YYYY-MM-DD
msgType	No	String	Message type: <ul style="list-style-type: none"> notify: notification message: silent message
pushType	No	String	Push type: <ul style="list-style-type: none"> all: full push tag: tag push token: device list/device single push account: account list/account single push
offset	No	Integer	Start offset for paginated query
limit	No	Integer	Number of messages per page for paginated query (maximum value: 200)

Response parameters

--	--	--

Parameter Name	Type	Description
retCode	Integer	Returned status code
errMsg	String	Error message
pushRecordData	Array	Returned result, with <code>pushRecordData</code> structure variables shown in following table
count	Integer	Number of eligible records

pushRecordData

Parameter Name	Type	Description	Value Description
date	String	Push time	Format: YYYY-MM-DD hh:mm:ss
pushId	String	Message ID	-
title	String	Push title	-
content	String	Push content	-
status	String	Push status	<ul style="list-style-type: none"> PUSH_INIT //Task created PUSH_WAIT// Waiting for task to be scheduled PUSH_STARTED// Push started PUSH_FINISHED// Push finished PUSH_FAILED// Push failed PUSH_CANCELED// Push canceled by user PUSH_DELETED// Push deleted PUSH_REVOKED// Push revoked PUSH_COLLAPSED// Push overwritten PUSH_DELETED_PUSH_MSG// Push terminated
pushType	String	Push target	<ul style="list-style-type: none"> all //Full push tag //Tag push token_list //Device list account_list //Account list package_account_push //Number package push
messageType	String	Push type	<ul style="list-style-type: none"> notification //Notification message //Message
environment	String	Push environment	<ul style="list-style-type: none"> product //Production environment dev //Development environment
expireTime	Integer	Expiration time	Unit: second
xgMediaResources	String	Rich media information	-
multiPkg	Boolean	Whether it is	<ul style="list-style-type: none"> true // Enable multi-package name push

		multi-package name push	<ul style="list-style-type: none"> false // Disable multi-package name push
targetList	Array(String)	Push account or push device list	Valid if <code>pushType</code> is <code>token_list</code> or <code>account_list</code>
collapseID	Integer	Message overwriting ID	Valid if <code>pushType</code> is <code>all</code> , <code>tag</code> , or <code>package_account_push</code>
tagSet	Object	Tag settings	Valid if <code>pushType</code> is <code>tag</code> Data structure: <pre>{ "op": "OR", // Inter-tag logic operation "tagWithType": [{ "tagTypeName": "xg_user_define", // Tag type "tagValue": "test68" // Tag value }] }</pre>
uploadId	Integer	Number package ID	Valid if <code>pushType</code> is <code>package_account_push</code>
pushConfig	Object	Push configuration information	<p>"Android": for specific push configuration information related to Android, please see the following code</p> <p>"iOS": for specific push configuration related to iOS, please see the following code</p>

Configuration Information

Android push configuration information

```
"android": {
  "ring": 1, // Ring
  "vibrate": 1, // Vibrate
  "lights": 1, // LED indicator
  "clearable": 1, // Whether dismissible or not
  "action": {
    "action_type": 3, // Action type; 1. open activity or application; 2. open browser; 3. open Intent
    "intent": "" // The SDK version must be 1.0.9 or higher. Configure the data tag in the client's intent and set the scheme attribute
  },
  "custom_content": "{}"
}
```

iOS push configuration information

```
"ios": {
  "aps": {
```

```
"alert": {
  "subtitle": "my subtitle"
},
"badge_type": 5, // Badge number displayed by application (optional). -2: auto-increment, -1: unchanged,
"category": "INVITE_CATEGORY",
"sound": "default", // If this parameter is left empty, the default sound effect will be used
"mutable-content": 1
},
```

Samples

Sample request

```
{
  "limit": 50,
  "startDate": "2019-07-01",
  "endDate": "2019-08-01",
  "msgType": "notify",
  "pushType": "all",
  "offset": 0
}
```

Sample response

```
{
  "retCode": 0,
  "errMsg": "NO_ERROR",
  "count": 126,
  "pushRecordData": [
    {
      "date": "2019-11-18 11:26:54",
      "pushId": "12",
      "title": "test title",
      "content": "test log",
      "status": "PUSH_FINISHED",
      "pushType": "all",
      "targetList": null,
      "tagSet": null,
      "uploadId": 0,
      "groupId": "",
      "expireTime": 43200,
      "messageType": "notify",
      "xgMediaResources": "",
      "environment": "product",
      "pushConfig": {
        "android": {
          "n_id": 0,
          "builder_id": 0,
          "ring": 1,
          "ring_raw": ""
        }
      }
    }
  ]
}
```

```
"vibrate": 1,  
"lights": 1,  
"clearable": 1,  
"icon_type": 0,  
"icon_res": "",  
"style_id": 0,  
"small_icon": "",  
"action": {  
  "action_type": 3,  
  "activity": "",  
  "aty_attr": null,  
  "browser": null,  
  "intent": ""  
},  
"custom_content": ""  
},  
"ios": null,  
"iot": null  
},  
"multiPkg": true,  
"source": "api"  
}  
]  
}
```


Querying Push Information for One Task

Last updated : 2020-10-22 18:45:18

API Description

Request method: POST.

Calling frequency limit: 200 times/hour.

```
request address/v3/statistics/get_push_record
```

The API request address is corresponding to the service access point. Select the [request address](#) corresponding to the service access point of your application.

Feature: this API is used to query the basic information and settings of a task by using its pushid.

Parameter Description

Request parameters

Parameter Name	Required	Type	Description
pushId	Yes	String	Push task ID. Only the push task for the last one month can be queried.

Response parameters

Parameter Name	Type	Description
retCode	Integer	Returned status code
errMsg	String	Error message
pushRecordData	Array	Returned result, with <code>pushRecordData</code> structure variables shown in following table

pushRecordData

Parameter Name	Type	Description	Value Description
date	String	Push time	Format: YYYY-MM-DD hh:mm:ss
pushId	String	Message ID	-
title	String	Push title	-

content	String	Push content	-
status	String	Push status	<ul style="list-style-type: none"> PUSH_INIT //Task created PUSH_WAIT// Waiting for task to be scheduled PUSH_STARTED// Push started PUSH_FINISHED// Push finished PUSH_FAILED// Push failed PUSH_CANCELED// Push canceled by user PUSH_DELETED// Push deleted PUSH_REVOKED// Push revoked PUSH_COLLAPSED// Push overwritten PUSH_DELETED_PUSH_MSG// Push terminated
pushType	String	Push target	<ul style="list-style-type: none"> all // Full push tag // Tag push token_list // Device list account_list // Account list package_account_push // Number package push
messageType	String	Push type	<ul style="list-style-type: none"> notification //Notification message //Message
environment	String	Push environment	<ul style="list-style-type: none"> product //Production environment dev //Development environment
expireTime	Integer	Expiration time	Unit: second
xgMediaResources	String	Rich media information	-
multiPkg	Boolean	Whether it is multi-package name push	<ul style="list-style-type: none"> true // Enable multi-package name push false // Disable multi-package name push
targetList	Array(String)	Push account or push device list	Valid if <code>pushType</code> is <code>token_list</code> or <code>account_list</code>
collapseID	Integer	Message overwriting ID	Valid if <code>pushType</code> is <code>all</code> , <code>tag</code> , or <code>package_account_push</code>
tagSet	Object	Tag settings	Valid if <code>pushType</code> is <code>tag</code> Data structure: <pre> { "op": "OR", // Inter-tag logic operation "tagWithType": [{ "tagTypeName": "xg_user_define", // Tag type "tagValue": "test68" //Tag value}] } </pre>
uploadId	Integer	Number package ID	Valid if <code>pushType</code> is <code>package_account_push</code>

pushConfig	Object	Push configuration information	"Android": for specific push configuration information related to Android, please see the following code "iOS": for specific push configuration related to iOS, please see the following code
------------	--------	--------------------------------	--

Configuration Information

Android push configuration information

```

"android": {
  "ring": 1, // Ring
  "vibrate": 1, // Vibrate
  "lights": 1, // LED indicator
  "clearable": 1, // Whether dismissible or not
  "action": {
    "action_type": 3, // Action type; 1. open activity or application; 2. open browser; 3. open Intent
    "intent": "" // The SDK version must be 1.0.9 or higher. Configure the data tag in the client's intent and set the scheme attribute
  },
  "custom_content": "{}"
}
...

#### iOS push configuration information

```json
"ios": {
 "aps": {
 "alert": {
 "subtitle": "my subtitle"
 },
 "badge_type": 5, // Badge number displayed by application (optional). -2: auto-increment, -1: unchanged,
 "category": "INVITE_CATEGORY",
 "sound": "default", // If this parameter is left empty, the default sound effect will be used
 "mutable-content": 1
 },

```

## Samples

### Sample request

```

{
 "pushId": "133703"
}

```

### Sample response

```
{
 "retCode": 0,
 "errMsg": "NO_ERROR",
 "pushRecordData": [
 {
 "date": "2019-07-25 20:06:28",
 "pushId": 133703,
 "title": "1",
 "content": "2",
 "status": "PUSH_FINISHED",
 "pushType": "tag",
 "targetList": null,
 "tagSet": {
 "op": "OR",
 "tagWithType": [
 {
 "tagTypeName": "xg_user_define",
 "tagValue": "test68"
 }
]
 },
 "uploadId": 0,
 "expireTime": 86400,
 "messageType": "notify",
 "xgMediaResources": "",
 "environment": "product",
 "collapseID": 0,
 "pushConfig": {
 "android": {
 "ring": 1,
 "vibrate": 0,
 "lights": 1,
 "clearable": 1,
 "action": {
 "action_type": 1
 }
 },
 "custom_content": "{}"
 },
 "ios": null
 },
 {
 "multiPkg": false
 }
]
}
```

# Querying Push Statistics Information for One Task

Last updated : 2020-10-22 18:45:18

## API Description

**Request method:** POST.

**Calling frequency limit:** 200 times/hour.

```
request address/v3/statistics/get_push_task_stat_channel
```

The API request address is corresponding to the service access point. Select the [request address](#) corresponding to the service access point of your application.

**Feature:** this API is used to query the detailed statistics for each push task, including all channel information and summary results. The channel types in pushStatDataAll will be changed based on their differences in terms of iOS/Android and push channels.

## Parameter Description

### Request parameters

Parameter Name	Required	Type	Description
pushId	Yes	String	Message ID

### Response parameters

Parameter Name	Type	Description
retCode	Integer	Returned status code
errMsg	String	Error message
pushStatDataAll	Object	Returned result: individual element formed by <code>channel</code> and <code>pushState</code> , where <code>channel</code> is the push channel name. <code>pushState</code> structure variables are shown in following table

### PushState (Android)

Parameter Name	Type	Description
pushActiveUv	Integer	Scheduled Number of available devices connected to the Internet within 90 days that

		meet the target push conditions and on which the notification bar is enabled.
pushOnlineUv	Integer	Sent Actual number of available devices in the scheduled devices that have been delivered to vendor channels or to process online terminal using TPNS channel.
arrivalUv	Integer	Devices reached (including arrival receipts for TPNS and vendor channels. For the Huawei and Meizu channels, you need to configure the arrival receipts manually. For more information, see <a href="#">Acquisition of Vendor Channel Arrival Receipt</a> )
verifySvcUv	Integer	Devices reached (only for TPNS, ROG and FCM channels. Use the TPNS <code>pushOnlineUv</code> metric for other vendor channels.) <b>Note:</b> This field will be deprecated later. Use <code>arrivalUv</code> to get the arrival data
callbackVerifySvcUv	Integer	Vendor channel arrival receipt (For the Huawei and Meizu channels, you need to configure the arrival receipts manually. For more information, see <a href="#">Acquisition of Vendor Channel Arrival Receipt</a> ) <b>Note:</b> This field will be deprecated later. Use <code>arrivalUv</code> to get the arrival data
verifyUv	Integer	Display (obsolete)
clickUv	Integer	Click
cleanupUv	Integer	Dismissal

**Note:**

The "all" channel in the array corresponds to the aggregated statistics.

- In the aggregated statistics, the `verifySvcUv` (device reached), `verifyUv` (display), `clickUv` (click), and `cleanupUv` (dismissal) metrics only aggregates the data of the TPNS, ROG, and FCM channels.
- In the aggregated statistics, `pushActiveUv` (scheduled delivery) and `pushOnlineUv` (actual delivery) aggregates the data of the TPNS channel and vendor channels.
- In the aggregated statistics, `callbackVerifySvcUv` (arrival receipt of vendor channel) aggregates the data of vendor channel's `callbackVerifySvcUv` (arrival receipt of vendor channel) + TPNS channel's `verifySvcUv` (device reached) + ROG channel's `verifySvcUv` (device reached) + FCM channel's `verifySvcUv` (device reached).

**pushState (iOS and macOS)**

Parameter Name	Type	Description
pushActiveUv	Integer	Scheduled delivery
pushOnlineUv	Integer	APNs successfully received
verifySvcUv	Integer	Arrival

clickUv

Integer

Click

## Samples

### Sample request

```
{
 "pushId": "130248"
}
```

### Sample response

```
{
 "retCode": 0,
 "errMsg": "NO_ERROR",
 "pushStatDataAll": [
 {
 "channel": "xm",
 "pushState": {
 "pushActiveUv": 1000,
 "pushOnlineUv": 1000,
 "verifySvcUv": 1000,
 "callbackVerifySvcUv": 800,
 "arrivalUv": 1000,
 "verifyUv": 1000,
 "clickUv": 0,
 "cleanupUv": 0
 }
 },
 {
 "channel": "mz",
 "pushState": {
 "pushActiveUv": 1000,
 "pushOnlineUv": 1000,
 "verifySvcUv": 1000,
 "callbackVerifySvcUv": 800,
 "arrivalUv": 1000,
 "verifyUv": 1000,
 "clickUv": 0,
 "cleanupUv": 0
 }
 },
 {
 "channel": "vivo",
 "pushState": {
 "pushActiveUv": 1000,
 "pushOnlineUv": 1000,
 "verifySvcUv": 1000,
 "callbackVerifySvcUv": 800,
 "arrivalUv": 1000,
 "verifyUv": 1000,
 "clickUv": 0,
 "cleanupUv": 0
 }
 }
]
}
```

```
"clickUv": 0,
"cleanupUv": 0
},
{
"channel": "hw",
"pushState": {
"pushActiveUv": 1000,
"pushOnlineUv": 1000,
"verifySvcUv": 1000,
"callbackVerifySvcUv": 800,
"arrivalUv": 1000,
"verifyUv": 1000,
"clickUv": 0,
"cleanupUv": 0
},
},
{
"channel": "xg",
"pushState": {
"pushActiveUv": 1000,
"pushOnlineUv": 800,
"verifySvcUv": 800,
"callbackVerifySvcUv": 0,
"arrivalUv": 1000,
"verifyUv": 800,
"clickUv": 300,
"cleanupUv": 500
},
},
{
"channel": "oppo",
"pushState": {
"pushActiveUv": 1000,
"pushOnlineUv": 1000,
"verifySvcUv": 1000,
"callbackVerifySvcUv": 800,
"arrivalUv": 1000,
"verifyUv": 1000,
"clickUv": 0,
"cleanupUv": 0
},
},
{
"channel": "fcm",
"pushState": {
"pushActiveUv": 0,
"pushOnlineUv": 0,
"verifySvcUv": 0,
"callbackVerifySvcUv": 0,
"arrivalUv": 0,
"verifyUv": 0,
"clickUv": 0,
"cleanupUv": 0
}
```



```
}
},
{
 "channel": "rog",
 "pushState": {
 "pushActiveUv": 0,
 "pushOnlineUv": 0,
 "verifySvcUv": 0,
 "callbackVerifySvcUv": 0,
 "arrivalUv": 0,
 "verifyUv": 0,
 "clickUv": 0,
 "cleanupUv": 0
 }
},
{
 "channel": "all",
 "pushState": {
 "pushActiveUv": 6000,
 "pushOnlineUv": 5800,
 "verifySvcUv": 5800,
 "callbackVerifySvcUv": 4000,
 "arrivalUv": 5800,
 "verifyUv": 5800,
 "clickUv": 300,
 "cleanupUv": 500
 }
}
]
}
```

# Querying Aggregated Statistics for Multiple Tasks

Last updated : 2020-11-06 11:31:51

## API Description

**Request method:** POST.

**Calling frequency limit:** 200 times/hour.

```
request address/v3/statistics/get_push_group_stat_channel
```

The API request address is corresponding to the service access point. Select the [request address](#) corresponding to the service access point of your application.

**Feature:** this API is used to query the aggregated statistics by push channel of all push tasks *\*with the same GroupID or planId \** in the last 7 days according to GroupID or planId .

## Parameter Description

### Request parameters

Parameter Name	Required	Type	Description
planId	Yes	String	Field for aggregated statistics of multiple tasks, using corresponding "planId" in push parameters
groupId	Yes	String	Field for aggregated statistics of multiple tasks, using corresponding "group_id" in push parameters. This field will be disused gradually
startDate	Yes	String	Query start date <ul style="list-style-type: none"> <li>Format: YYYYMMDD</li> <li>Limit: only data of push tasks in the last 7 days can be aggregated</li> </ul>
endDate	Yes	String	Query end date. Format: YYYYMMDD

### **Note :**

If both `planId` and `groupId` exist in the request parameter, the query will be based on `planId` by default.

### Response parameters

Parameter Name	Type	Description

retCode	Integer	Returned status code
errMsg	String	Error message
pushStatDataAll	Array	Returned result: individual element formed by <code>channel</code> and <code>pushState</code> , where <code>channel</code> is the channel name. <code>pushState</code> structure variables are shown in following table

**pushState (Android)**

Parameter Name	Type	Description
pushActiveUv	Integer	Attempted The number of available devices online in the last 90 days with the notification bar enabled in the push target devices.
pushOnlineUv	Integer	Sent The number of available devices to which the message was successfully delivered through the vendor or TPNS channel out of the attempted devices.
arrivalUv	Integer	Number of reached devices (including arrival receipts for the TPNS and vendor channels. For the Huawei and Meizu channels, you need to add a configuration item for arrival receipt manually. For more information, please see <a href="#">Acquisition of Vendor Channel Arrival Receipt</a> )
verifySvcUv	Integer	Number of reached devices (only valid for TPNS, ROG, and FCM channels. For other vendor channels, the <code>pushOnlineUv</code> metric of actual deliveries by TPNS will be used). <b>Note:</b> this field will be disused subsequently. You are recommended to check the <code>arrivalUv</code> field for arrival data
callbackVerifySvcUv	Integer	Arrival receipt for vendor channel (for the Huawei and Meizu channels, you need to add a configuration item for arrival receipt manually. For more information, please see <a href="#">Acquisition of Vendor Channel Arrival Receipt</a> ). <b>Note:</b> this field will be disused subsequently. You are recommended to check the <code>arrivalUv</code> field for arrival data
verifyUv	Integer	Displayed (this field has been disused and will be deactivated subsequently)
clickUv	Integer	Clicked
cleanupUv	Integer	Cleared

**Note :**

The "all" channel in the array corresponds to the aggregated statistics.

- In the aggregated statistics, the `verifySvcUv` (reached devices), `verifyUv` (displayed), `clickUv` (clicked), and `cleanupUv` (cleared) metrics only aggregates the data of the TPNS, ROG, and FCM channels.

- In the aggregated statistics, `pushActiveUv` (attempted) and `pushOnlineUv` (sent) aggregates the data of the TPNS channel and vendor channels.
- In the aggregated statistics, `callbackVerifySvcUv` (arrival receipt of vendor channel) aggregates the data of vendor channel's `callbackVerifySvcUv` (arrival receipt of vendor channel) + TPNS channel's `verifySvcUv` (reached devices) + ROG channel's `verifySvcUv` (reached devices) + FCM channel's `verifySvcUv` (device reached).

### pushState (iOS)

Parameter Name	Type	Description
<code>pushActiveUv</code>	Integer	Attempted
<code>pushOnlineUv</code>	Integer	Successfully received by APNs
<code>verifySvcUv</code>	Integer	Reached
<code>clickUv</code>	Integer	Clicked

## Samples

### Sample request

```
{
 "groupid": "pt:testGroup",
 "startDate": "20190912",
 "endDate": "20190912"
}
```

### Sample response

```
{
 "retCode": 0,
 "errMsg": "NO_ERROR",
 "pushStatDataAll": [
 {
 "channel": "fcm",
 "pushState": {
 "pushActiveUv": 0,
 "pushOnlineUv": 0,
 "verifySvcUv": 0,
 "callbackVerifySvcUv": 0,
 "arrivalUv": 0,
 "verifyUv": 0,
 "clickUv": 0,
 "cleanupUv": 0
 }
 },
 {

```

```
"channel": "rog",
"pushState": {
"pushActiveUv": 0,
"pushOnlineUv": 0,
"verifySvcUv": 0,
"callbackVerifySvcUv": 0,
"arrivalUv": 0,
"verifyUv": 0,
"clickUv": 0,
"cleanupUv": 0
}
},
{
"channel": "hw",
"pushState": {
"pushActiveUv": 0,
"pushOnlineUv": 0,
"verifySvcUv": 0,
"callbackVerifySvcUv": 0,
"arrivalUv": 0,
"verifyUv": 0,
"clickUv": 0,
"cleanupUv": 0
}
},
{
"channel": "xm",
"pushState": {
"pushActiveUv": 0,
"pushOnlineUv": 0,
"verifySvcUv": 0,
"callbackVerifySvcUv": 0,
"arrivalUv": 0,
"verifyUv": 0,
"clickUv": 0,
"cleanupUv": 0
}
},
{
"channel": "oppo",
"pushState": {
"pushActiveUv": 0,
"pushOnlineUv": 0,
"verifySvcUv": 0,
"callbackVerifySvcUv": 0,
"arrivalUv": 0,
"verifyUv": 0,
"clickUv": 0,
"cleanupUv": 0
}
},
{
"channel": "vivo",
"pushState": {
```

```
"pushActiveUv": 0,
"pushOnlineUv": 0,
"verifySvcUv": 0,
"callbackVerifySvcUv": 0,
"arrivalUv": 0,
"verifyUv": 0,
"clickUv": 0,
"cleanupUv": 0
},
{
"channel": "mz",
"pushState": {
"pushActiveUv": 0,
"pushOnlineUv": 0,
"verifySvcUv": 0,
"callbackVerifySvcUv": 0,
"arrivalUv": 0,
"verifyUv": 0,
"clickUv": 0,
"cleanupUv": 0
},
{
"channel": "xg",
"pushState": {
"pushActiveUv": 4641,
"pushOnlineUv": 4641,
"verifySvcUv": 4641,
"callbackVerifySvcUv": 0,
"arrivalUv": 0,
"verifyUv": 4639,
"clickUv": 3818,
"cleanupUv": 4200
},
{
"channel": "all",
"pushState": {
"pushActiveUv": 4641,
"pushOnlineUv": 4641,
"verifySvcUv": 4641,
"callbackVerifySvcUv": 4641,
"arrivalUv": 4641,
"verifyUv": 4639,
"clickUv": 3818,
"cleanupUv": 4200
}
}
]
}
```

# Server-Side Error Codes

Last updated : 2020-07-03 17:38:45

The common errors of the TPNS RESTful API mainly include the following:

Error code	English descriptor	Error description	Feedback API
1008001	paramter parse error	Parameter parse error	General API
1008002	missing parameter	A required parameter is missing	General API
1008003	auth failure	Verification and authentication failed	General API
1008004	invoke service error	Failed to call the internal service	General API
1008006	invalid token	Invalid token. Please check it	Tag operation and account operation APIs
1008007	invalid parameter	Parameter verification failed	General API
1008011	upload file error	File upload failed	Certificate upload API and number package upload API
1008012	upload empty file	The uploaded file is empty	Certificate upload API and number package upload API
1008013	parse certificate error	Certificate parse failed	Certificate upload API
1008015	push id not exist	The push task ID does not exist	Statistics query API
1008016	date param formate error	Incorrect date and time parameter format	Statistics query API
1008019	content may be evil	Determined to be non-compliant by the content security service	Push API
1008020	cert bundled check failed	Certificate package name verification failed	Certificate upload API
1008021	not correct p12 file	p12 certificate format verification failed	Certificate upload API
1008022	p12 password is not correct	Incorrect p12 certificate password	Certificate upload API

Error code	English descriptor	Error description	Feedback API
1008025	create app , platform exist	Application creation failed. The application already exists under the product	Application creating API
1008026	batch op, partly error	Batch operation partially failed	Account binding and query API
1008027	batch op, total error	Batch operation fully failed	Account binding and query API
1008028	request too fast	Limit exceeded	Statistics API
1008029	token is not valid	Invalid token	Single push API, account binding API, and tag binding API
1008030	app not paid	No payment has been made for the application	General error
1008031	app was expired	The application resource has been terminated	General error
10110008	no token, no account	The queried token and account do not exist	Account binding and query API
10010005	internal error	The push target does not exist	Push API
10010018	Push repeat	Repeated push	Push APIs for full push and tag push



# API (java)

Last updated : 2020-12-09 15:11:55

## SDK Description

This SDK is used to encapsulate the TPNS server APIs and communicate with the TPNS server. To use it, you only need to import the `XingeApp` package. This SDK mainly encapsulates push-related V3 APIs.

## Integration Methods

Import Maven dependencies:

```
<dependency>
<groupId>com.github.xingepush</groupId>
<artifactId>xinge</artifactId>
<version>1.2.4</version>
</dependency>
```

## Directions

### XingeApp API description

This class provides APIs for interaction with the TPNS server, which is constructed by `XingeApp.Builder` with the following parameters:

Parameter Name	Type	Required	Default Value	Description
appId	Integer	Yes	Empty	Push target <code>AccessID</code> , which can be obtained in the <a href="#">Product Management</a> page in the console
secretKey	String	Yes	Empty	Push key
proxy	Proxy	No	Proxy.NO_PROXY	This parameter can be set if a proxy needs to be configured
connectTimeOut	Integer	No	10s	Connection timeout period
readTimeOut	Integer	No	10s	Request timeout period
domainUrl	String	No	<code>https://openapi.xg.qq.com/</code>	API request service domain name address, which is the XG API address by default. Change the value to the <a href="#">service URL</a>

corresponding to your service access point

## Samples

```
XingeApp xingeApp = new XingeApp.Builder()
 .appId(appid)
 .secretKey(secretKey)
 .domainUrl("https://api.tdns.tencent.com/")
 .build();

PushAppRequest pushAppRequest = new PushAppRequest();
// Complete the `PushAppRequest` message
...
JSONObject ret = xingeApp.pushApp(pushAppRequest);
```

## pushAppRequest API description

This class provides an encapsulated push message body. For the description and usage of parameters, please see [Push API Description](#).

## Server Return Codes

For the meaning of `ret_code` , please see [Server-Side Error Codes](#).

## FAQs

1. What should I do if the error code 10101 or 403 is returned?

Check whether the application `AccessID` matches with `SecretKey` , and whether `domainUrl` corresponds to the [service access point](#) of the application.

2. What should I do if the error code 1008007 indicating parameter verification failure is returned?

Refer to [Push API](#) and ensure that all the parameters are entered correctly.

# User Attribute APIs

Last updated : 2020-11-04 17:05:53

## API Description

**Request method:** POST.

```
service address/v3/device/set_custom_attribute
```

The API service address corresponds to the service access point one by one; therefore, please select the service address corresponding to your application [service access point](#).

**API feature:** this API is used to configured personalized attributes at the token level, including CRUD operations.

## Parameter Description

### Request parameters

Parameter Name	Required	Type	Description
cmd	Yes	Integer	Operation type: <ul style="list-style-type: none"> <li>1: adds attribute.</li> <li>2: updates attribute description.</li> <li>3: deletes attribute.</li> <li>4: deletes all attributes.</li> <li>5: queries attribute.</li> </ul>
accessId	Yes	Integer	Application ID, which can be obtained in the <a href="#">Product Management</a> page in the console.
token	Yes	String	Device ID of 36 bits. <ul style="list-style-type: none"> <li><a href="#">Suggestions on getting TPNS token (Android)</a></li> <li><a href="#">Suggestions on getting TPNS token (iOS)</a></li> </ul>
attributeInfo	Yes if cmd is 1, 2, or 3	Map	Attribute details: <ul style="list-style-type: none"> <li>value is attributeMap in Map type.</li> <li>key is attribute and attribute value. <b>Note:</b> you should have created an attribute in <a href="#">TPNS Console &gt; Toolbox &gt; User Attribute Management</a>; otherwise, this parameter will be filtered out, and invalidAttribute will be returned.</li> </ul>

### Response parameters

Parameter Name	Required	Type	Description

retCode	Yes	Integer	Error code. For more information, please see <a href="#">Server-Side Error Codes</a> .
errMsg	Yes	String	Error message when an error occurs in the request.
attributeInfo	cmd = 5	Map	Attribute details.
invalidAttribute	When the attribute is invalid	Array	Details of invalid attribute.

## Samples

### Sample request for adding attribute

```
{
 "cmd": 1,
 "accessId": 1500004469,
 "token": "04cac74a714f61bf089987a986363d88****",
 "attributeInfo": {
 "attributeMap": {
 "age": "100",
 "name": "Ming",
 "high": "2.66"
 }
 }
}
```

### Sample response for adding attribute

```
{
 "retCode": 0,
 "errMsg": "success",
 "invalidAttribute": [
 "high"
]
}
```