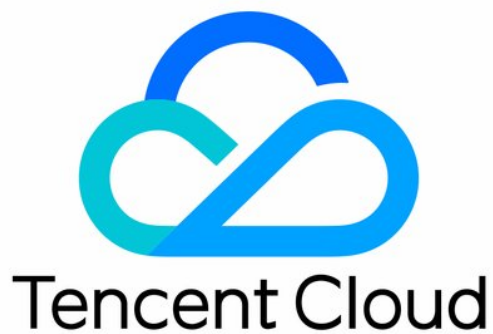


Tencent Push Notification Service

API Documentation

Product Documentation



Copyright Notice

©2013-2025 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 the Tencent corporate group, including its parent, subsidiaries and affiliated companies, as the case may be. 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

- Call Method

 - Signature Authentication (Recommended)

 - Basic Auth Authentication

 - Request Structure

- Push APIs

 - Push API

 - Creating a Push Plan

 - Account Package Upload API

 - Token Package Upload API

- Tag APIs

 - Tag Binding and Unbinding

 - Deleting All Devices Under Tag

- Account APIs

 - Account Binding and Unbinding

 - Account and Device Binding Query

- 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 for a Single Task

 - Querying Aggregated Statistics for Multiple Tasks

 - Querying All Push Tasks on the Day by Token

- User Attribute APIs

- Server-Side Error Codes

- Server SDKs

- API (Java)

API Documentation

Overview

Last updated : 2024-01-17 14:23:45

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 Tencent Push Notification Service Console. For more information, please see [Creating Products and Applications](#).

API Overview

Last updated : 2024-01-17 14:23:45

Push APIs

API Name	Description
v3/push/app	Pushes notification and in-app messages
v3/push/package/upload	Uploads number packages Uploads token packages
v3/push/plan/add_plan_push	Creates push plans

Tag APIs

API Name	Description
v3/device/tag	Binds and unbinds tags
v3/device/tag/delete_all_device	Deletes all devices under a tag

Account APIs

API Name	Description
v3/device/account/batchoperate	Binds and unbinds accounts
v3/device/account/query	Queries account and device binding

Statistics APIs

API Name	Description
v3/statistics/get_push_stat_overview	Pushes daily aggregated 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 the push information of a single task
v3/statistics/get_push_task_stat_channel	Queries the push statistics of a single task
v3/statistics/get_push_group_stat	Queries the aggregated push statistics of multiple tasks

User Attribute APIs

API Name	Description
v3/device/set_custom_attribute	User attribute API , used to create, delete, update, and query custom attributes at the token level

Call Method

Signature Authentication (Recommended)

Last updated : 2024-01-17 14:24:51

Overview

This document describes the signature authentication methods of Tencent Push Notification Service. The HMAC-SHA256 algorithm is used to generate 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 Tencent Push Notification Service backend, which can be obtained in Configuration Management > Basic Configuration in the Tencent Push Notification Service console
SecretKey	<code>SecretKey</code> assigned by the Tencent Push Notification Service backend, which corresponds to <code>AccessId</code> and can be obtained in Configuration Management > Basic Configuration in the Tencent Push Notification Service console
Sign	API signature
TimeStamp	Request timestamp in seconds

Signature Generation Method

1. Concatenate the request timestamp + `AccessId` + request body to get the original string to sign:

```
String to sign = ${TimeStamp} + ${AccessId}} + ${request body}
```

2. Use `SecretKey` as the key to sign the original string to generate a signature:

```
Sign = Base64(HMAC_SHA256(SecretKey, string to sign))
```

HTTP Request Description

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

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

The specific HTTP request message is as follows:

```
POST /v3/push/app HTTP/1.1
Host: api.tpns.tencent.com
Content-Type: application/json
AccessId: 1500001048
TimeStamp: 1565314789
Sign: Y2QyMDc3NDY4MmJmNzhiZmRiNDNlMTdkMWQ1ZDU2YjNlNWl3ODlhMTY3MGZjMTUyN2VmNTRjNjVzM
{"audience_type": "account", "message": {"title": "test title", "content": "test cont
```

Signature Generation Sample

1. Generate the string to sign as follows:

```
String to encrypt = 15653147891500001048{"audience_type": "account", "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"] }
```

Note:

The `${request body}` in the string to sign must be exactly the same as the data in the message body, including spaces and encoding.

2. Generate a hexadecimal hash based on the key through the HMAC-SHA256 algorithm, i.e., `secretKey` =1452fcebae9f3115ba794fb0fff2fd73 in the sample.

```
hashcode = hmac-sha256(SecretKey, string to sign)
Result:
hashcode="09f07d2a518a8814e369dcd9534f10b8a29d128531a19adaa28cb247605c1e84"
```

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

```
Sign=Base64(hashcode)
Sign="MDlmMDdkMmE1MTNhODgxNGUzNjlkY2Q5NTM0ZjEwYjhhMjlkMTI4NTMxYTE5YWRhYTI4Y2IyN
Dc2MDVjMWU4NA=="
```


Signature Code Samples in Various Languages

Python2

Python3

Java

Golang

C#

PHP

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

s = '15653147891500001048{"audience_type": "account","message": {"title": "test tit
key = '1452fcebae9f3115ba794fb0fff2fd73'
hashcode = hmac.new(key, s, digestmod=sha256).hexdigest()
print base64.b64encode(hashcode)

import hmac
import base64
from hashlib import sha256

s = '15653147891500001048{"audience_type": "account","message": {"title": "test tit
key = '1452fcebae9f3115ba794fb0fff2fd73'
hashcode = hmac.new(bytes(key, "utf-8"), bytes(s, "utf-8"),
                    digestmod=sha256).hexdigest()
print (base64.b64encode(bytes(hashcode, "utf-8")))
```

```
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\\": \\\"acc
        String appSecret = "1452fcebae9f3115ba794fb0fff2fd73";

        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 | UnsupportedEncodi
        e.printStackTrace();
    }
}

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

func TestSign(t *testing.T) {
    requestBody := "15653147891500001048{\\\\"audience_type\\": \\\"account\\", \\\"messa
    secretKey := "1452fcebae9f3115ba794fb0fff2fd73"

    h := hmac.New(sh256.New, []byte(secretKey))
    h.Write([]byte(requestBody))
    sha := hex.EncodeToString(h.Sum(nil))
    sign := base64.StdEncoding.EncodeToString([]byte(sha))
    println(sign)
}

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\\",\\"message\\": {\\"title\\": \\"
        string genSign = GenSign.genSign("1621307510", "1500004469", reqBody, "
        Console.WriteLine(genSign);
    }
    public static string HmacSHA256(string key, string data)
    {
        string hash;
        Byte[] code = System.Text.Encoding.UTF8.GetBytes(key);
        using (HMACSHA256 hmac = new HMACSHA256(code))
        {
            Byte[] d = System.Text.Encoding.UTF8.GetBytes(data);
            //Byte[] hmBytes = hmac.ComputeHash(encoder.GetBytes(data));
            Byte[] hmBytes = hmac.ComputeHash(d);
            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 r
    {
        string data = timeStampStr + accessId + requestBody;
        string hash = HmacSHA256(keySecret, data);
        string sign = Base64Encode(hash);
        Console.WriteLine("timeStampStr: " + timeStampStr + " accessId:" + acc
        return sign;
    }

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

```
}

...

<?php
$accessId = "1500001048";
$secretKey = "1452fceb9f3115ba794fb0fff2fd73";
$timeStamp = "1565314789";
$requestBody = "{\\"audience_type\\": \\"account\\",\\"message\\": {\\"title\\": \\"
$hashData = "{$timeStamp}{$accessId}{$requestBody}";
echo "reqBody: " . $hashData . "\\n";
//Get the SHA256 and hex results
$hashRes = hash_hmac("sha256", $hashData, $secretKey, false);
//Base64-encode the results
$sign = base64_encode($hashRes);
echo $sign . "\\n";
?>
...
```

Basic Auth Authentication

Last updated : 2024-01-17 14:24:51

This document describes the Basic Auth authentication methods of Tencent Push Notification Service.

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 a Key

1. Log in to the [Tencent Push Notification Service console](#), and select **Configuration Management > Basic Configuration** in the left sidebar.
2. Get `Access ID` and `SECRET KEY` from the Application Information column.

Generating a 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.

Sample:

```
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
```

Sample:

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

Request Structure

Last updated : 2024-01-17 14:24:51

Service URL

Tencent Push Notification Service 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 : 2024-01-17 14:24:51

API Calling Description

Request method: POST.

```
Service URL/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 supports different push targets. All request fields are JSON encapsulated and uploaded to the backend, which differentiates push targets based on the request parameters. If an error code is returned, see [Server-Side Error Codes](#).

Note:
Based on the needs of business optimization, Tencent Push Notification Service limits the frequency of **push to all devices, push to devices with specified tags, and push by account package** through the console and API to one message per second.
If the frequency is exceeded, a push exception may occur. If you need a higher frequency, [contact our online customer service](#).

Required Parameters

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

Parameter	Type	Required	Description
audience_type	String	Yes	Push target. Valid values: all: Push to all devices tag: Push to devices with specific tags token: Push to a single device token_list: Push to a list of devices account: Push to a single account account_list: Push to a list of accounts package_account_push: Push by account package package_token_push: Push by token package

message	Object	Yes	Message body. For more information, see [message body type](#message body type).
message_type	String	Yes	Message type. Valid values: notify: Notification message: In-app message/Silent message
environment	String	Yes (only for iOS)	Push environment (only available for pushes on iOS). Valid values: product: Production environment dev: Development environment The phone registration environment and push environment must be consistent. For more information, see Push Environment Selection Description
upload_id	Integer	Yes (only available for push by account/token package)	Account/Token package upload ID

audience_type: Push target

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

Push API supports a variety of push targets, such as all devices, devices with specified tags, a single device, a list of devices, a single account, or a list of accounts.

Push Target	Description	Required Parameters and Instructions
all	Push to all devices	None
tag	Push to devices with specific tags	<code>tag_rules</code> (recommended): Push by a combination of tags. You can set 'AND', 'OR' and 'NOT' rules. Note:If both <code>tag_rules</code> and <code>tag_list</code> are specified, <code>tag_list</code> becomes invalid automatically. For parameter descriptions, see Tag combination rules. <code>tag_list</code> (no further updates): Push to devices with tag1 and tag2 {"tags": ["tag1","tag2"],"op":"AND"} Push to devices with tag1 or tag2 {"tags": ["tag1","tag2"],"op":"OR"} A tag list cannot exceed 512 characters

token	Push to a single device	<code>token_list</code> If the parameter contains multiple tokens, messages are pushed to the device with the first token only. Format example: ["token1"] A token string cannot exceed 36 characters.
token_list	Push to a list of devices	<code>token_list</code> Up to 1,000 tokens Format example: ["token1","token2"] A token string cannot exceed 36 characters. Note: If the token list contains more than 1,000 tokens, the push will fail. To push messages to devices corresponding with more than 1,000 tokens, we recommend you use the Token Package Upload API .
account	Push to a single account	<code>account_list</code> If the parameter contains multiple accounts, messages are pushed to the first account only. Format example: ["account1"]
account_list	Push to a list of accounts	<code>account_list</code> Up to 1,000 accounts Format example: ["account1","account2"] Note: If the account list contains more than 1,000 accounts, the push will fail. To push messages to more than 1,000 accounts, you are recommended to use the Account Package Upload API .
package_account_push	Push by account package	Required for uploading the account package to push
package_token_push	Push by token package	Required for uploading the token package to push

Push to all devices

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

Push to devices with specified tags (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
}

```

Push to a single device: Push to the device with `token1` as the token

```

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

```

Push to a list of devices: Push to devices with `token1` and `token2` as their tokens

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

Push to a single account: Push to the device bound to `account1`

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

Push to a list of accounts: Push to devices bound to `account1` and `account2`

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

message_type: Message type

The message types may vary slightly by platform. For more information, see the table:

Message Type	Description	Supported Platform	Feature Description
notify	Notification bar message	Android and iOS	Messages are displayed in the notification bar. Note: This parameter is mutually exclusive with content-available: 1 . Do not use them at the same time.
message	In-app message or silent message	Android (in-app message) iOS (silent message)	Messages are not displayed in the notification bar. Note: Due to vendor restrictions, Android in-app messages can be delivered only through the Tencent Push Notification Service channel, but not 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 message pushes are implemented for the two platforms separately. The push message body is in the JSON format.

General message on Android

The table specifies parameters for the Android platform:

Parameter	Type	Parent Project	Default Value	Required	Description
title	String	message	Empty	Yes	Message title
content	String	message	Empty	Yes	Message content
accept_time	Array	message	Empty	No	The time period that allows pushes. A single element is formed by a "start" time and an "end" time. "start" and "end" are expressed in hour and minute. For more information, see the samples. Note: Due to vendor restrictions, this is valid only for the Tencent Push Notification Service channel.
thread_id	String	message	Empty	No	Thread ID for collapsed notification in threaded display. Note: Due to vendor restrictions, this is valid only for the Tencent Push Notification Service channel.
thread_sumtext	String	message	Empty	No	Summary displayed after the notification is collapsed in a thread, which is valid if thread_id is not empty. Note: Due to vendor restrictions, this is valid only for the Tencent Push Notification Service channel.

xg_media_resources	String	message	Empty	No	URL of large image in the notification bar, which takes effect only for the Tencent Push Notification Service and Mi channels. 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 pic_url image address specified by Mi, and enter it in the xg_media_resources parameter of Tencent Push Notification Service. For more information, see the image uploading API section in Rich Text Message in Mi Push .
xg_media_audio_resources	String	message	Empty	No	URL of audio rich media elements. It supports audio in MP3, with a recommended size not exceeding 5 MB. Note: This parameter is valid only for the Tencent Push Notification Service channel.
android	Object	message	Empty	No	Structure of advanced settings for Android notification. For more information, see Android structure description .

Android structure description

Parameter	Type	Parent Project	Default Value	Required	Description
n_ch_id	String	android	Empty	No	Notification channel ID (valid only for the Tencent Notification Service channel). For more information, see "Creating a notification channel" in API Document .
n_ch_name	String	android	Empty	No	Notification channel name (valid only for the Tencent Notification Service channel).

					Notification Service channel). For more information, see "Creating a notification channel" in API Document .
xm_ch_id	String	android	Empty	No	Mi channel ID (valid only for the Mi channel)
fcm_ch_id	String	android	Empty	No	FCM channel ID (valid only for the FCM channel)
hw_biz_type	Integer	android	0	No	Whether to enable notifications for Huawei quick messages. Valid values: 1: Enable 0: Disable Note: This parameter takes effect only for the Huawei quick message channel, and you need to contact Huawei business team .
hw_ch_id	String	android	Empty	No	Huawei channel ID (valid only for the Huawei channel)
hw_category	String	android	Empty	No	Huawei message type, which identifies the message type and accelerates sending messages of the corresponding type. For more information, see the category parameter in Sending Downlink Messages . IM: Instant messaging VOIP: Audio/video call SUBSCRIPTION: Subscription
hw_importance	Integer	android	Empty	No	Message reminding level. Valid values: 1: Silent reminder in the notification bar, where no sound and vibration are used to remind the user when a message arrives. 2: Strong reminder in the notification bar, where sound and vibration are used to remind the user when a message arrives. The actual message reminding method will be determined according to the hw_category field value or smart categorization result.
oppo_ch_id	String	android	Empty	No	OPPO channel ID (valid only for the OPPO channel)
vivo_ch_id	String	android	0	No	vivo channel ID (valid only for the vivo channel). 0 for operation message, and 1 for System message.
n_id	Integer	android	0	No	(This parameter has been disused and will be unavailable in the future. If you need the same feature, use the overriding parameter <code>cc_id</code>.) A unique ID of the notification message object (valid only for the Tencent Push Notification Service channel). (1) Greater than 0: Overrides the previous message ID. (2) Equal to 0: Displays this message without overriding the previous message ID.

					(3) Equal to -1: Clears all previous messages & message only.
builder_id	Integer	android	0	No	Local notification style identifier
badge_type	Integer	android	-1	No	Notification badge: -2: Automatically increased by 1, valid only for -1: Unchanged, valid only for Huawei and vivo [0, 100): Direct configuration, valid only for Hu devices Note: The badge adaptation capabilities vary vendor device. For details about the implemen each parameter value, see Badge Adaptation
ring	Integer	android	1	No	Whether there is a ringtone. Valid values: 0: No 1: Yes
ring_raw	String	android	Empty	No	Name of the ringtone file in the raw directory o project; no extension is needed. Note: Custom ringtones are supported only for FCM, and Tencent Push Notification Service c must be used with the field n_ch_id. For the cc process, see How do I set a custom ringtone?
vibrate	Integer	android	1	No	Whether to enable vibration. Valid values: 0: No 1: Yes
lights	Integer	android	1	No	Whether to use the breathing light. Valid value 0: No 1: Yes
clearable	Integer	android	1	No	Whether messages can be cleared from the n
icon_type	Integer	android	0	No	Whether the notification bar thumbnail is an in- online resource icon. Valid values: 0: In-app icon (for the Tencent Push Notificati channel only). 1: Online resource icon. This parameter is sup the Tencent Push Notification Service, FCM, H HONOR channels.
icon_res	String	android	N/A	No	Specifies image resource of the notification ba If icon_type is 0, enter the filename (without ex image resource in the Android application (for Push Notification Service channel only).

					If icon_type is 1, enter the URL of the thumbnail information on the thumbnail formats, see Rich Notification . This parameter is supported only Push Notification Service, FCM, Huawei, and channels.
style_id	Integer	android	1	No	Whether the notification style with the specified overwritten
small_icon	String	android	Empty	No	The icon that the message displays in the status bar. If the parameter is not set, the application icon will be used.
icon_color	Integer	android	0	No	Color of the icon in the notification bar This parameter takes effect only for the Tencent Push Notification Service channel. To use an RGB color such as #01e240, enter the hex value.
action	Object	android	Yes	No	The action after the notification bar is clicked; for example, to open the app. For more information, see Action description .
custom_content	String	android	N/A	No	A custom parameter (which should be serialized as a string). For how to obtain this parameter, see [How to Redirect-Getting parameters on the client] (https://www.tencentcloud.com/document/product/1001/34079) Note : Huawei officially announced that "the v2 protocol will be discontinued starting September 30, 2021". Tencent Push Notification Service has upgraded the Huawei channel to v5, which does not support carrying custom parameters through the extra parameter(s) field. If you have a Huawei channel, we recommend you use Inter parameters; otherwise, custom parameters can still be carried through the Huawei channel.
show_type	Integer	android	2	No	Whether to display the notification when the app is running in the foreground, which is displayed in the status bar. This parameter takes effect only for the Tencent Push Notification Service and FCM channels. Valid values: 1 : No 2 : Yes Note: If the value is 1 and the application is running in the foreground, this push is imperceptible to the user and no data will be reported.

action parameter description

Parameter	Type	Parent Project	Default Value	Required	Description
action_type	Integer	action	1	No	One-click actions. Valid values: 1: Open activity or the application 2: Open the browser 3: Open the application's custom page (recommended; for more information, see here).
activity	String	action	Empty	Yes if action_type is 1 and an activity needs to be opened	Full name of activity, such as com.x.y.PushActivity
aty_attr	Object	action	Empty	No if action_type is 1 and an activity needs to be opened	Activity attribute if: Flag attribute of Intent in Integer type pf: Flag attribute of PendingIntent in Integer type
browser	Object	action	Empty	Yes if action_type is 2	Action to open a browser url: Webpage URL in String type. Only HTTP and HTTPS URLs are supported confirm: Whether user's confirmation is required. The value is in Integer type. 1: Yes 0: No
intent	String	action	Empty	Yes if action_type is 3	Custom scheme, such as xgscheme://com.tpns.push/notify_detail

Below is a sample of a complete message:

```
{
  "title": "xxx",
  "content": "xxxxxxxxxx",
  "xg_media_resources": "xxx" , // Enter the URL of rich media elements, such as
  "xg_media_audio_resources": "xxx", // Enter the URL of audio rich media elements
  "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 applic
      "activity": "com.x.y.PushActivity",
      "aty_attr": { // Activity attribute, only for action_type=1
        "if": 0, // Intent's flag attribute
        "pf": 0 // PendingIntent's flag attribute
      },
      "browser": {
        "url": "https://cloud.tencent.com ", // Only HTTP a
        "confirm": 1 // Whether user's confirmation is requ

```

```
    },
    "intent": "xgscheme://com.tpins.push/notify_detail" //The SD
  },
  "custom_content": "{\\"key\\":\\"value\\"}"
}
```

Notification message on iOS

The table below specifies parameters for the iOS platform.

Parameter	Type	Parent Project	Default Value	Required	Description
title	String	message	Empty	Yes	Message title, which will override the content in title under alert.
content	String	message	Empty	Yes	Message content, which will override the content in body under alert.
thread_id	String	message	Empty	No	Thread ID for collapsed notification in threaded display
ios	Object	message	Empty	Yes	iOS message structure. See iOS parameter description for more information.
show_type	Integer	message	2	No	Whether to display the notification when the application is running in the foreground. Valid values: 1: No 2: Yes 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.
xg_media_resources	String	message	Empty	No	URL of rich media elements such as image, audio, and video. For more information, see Rich Media Notification .

iOS parameter description

--	--	--	--	--	--

Field	Type	Parent Project	Default Value	Required	Description
aps	Object	ios	Empty	Yes	APNs-specific parameter. For more information, see aps parameter description . For further information, see Payload .
custom_content	String	ios	Empty	No	Custom parameter for delivery, which must be serialized to a JSON string.

aps parameter description

Parameter	Type	Parent Project	Default Value	Required	Description
alert	Object	aps	Empty	Yes	Contains the title and message content.
badge_type	Integer	aps	Empty	No	User-configured badge number. Valid values: -1: The badge number does not change. -2: The badge number automatically increases by 1. ≥0: A custom badge number is configured.
category	String	aps	Empty	No	Action identifier displayed when the message is pulled down.
mutable-content	Integer	aps	1	No	This is an additional notification field that carries "mutable-content" during push. 1 means that the Service Extension supports iOS 10. Once enabled, the push details will include the arrival data report. Before using this feature, see Notification Service Extension to implement the Service Extension API. If mutable-content is not carried, arrival data will not be reported.
sound	String	aps	Empty	No	Use instructions: 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 sound parameter. 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 bundle directory of the project, and last for 30 seconds at most; otherwise, the system default ringtone will be used.
interruption-level	String	aps	active	No	Valid only for devices with iOS 15 or later. It needs to enable Time Sensitive Notifications in Capabilities. There are four interruption levels: passive: Indicates notifications not requiring immediate attention. active: Indicates default notifications. time-sensitive: Indicates notifications requiring immediate attention. critical: Indicates highly important notifications requiring immediate attention.

Below is a sample of a complete message:

```
{
  "title": "xxx",
  "content": "xxxxxxxxxx",
  "thread_id": "Activity_id",
  "xg_media_resources": "https://www.xx.com/img/bd_logo1.png",
  "show_type": 1,
  "ios": {
    "aps": {
      "alert": {
        "subtitle": "my subtitle"
      },
      "badge_type": 5,
      "category": "INVITE_CATEGORY",
      "sound": "default",
      "interruption-level": "time-sensitive",
      "mutable-content": 1
    },
    "custom_content": "{\\"key\\":\\"value\\"}"
  }
}
```

```
}
```

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 be delivered through only the Tencent Push Notification Service channel and cannot be delivered through vendor channels.

The table specifies parameters for the Android platform:

Parameter	Type	Parent Project	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
accept_time	Array	message	Empty	No	The time period that allows pushes. 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, see the samples. Note: This is valid only for the Tencent Push Notification Service channel due to vendor restrictions.
custom_content	String	android	Empty	No	Custom content, which must be serialized to a JSON string

Complete sample:

```
{
  "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"
      }
    }
  ]
}

```

iOS silent messages

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

The specific parameters are as follows:

Parameter	Type	Parent Project	Default Value	Required	Description
ios	Object	message	Empty	Yes	iOS message structure
aps	Object	ios	Empty	Yes	APNs-specific parameter, where the most important key-value pair is as follows: content-available: Identifies the message type (which must be 1), in integer. The value cannot contain the <code>alert</code>, <code>sound</code>, or <code>badge_type</code> parameters. For more information, see Payload. Note: <code>content-available: 1</code> is mutually exclusive with <code>[message_type:"notify"](#message</code>

					body type). Do not use them at the same time.
custom_content	String	ios	Empty	No	Custom content, which must be serialized to a JSON string.

Complete sample:

```
{
  "ios":{
    "aps": {
      "content-available": 1
    },
    "custom_content":"{\\\\"key\\":\\\\"value\\"}"
  }
}
```

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` .

Parameter	Type	Parent Project	Required	Default Value	Description
expire_time	Integer	None	No	86,400 (24 hours)	Offline message retention duration (in seconds), up to 72 hours If expire_time is 0, it indicates a real-time message. If expire_time is greater than 0 and less than 800s, the system will reset it to 800s. If expire_time is greater than or equal to 800s, the message is retained according to the set value, up to 72 hours. The value set cannot exceed 259200; otherwise, the push fails. To adjust the offline message retention duration, contact our online customer service .

send_time	String	None	No	Current system time	Push time. You can specify a time in the next 90 days. The format is yyyy-MM-DD HH:MM:SS. If the push time specified is earlier than the current server time, push starts immediately. This parameter is supported for push to all devices, push account package, or push to devices with specified tags.
multi_pkg	Boolean	None	No	false	Multi-package name push: For an application that has multiple packages (such as for MyAppWandoujia), if you want the application in all channels to receive the push message, you can set this value to true. Note: This parameter controls multi-package name push of Tencent Push Notification Service channel by default. To implement multi-package name push of vendor channels, see Configure vendor channel for multi-package name.
loop_param	Object	None	No	0	Loop push parameters. This parameter is supported only for push to all devices, push by account package, or push to devices with specified tags. For more information about loop parameters, see the loop parameter description below.
group_id	String	None	No	tpns_yyyymmdd, where yyyymmdd indicates the push date	This parameter has been disused and will be unavailable in the future. If you need the aggregate statistics feature, use the push plan parameter (plan_id).
plan_id	String	None	No	Empty	Push plan ID. For more information about how to create and use

					push plan, see Push Plan .
tag_rules	Array	None	Yes for tag push only	Empty	For push based on a combir of tags, you can set 'AND', 'OR' and 'NOT' rules Note: If both <code>tag_rules</code> and <code>tag_list</code> are specified <code>tag_list</code> becomes invalid automatically. For parameter descriptions, see tag_rules parameter description .
account_list	Array	None	Yes for push to a single account or a list of accounts	Empty	For push to a single account <code>audience_type</code> must be <code>account</code> Format: ["account1"] For push to a list of accounts Format: ["account1", "account2"] Up to 1,000 accounts
account_push_type	Integer	None	No for push by account	0	Push type. Valid values: 0: Push messages to the latest device of the account 1: Push messages to all devices associated with the account
account_type	Integer	None	No	0	Account type, which must be consistent with that of the accounts to push to. For valid values, see Account Type Value Table .
token_list	Array	None	Yes for push to a single device or a list of devices	Empty	For push to a single device: <code>audience_type</code> must be <code>token</code> Format: ["token1"] For push to a list of devices: Format: ["token1", "token2"] Up to 1,000 tokens
ignore_invalid_token	int	None	No	0	<code>0</code> : The API call will fail if there is an invalid token. <code>1</code> : Ignore the invalid token and continue to deliver.

					Note: This parameter takes only for push by a token or token list.
push_speed	Integer	None	No	Empty	Push speed limit to X pushes per second. Value range of X: 1, 50,000 This parameter is valid only for push to all devices, push by account package, and push to devices with specified tags.
collapse_id	Integer	None	No	System-assigned collapse_id	Message overriding parameter After the first push task has been scheduled and delivered, if a second push task carries the collapse_id, it will stop the Tencent Push Notification Service channel data in the first push task that has not been delivered, and will also overwrite the message in the first push task. The collapse_id of a complete task can be obtained via the Querying Push Information for One Task API. Currently, this is supported for push to all devices, push to devices with specified tags, and push by account package.
channel_rules	Array	None	No	Empty	Push channel selection policy You can select the channels through which a push can be delivered. Messages are pushed through all channels by default. For more information about the push policy, see Channel Policy . For the data structure of single elements in the channel_rules array, see [channel_rules field description](#channel_rules_description_1) below.
tpns_online_push_type	Integer	None	No	0	Whether to push the message to online devices through the

					Tencent Push Notification S channel. Valid values: <code>0</code> : Yes <code>1</code> : No
<code>force_collapse</code>	Boolean	None	No	false	Whether to deliver message OPPO or vivo devices that c support message overriding values: <code>false</code> : No <code>true</code> : Yes

Note:

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

This parameter currently cannot be customized. Only `collapse_id` values generated by Tencent Push Notification Service can be used.

This parameter is supported only for the Tencent Push Notification Service channel, APNs channel, Mi channel, Meizu channel, and Huawei devices on EMUI 10 or later.

For the Huawei channel, you must use [intent](#) to carry custom parameters to implement message overriding. If you use `custom_content`, the API layer will block custom parameters.

Currently, the OPPO and vivo channels do not support message overriding. When an overriding message is created, delivery to the OPPO and vivo channels can be disabled by setting the `force_collapse` parameter to `false`.

tag_rules parameter description

Parameter	Type	Parent Project	Required	Description
<code>tag_items</code>	Array	<code>tag_rules</code>	Yes	Tag rule. See tag_items description .
<code>operator</code>	String	<code>tag_rules</code>	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: <code>OR</code> : OR operation <code>AND</code> : AND operation
<code>is_not</code>	Boolean	<code>tag_rules</code>	Yes	Whether to perform "NOT" operation on the calculation result of the <code>tag_items</code> array. <code>true</code> : Yes <code>false</code> : No

tag_items description

Parameter	Type	Parent Project	Required	Description
tags	Array	tag_items	Yes	Specific tag value in string type, such as <code>tag1</code> and <code>guangdong</code>
is_not	Boolean	tag_items	Yes	Whether to perform "NOT" operation on the calculation result of the <code>tags</code> array. Valid values: <code>true</code> : Yes <code>false</code> : No
tags_operator	String	tag_items	Yes	Operator for tag in <code>tags</code> . Valid values: <code>OR</code> : OR operation <code>AND</code> : 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 <code>tag_items</code> element is the operator between the first and second <code>tag_items</code> elements, and so on. Valid values: <code>OR</code> : OR operation <code>AND</code> : AND operation Note: For different rules, AND takes precedence over OR.
tag_type	String	tag_items	Yes	See tag_type value table .

tag_type value table

Tag Name	tag_type Value	Sample Tag
Custom tag	xg_user_define	tag1, tag2
Application version	xg_auto_version	1.1.0, 1.2.0.1
Device district information	xg_auto_province	guangdong, shanghai
Active information	xg_auto_active	20200131, 20200201
XG SDK version	xg_auto_sdkversion	1.1.5.2, 1.1.5.3

System language	xg_auto_systemlanguage	zh, en
Mobile phone brand	xg_auto_devicebrand	Mi, vivo
Mobile phone model	xg_auto_deviceversion	MI 9 SE, vivo X9Plus
Country/Region	xg_auto_country	CN, SG

Note:

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

channel_rules parameter description

Parameter	Type	Parent Project	Required	Description
channel	String	channel_rules	Yes	Delivery push channel. Valid values: xg: Tencent Push Notification Service channel hw: Huawei channel xm: Mi channel mz: Meizu channel vivo: vivo channel oppo: OPPO channel apns: APNs channel honor: HONOR channel fcm: FCM channel
disable	Boolean	channel_rules	Yes	Whether to disable the channel specified in channel , which is enabled by default. Valid values: true: Disable false: Enable

loop_param parameter description

Parameter	Type	Parent Project	Required	Description
startDate	String	loop_param	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	loop_param	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	loop_param	Yes	Loop type. Valid values: 1 : Daily 2 : Weekly 3 : Monthly
loopDayIndexs	Array	loop_param	Yes	Daily loop value: [0], indicating the push will be done every day. Weekly loop value: [0-6]; for example, [0, 1, 2] indicates the push will be done on every Sunday, Monday, and Tuesday. Monthly loop value: dates; for example, [1, 10, 20] indicates the push will be done on the 1st, 10th, and 20th days of each month.
dayTimes	Array	loop_param	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

Parameter	Type	Description
seq	Integer	Same as the request (if the request does not contain this parameter, this parameter returns 0).
push_id	String	Push ID Note: If you use the loop push type, multiple pushid values will be returned and placed in an array.
invalid_targe_list	Array	This parameter is returned only when the push target is a token list or a single token and the value of ignore_invalid_token is 1 . This parameter stores filtered invalid tokens and delivers pushes to devices with valid tokens properly.
ret_code	Integer	Error code. For more information, see the error codes table.
environment	String	Push environment specified by the user (only for iOS). Valid values: product : production environment dev : development environment
err_msg	String	Error message when a request error occurs
result	String	When the request is correct: If there is extra data to be returned, the result will be encapsulated in this parameter in JSON format.

		If there is no extra data, this parameter may not exist.
--	--	--

Examples

Android request message for push by account

```
{
  "audience_type": "account",
  "account_list": [
    "account1"
  ],
  "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
    "xg_media_audio_resources": "xxx", // Enter the URL of audio rich media elements
    "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 applic
        "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": "xxxx ", // Only HTTP and HTTPS URLs are sup
            "confirm": 1 // Whether user's confirmation is requ
        },
        "intent": "xxx" // The SDK must be version 1.0.9 or later.
    },
    "custom_content": "{\\"key\\":\\"value\\"}"
}
}
}

```

Response message for push by account

```

{
    "seq": 0,
    "environment": "product",
    "ret_code": 0,

```

```
"push_id": "3895624686"
}
```

iOS request message for push to a single device

```
{
  "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\\"}"
    }
  }
}
```

Response message for push to a single device

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

Push by tag (using tag_rules)

Sample 1: Push messages 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 the "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 messages to Huawei users who were active in the last 3 days on an 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 the "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_verison"
        },
        {
          "tags": [
            "huawei"
          ],
          "is_not": false,
          "tags_operator": "OR",
          "items_operator": "AND",
          "tag_type": "xg_auto_devicebrand"
        }
      ],
    }
  ],
}
```

```
        "operator": "OR",
        "is_not": false
    }
]
```

Creating a Push Plan

Last updated : 2024-01-17 14:24:51

API Description

Request method: POST

Request address:

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

API service URLs correspond to service access points one to one. Please select the [service URL](#) corresponding to the service access point of your application.

Note:

Call rate limit: 200 calls/day.

Request Parameters

Parameter	Type	Required	Description
planName	string	Yes	Push plan name, which can contain up to 60 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

Examples

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
  }
}
```


Account Package Upload API

Last updated : 2024-11-20 16:59:51

API Description

Request method: POST

```
Service URL/v3/push/package/upload
```

API service URLs correspond to service access points one to one. Please select the [service URL](#) corresponding to the service access point of your application.

Feature: you can upload a package of multiple accounts and push to the accounts in batches. The two main APIs used are the account package upload API and the account package push API.

Note:

Account package file name: [1, 100] characters.

Account package format and size: `.zip` , `.txt` , or `.csv` file within 100 MB.

`.zip` file requirements: can contain single `.txt` or `.csv` files but not folders.

`.txt` file requirements: encoded in UTF-8; one account ([2, 100] characters) per row.

`.csv` file requirements: one column only; one account ([2, 100] characters) per row.

Request Parameters

Parameter	Type	Required	Description
file	form-data	Yes	Account package format and size: <code>.zip</code> , <code>.txt</code> , or <code>.csv</code> file within 100 MB <code>.zip</code> file requirements: can contain single <code>.txt</code> or <code>.csv</code> files but not folders <code>.txt</code> file requirements: encoded in UTF-8; one account ([2, 100] characters) per row <code>.csv</code> file requirements: one column only; one account ([2, 100] characters) per row

Response Parameters

Parameter	Type	Required	Description
-----------	------	----------	-------------

retCode	Integer	Yes	Error code
errMsg	String	Yes	Error message of a request error
uploadId	Integer	Yes	When a file is uploaded, a positive integer is returned, which represents the ID of the uploaded file (<code>uploadId</code>). It is provided to the push API for account package push.

Sample Request

Python3

Golang

C#

PHP

JAVA

C++

```
import base64
from pip._vendor import requests
from pip._vendor.urllib3 import encode_multipart_formdata

def upload(url, filePath, accessId, secret, data={}, header={}):
    openFile = open(filePath, 'rb')
    data['file'] = (openFile.name, openFile.read())
    encode_data = encode_multipart_formdata(data)
    data = encode_data[0]
    header['Content-Type'] = encode_data[1]
    authInfo = accessId + ":" + secret

    header['Authorization'] = "Basic " + str(base64.b64encode(bytes(authInfo,
encoding="utf8")),encoding="utf8")

    r = requests.post(url, headers=header, data=data)
    print(r.json())

func Upload(url string, filePath string, accessId string, secret string)(resp
string , err error) {
    fp, err := os.Open(filePath)
    if err != nil {
        return resp, err
    }
    defer fp.Close()
```

```
body := &bytes.Buffer{}
writer := multipart.NewWriter(body)
defer writer.Close()
part, err := writer.CreateFormFile("file", filepath.Base(fp.Name()))
if err != nil {
    return resp, err
}
io.Copy(part, fp)
writer.Close()
httpReq, err := http.NewRequest(http.MethodPost, url, body)
if err != nil {
    return resp, err
}

httpReq.Header.Add("Content-Type", writer.FormDataContentType())

authInfo := base64.StdEncoding.EncodeToString([]byte(accessId + ":" +
secret))
httpReq.Header.Add("Authorization", fmt.Sprintf("Basic %s", authInfo))
cli := &http.Client{}
httpResp, err := cli.Do(httpReq)
if err != nil {
    return resp, err
}
defer httpResp.Body.Close()
data, err := ioutil.ReadAll(httpResp.Body)
if err != nil {
    return resp, err
}
if httpResp.StatusCode != http.StatusOK {
    return resp, fmt.Errorf("response error, status: %s, body: %s",
httpResp.Status, string(data))
}

resp = string(data)
return resp, nil
}
```

```
public string Upload(string url, string filePath, uint accessId, string
secretKey)
{
    var tmp = accessId.ToString() + ":" + secretKey;
```

```

        var sign =
System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetBytes(tmp));
        MultipartFormDataContent form = new MultipartFormDataContent();
        var content = new StreamContent(new FileStream(filePath,
 FileMode.Open));
        content.Headers.ContentDisposition = new
ContentDispositionHeaderValue("form-data")
        {
            Name = "file",
            FileName = Path.GetFileName(filePath)
        };
        form.Add(content);
        using(HttpClient client = new HttpClient())
        {
            client.DefaultRequestHeaders.Add("Authorization","Basic " +
sign);

            client.DefaultRequestHeaders.ExpectContinue = true;
            var response = (client.PostAsync(url, form)).Result;
            var result = response.Content.ReadAsStringAsync().Result;
            if (response.StatusCode != HttpStatusCode.OK) {
                throw new Exception(result);
            }
            return result;
        }
    }

function upload($url, $file, $accessId, $secretkey) {
    $sign = base64_encode($accessId . ":" . $secretKey);
    $headers = array("Content-type: multipart/form-data", "Authorization: Basic
" . $sign);
    $cfile = new \\CURLFile($file,'multipart/form-data',basename($file));
    $options = array(
        CURLOPT_HTTPHEADER => $headers,
        CURLOPT_HEADER => 0,
        CURLOPT_SSL_VERIFYPEER => false,
        CURLOPT_SSL_VERIFYHOST => 0,
        CURLOPT_POST => 1,
        CURLOPT_URL => $url,
        CURLOPT_RETURNTRANSFER => 1,
        CURLOPT_TIMEOUT => 10000,
        CURLOPT_POSTFIELDS => array("file" => $cfile)
    );
    $ch = curl_init();
    curl_setopt_array($ch, $options);
    $ret = curl_exec($ch);

```

```
$error = curl_error($ch);
$info = curl_getinfo($ch);
curl_close($ch);
if ($error != "") {
    throw new \Exception($error);
}
$code = $info["http_code"];
if ($code != 200) {
    throw new \Exception("status: " . $code . ", message: " . $ret);
}
return $ret;
}
...

public JSONObject Upload(String url, String filePath, String accessId, String
secret) {
    OkHttpClient client = new OkHttpClient();
    File file = new File(filePath);
    RequestBody requestBody = new MultipartBody.Builder()
        .setType(MultipartBody.FORM)
        .addFormDataPart("file", file.getName(),
            RequestBody.create(MediaType.parse("multipart/form-
data"), file))
        .build();

    String authString = accessId + ":" + secret;
    byte[] authEncBytes = Base64.encodeBase64(authString.getBytes());
    String authStringEnc = new String(authEncBytes);

    Request.Builder builder = new Request.Builder()
        .url(url)
        .post(requestBody)
        .addHeader("Authorization", "Basic " + authStringEnc);

    Request request = builder.build();
    JSONObject jsonRet = null;
    Response response = null;
    try {
        response = client.newCall(request).execute();
        if(response.code() == 200){
            String retMsg = response.body().string();
            jsonRet = new JSONObject(retMsg);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    return jsonRet;
}
```

```
    }
    else{
        jsonRet = new JSONObject();
        jsonRet.put("retCode", 10101);
        jsonRet.put("errMsg", "CallApiError,HttpStatus Code:" +
response.code());
    }
} catch (IOException e) {
    jsonRet = new JSONObject();
    jsonRet.put("retCode", 10100);
    jsonRet.put("errMsg", e.getMessage());
}finally {
    if (response != null) {
        response.close();
    }
}
return jsonRet;
}
```

```
struct memory {
    char *data;
    size_t size;
    memory(): data(NULL), size(0) {}
    ~memory() {
        if (data) {
            free(data);
        }
    }
};
```

```
static size_t callback(void *data, size_t size, size_t nmemb, void *userp)
{
    size_t realsize = size * nmemb;
    struct memory *mem = (struct memory *)userp;

    char *ptr = (char *)realloc(mem->data, mem->size + realsize + 1);
    if(ptr == NULL)
        return 0;

    mem->data = ptr;
    memcpy(&(mem->data[mem->size]), data, realsize);
    mem->size += realsize;
    mem->data[mem->size] = 0;
    return realsize;
}
```

```
std::string upload(const std::string url, const std::string &file, uint32_t
accessId, const std::string &secretKey, std::string &err) {
    CURL *pcurl = curl_easy_init();
    if (pcurl == NULL) {
        err.assign("curl_easy_init error");
        return "";
    }
    struct curl_httppost *post = NULL;
    struct curl_httppost *last = NULL;
    std::string base;
    //for unix/linux
    size_t pos = file.find_last_of("/");
    if (pos >= 0) {
        base = file.substr(pos + 1);
    }
    //for windows
    pos = file.find_last_of("\\");
    if (pos >= 0) {
        return base = file.substr(pos + 1);
    }
    curl_formadd(&post, &last, CURLFORM_PTRNAME, "file", CURLFORM_FILE,
file.c_str(), CURLFORM_FILENAME, base.c_str(), CURLFORM_END);

    char buf[128];
    char dst[128];
    char authInfo[256];
    snprintf(buf, sizeof(buf), "%u:%s", accessId, secretKey.c_str());
    Base64encode(dst, buf, strlen(buf));
    snprintf(authInfo, sizeof(authInfo), "Authorization: Basic %s", dst);

    curl_slist *plist = NULL;
    plist = curl_slist_append(plist, "Content-Type: multipart/form-data");
    plist = curl_slist_append(plist, authInfo);

    curl_easy_setopt(pcurl, CURLOPT_CONNECTTIMEOUT, 10);
    curl_easy_setopt(pcurl, CURLOPT_TIMEOUT, 10);
    curl_easy_setopt(pcurl, CURLOPT_POST, true);

    curl_easy_setopt(pcurl, CURLOPT_HTTPHEADER, plist);
    curl_easy_setopt(pcurl, CURLOPT_URL, url.c_str());
```

```
curl_easy_setopt(pcurl, CURLOPT_HTTPPOST, post);
curl_easy_setopt(pcurl, CURLOPT_WRITEFUNCTION, callback);
struct memory chunk;
curl_easy_setopt(pcurl, CURLOPT_WRITEDATA, (void *)&chunk);
curl_easy_setopt(pcurl, CURLOPT_NOSIGNAL, 1);

CURLcode res = curl_easy_perform(pcurl);
curl_slist_free_all(plist);

// Check for errors
if (res != CURLE_OK)
{
    char buf[1024];
    snprintf(buf, sizeof(buf), "curl_easy_perform error: %s",
curl_easy_strerror(res));
    err.assign(buf);
    curl_easy_cleanup(pcurl);
    return "";
}
long status = 0;
curl_easy_getinfo(pcurl, CURLINFO_RESPONSE_CODE, &status);
curl_easy_cleanup(pcurl);
if (status != 200) {
    char buf[1024];
    snprintf(buf, sizeof(buf), "%ld %s", status, chunk.data);
    err.assign(buf);
    return "";
}

std::string ret;
ret.assign(chunk.data);
return ret;
}
...
```


Token Package Upload API

Last updated : 2024-01-17 14:24:51

API Description

Request method: POST

Service URL/v3/push/package/upload

API service URLs correspond to service access points one by one. Please select the [service URL](#) corresponding to the service access point of your application.

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	Type	Required	Description
file	form-data	Yes	Token package file name: [1, 100] characters Token package format and size: <code>.zip</code> , <code>.txt</code> , or <code>.csv</code> file within 100 MB <code>.zip</code> file requirements: contains a single <code>.txt</code> or <code>.csv</code> file but not folders <code>.txt</code> file requirements: encoded in UTF-8; one token (36 characters) per row <code>.csv</code> file requirements: one column only; one token (36 characters) per row

Response Parameters

Parameter	Type	Required	Description
retCode	Integer	Yes	Error code
errMsg	String	Yes	Error message when a request error occurs
uploadId	Integer	Yes	When a file is uploaded, <code>uploadId</code> is provided as a positive integer, which represents the ID of the uploaded file. It is provided to the push

		API for token package push.
--	--	-----------------------------

Sample Request

Curl sample

```
curl -X POST
https://api.tpns.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.tpns.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=f
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 : 2024-01-17 14:24:51

API Calling Description

Request method: POST.

```
Service URL/v3/device/tag
```

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 features:

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

For use cases of tags, see [Tag Feature](#).

Parameter description

Request parameters

Parameter	Type	Required	Description
operator_type	Integer	Yes	Operation type: 1: add a tag to a token. 2: delete a tag from a token. 3: add multiple tags to a token. 4: delete multiple tags from a token. 5: delete all tags from a token. 6: add one or more tags or custom tags to a token in an overwriting manner. (This API starts to set new tags only after the tag history is cleared. Therefore, for the same token, it is recommended that the interval between API calls be over 1s; otherwise, the update may fail.) 7: add a tag to multiple tokens. 8: delete a 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).
token_list	Array	No	Device list: Required when operator_type is 1, 2, 3, 4, 5, 6, 7, or 8. When operator_type is 1, 2, 3, 4, 5, or 6 and this parameter contains multiple tokens, only the first token will be set. Format example: ["token1","token2"] The list can contain up to 500 values. A token string cannot exceed 36 characters.
tag_list	Array	No	Tag list: Required when operator_type is 1, 2, 3, 4, 6, 7, or 8; ignored when operator_type is 5. When operator_type is 1, 2, 7, or 8 and this parameter contains multiple tags, only the first tag will be set. Format example: ["tag1","tag2"] The list can contain up to 500 values. A tag string cannot exceed 50 characters.
tag_token_list	Array	No	Tag-device pair list: Required when operator_type is 9 or 10. Format example: [{"tag":"tag123", "token":"token123"}] In each pair, the tag is before the token. The list can contain up to 500 values. A tag string cannot exceed 50 characters. A token string cannot exceed 36 characters.

One application can have up to 10,000 custom tags. One device token can be bound to up to 100 custom tags. If you want to increase this limit, contact [our online customer service](#). One custom tag can be bound to an unlimited number of device tokens.

If the API needs to be called only once, then there are no restrictions on the API call method.

If the API needs to be called continuously, note the following:

If the API needs to be called continuously to set more than 10 tags or tokens, you are advised to use batch APIs. It is recommended that the interval between API calls be no less than 1s to guarantee correct tag operations.

When not using batch APIs, then the API for the next non-batched tag is called after the returned value of the Tencent Push Notification Service 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 Tencent Push Notification Service backend for tag categorization. If you include `:` in a tag when setting it, the string before the first `:` serves as the category name of the tag. This is only used when operator_type is 6, which means overwriting tags by category. For example, if the existing tags of a device are `level:1`, `level:2`, and `male`, you can directly overwrite `level:1` and `level:2` with

`level:3` without having to unbind `level:1` and `level:2` one by one, and then bind `level:3` . For more information, please see the following sample request where `operator_type` is `6` .

Response parameters

Parameter	Type	Required	Description
<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 a request error occurs
<code>result</code>	String	No	When the request is correct: If there is extra data to return, the result will be encapsulated in this parameter in JSON format. If there is no extra data, this parameter may not exist.

Samples

Examples of Tag Binding and Unbinding Requests

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

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

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

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

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

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

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

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

Delete all tags from a token (token1)

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

Overwrite custom tags of a token (token1)

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

```
]
}
```

Note:

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

Batch overwrite custom tags of a token (this API overwrites custom tags only, not all tags)

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

Note:

If all the tags contain `:`, since the string before the first `:` is the tag category, only tags in the same category bound to the device will be overwritten; for example, `test:2` overwrites `test:*`, and `level:2` overwrites `level:*`, without affecting other tags.

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

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

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

```
{
  "operator_type": 8,
  "tag_list": [
    "tag1"
  ],
  "token_list": [
    "token1",
    "token2"
  ]
}
```

```
]
}
```

Batch set tags

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

Batch delete tags (tag1, tag2, and tag3)

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

Example of Tag Setting Request

```
POST /v3/device/tag HTTP/1.1
Host: api.tpns.tencent.com
Content-Type: application/json
Authorization: Basic YTViNWYwNzFmZjc3YTplYTUxMmViNzZcwNGQ1ZmI1YTZhOTM3Y2FmYTcwZTc3MQ
Cache-Control: no-cache
Postman-Token: 4b82a159-afdd-4f5c-b459-de978d845d2f
{
  "operator_type": 1,
  "tag_list": [
```



```
        "tag1"  
    ],  
    "token_list": [  
        "token1"  
    ]  
}
```

Example of Tag Setting Response

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

Deleting All Devices Under Tag

Last updated : 2024-01-17 14:24:51

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: 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 : 2024-01-17 14:26:44

API Calling Description

Request method: POST.

```
Service URL/v3/device/account/batchoperate
```

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

Parameter description

Request parameters

Parameter	Type	Required	Description
operator_type	Integer	Yes	Operation type. Value range: [1, 5] (Disused) <code>1</code> : Bind a token to one more account. <code>2</code> : Bind a token to another account in an overriding manner. <code>3</code> : Unbind a token from multiple accounts. <code>4</code> : Unbind a token from all accounts. <code>5</code> : Unbind an account from all tokens.
account_list	Array	No	List of account IDs. This field is valid and required when <code>operator_type</code> is <code>5</code> . Each element contains the <code>account</code> and <code>account_type</code> (for type setting) fields. Example: <pre>[{"account": "926@126.com"}, {"account": "1527000000"}, {"account": "2849249569", "account_type": 9}]</pre>
token_list	Array	No	List of device IDs. This field is valid and required when <code>operator_type</code> is <code>4</code> .
token_accounts	Array	No	

			This field is valid and required when <code>operator_type</code> is <code>1</code> , <code>2</code> , or <code>3</code> . Up to 20 tokens can be set in each call. Each <code>token_account</code> consists of 1 token and 1 <code>account_list</code> . Example: <pre>[{"token": "token1", "account_list": [{"account": "926@126.com"}, {"account": "1527000000"}]}, {"token": "token2", "account_list": [{"account": "926@163.com"}, {"account": "1527000001"}]}]</pre>
account_type Integer	No	Account type, which can be selected during account binding. The default value is <code>0</code> . For valid values, see Account Type Value Table .	-

Note:

As the `token append account` API was seldom used and confusing to developers, it had been disused since October 26, 2020. If you have used it before, it will be replaced by the `Token clearAndAppendAccount` API. Only one account can be bound to an account type. If multiple accounts are bound to an account type, an account bound later will override the one bound earlier.

Response parameters

Parameter	Type	Description
ret_code	Integer	Error code. For more information, see Server-Side Error Codes .
err_msg	String	Error message.
result	Array	Operation status code corresponding to each token or account ["0", "1008006"].

Samples

Sample request

(Disused) Bind a token to one more account

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

Bind the token to another account in an overriding manner

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

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

Unbind a token from an account

```
{
  "operator_type": 3,
  "token_accounts": [
    {
      "token": "token1",
      "account_list": [
        {
          "account": "926@126.com"
        },
        {
          "account": "1527000000"
        }
      ]
    },
    {
      "token": "token2",
      "account_list": [
        {
          "account": "926@163.com"
        },
        {
          "account": "1527000001"
        }
      ]
    }
  ]
}
```

Unbind the token from all accounts

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

Unbind an account from all tokens

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

Response example

Unbind a token from an account

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


Account and Device Binding Query

Last updated : 2024-01-17 14:26:44

API Description

Request method: POST

```
Service URL/v3/device/account/query
```

API service URLs correspond to service access points one to one. Please select the [service URL](#) corresponding to the service access point of your application.

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

Parameters

Request parameters

Parameter	Type	Required	Description
operator_type	Integer	Yes	Operation type: 1 : Batch query tokens by account 2 : Query accounts by token
account_list	Array	No	Account list. This parameter is valid and required when <code>operator_type</code> is 1 . Each element of the list contains one set of accounts. Example: [{"account": "account1"}, {"account": "account2"}]
token_list	Array	No	Token list. This parameter is valid and required when <code>operator_type</code> is 2 .

Response parameters

Parameter	Type	Description
retCode	Integer	Error code. For more information, please see Server-Side Error Codes .
errMsg	String	Error message
account_tokens	Array	Array of mappings from accounts to tokens.

		Example: [{"account": "account1", "token_list": ["token1", "token2"]}, {"account": "account2", "token_list": ["token2", "token3"]}]
token_accounts	Array	Array of mappings from tokens to accounts Example: [{"token": "token1", "account_list": [{"account": "926@126.com"}, {"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 response

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 : 2024-11-19 11:31:42

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 Format: YYYYMMDD Limit: only data in the last 6 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
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.

channelDatas

Parameter Name	Type	Description
channel	String	Push channel name xg: Tencent Push Notification Service 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 apns: Apple channel iospk: Apple PushKit channel (available only in International Site) honor: Honor channel
pushState	Object	Push funnel statistics. For the data structures, please see pushState 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 Tencent Push Notification Service channel.
arrivalUv	Integer	Number of reached devices (including arrival receipts for the Tencent Push Notification Service 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 Tencent Push Notification Service, ROG, and FCM channels. For other vendor channels, the pushOnlineUv metric of actual deliveries by Tencent Push Notification Service will be used). Note: this field will be disused subsequently. You are recommended to check the arrivalUv 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 arrivalUv 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 Tencent Push Notification Service, ROG, and FCM channels.

In the aggregated statistics, `pushActiveUv` (attempted) and `pushOnlineUv` (sent) aggregates the data of the Tencent Push Notification Service 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) + Tencent Push Notification Service channel's `verifySvcUv` (reached devices) + ROG channel's `verifySvcUv` (reached devices) + FCM channel's `verifySvcUv` (device reached).

pushState (iOS)

Parameter Name	Type	Description
pushActiveUv	Integer	Attempted
pushOnlineUv	Integer	Successfully received by APNs
verifySvcUv	Integer	Reached
clickUv	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 : 2024-01-17 14:26:44

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 active users", 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 Format: YYYYMMDD Limit: 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 : 2024-01-17 14:26:44

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, 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: notify: notification message: silent message
pushType	No	String	Push type: 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	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	all //Full push tag //Tag push token_list //Device list account_list //Account list package_account_push //Number package push

messageType	String	Push type	notification //Notification message //Message
environment	String	Push environment	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	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. op
    "intent": "" // The SDK version must be 1.0.9 or higher. Configure the
  },
  "custom_content": "{}"
}
```

iOS push configuration information

```
"ios":{
  "aps": {
    "alert": {
      "subtitle": "my subtitle"
    },
    "badge_type": 5, // Badge number displayed by application (optional). -
    "category": "INVITE_CATEGORY",
    "sound": "default", // If this parameter is left empty, the default soun
    "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 : 2024-01-17 14:26:44

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	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	all // Full push tag // Tag push token_list // Device list account_list // Account list package_account_push // Number package push
messageType	String	Push type	notification //Notification message //Message
environment	String	Push environment	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	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. op
    "intent": "" // The SDK version must be 1.0.9 or higher. Configure the
  },
  "custom_content": "{}"
}
...

#### iOS push configuration information

```json
"ios": {
 "aps": {
 "alert": {
 "subtitle": "my subtitle"
 },
 "badge_type": 5, // Badge number displayed by application (optional). -

```

```
 "category": "INVITE_CATEGORY",
 "sound": "default", // If this parameter is left empty, the default sound
 "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 for a Single Task

Last updated : 2024-01-17 14:26:44

## API Description

**Request method:** POST

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

```
Service URL/v3/statistics/get_push_task_stat_channel
```

API service URLs correspond to service access points one by one. Please select the [service URL](#) 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 the summary. The channel types in `pushStatDataAll` vary depending on the OS (iOS/Android) and push channels.

## Parameters

### Request parameters

Parameter	Required	Type	Description
pushId	Yes	String	ID of a push task. You can only query push tasks within last 30 days.

### Response parameters

Parameter	Type	Description
retCode	Integer	Returned status code
errMsg	String	Error message
pushStatDataAll	Array	Variables in the <code>pushStatDataAll</code> structure. See the table below for details.

### pushStatDataAll

Parameter	Type	Description
channel	String	Name of a push channel



		xg: TPNS hw: Huawei xm: Mi mz: Meizu oppo: OPPO vivo: vivo apns: APNs fcm: FCM rog: ROG apns: Apple channel iospk: Apple PushKit channel (available only in International Site) honor: Honor channel all: all channels
pushState	Object	Variables in the pushState structure. See the table below for details.

**pushState (Android)**

Parameter	Type	Description
pushActiveUv	Integer	Scheduled delivery count Number of available devices online in last 90 days with the notification bar enabled in the push target devices
pushOnlineUv	Integer	Actual delivery count Number of available devices to which the message was successfully delivered through the vendor or TPNS channel out of the devices for scheduled delivery
arrivalUv	Integer	Number of reached devices (including arrival receipts for the TPNS and vendor channels. For Huawei and Meizu channels, you need to configure the arrival receipt manually. For more information, please see <a href="#">Acquisition of Vendor Channel Arrival Receipt</a> ).
verifySvcUv	Integer	Number of reached devices (only for TPNS, ROG, and FCM channels. The arrival data of other vendor channels is displayed using the <code>pushOnlineUv</code> parameter of TPNS). <b>Note:</b> This parameter will be discontinued later. Therefore, you are advised to use <code>arrivalUv</code> for the arrival data.
callbackVerifySvcUv	Integer	Arrival receipt for vendor channels (for Huawei and Meizu channels, you need to configure the arrival receipt manually. For more information, please see <a href="#">Acquisition of Vendor Channel Arrival Receipt</a> ). <b>Note:</b> This parameter will be discontinued later. Therefore, you are advised to use <code>arrivalUv</code> for the arrival data.

verifyUv	Integer	Displayed (this parameter has been discarded and will be discontinued later)
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 aggregate the data of the TPNS, ROG, and FCM channels.

In the aggregated statistics, `pushActiveUv` (scheduled delivery) and `pushOnlineUv` (actual delivery) aggregate 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` (reached devices).

**pushState (iOS and macOS)**

Parameter	Type	Description
pushActiveUv	Integer	Scheduled delivery
pushOnlineUv	Integer	Successfully received by APNs
verifySvcUv	Integer	Reached
clickUv	Integer	Clicked

## 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
 }
 },
 {
 "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 : 2024-01-17 14:26:44

## API Description

**Request method:** POST

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

```
Service URL/v3/statistics/get_push_group_stat_channel
```

API service URLs correspond to service access points one by one. Please select the [service URL](#) corresponding to the service access point of your application.

**Feature:** This API is used to query the aggregated statistics (by channel) of all push tasks that have the same `GroupID` or `planId` in last 7 days.

**Note:**

`GroupID` will be discarded gradually. Therefore, you are advised to use `planId` to query the aggregated statistics.

## Parameters

### Request parameters

Parameter	Required	Type	Description
planId	Yes	String	Parameter for aggregated statistics of multiple tasks, corresponding to <code>planId</code> in the push parameters
groupId	Yes	String	Parameter for aggregated statistics of multiple tasks, corresponding to <code>group_Id</code> in the push parameters <b>Note:</b> This parameter will be discarded gradually. Therefore, you are advised to use <code>planId</code> to query the aggregated statistics.
startDate	Yes	String	Query start date Format: YYYYMMDD Limit: only data of push tasks in last 7 days can be aggregated.
endDate	Yes	String	Query end date. Format: YYYYMMDD

**Note:**

If both `planId` and `groupId` are specified, the query will be based on `planId` by default.

**Response parameters**

Parameter	Type	Description
<code>retCode</code>	Integer	Returned status code
<code>errMsg</code>	String	Error message
<code>pushStatDataAll</code>	Array	Variables in the <code>pushStatDataAll</code> structure. See the table below for details.

**pushStatDataAll**

Parameter	Type	Description
<code>channel</code>	String	Name of a push channel xg: TPNS hw: Huawei xm: Mi mz: Meizu oppo: OPPO vivo: vivo apns: APNs fcm: FCM rog: ROG iospk: Apple PushKit channel (available only in International Site) apns: Apple channel honor: Honor channel all: all channels
<code>pushState</code>	Object	Variables in the <code>pushState</code> structure. See the table below for details.

**pushState (Android)**

Parameter	Type	Description
<code>pushActiveUv</code>	Integer	Scheduled delivery count Number of available devices online in last 90 days with the notification bar enabled in the push target devices
<code>pushOnlineUv</code>	Integer	Actual delivery count Number of available devices to which the message was successfully delivered through the vendor or TPNS channel out of the devices for

		scheduled delivery
arrivalUv	Integer	Number of reached devices (including arrival receipts for the TPNS and vendor channels. For Huawei and Meizu channels, you need to configure the arrival receipt manually. For more information, please see <a href="#">Acquisition of Vendor Channel Arrival Receipt</a> )
verifySvcUv	Integer	Number of reached devices (only for TPNS, ROG, and FCM channels. The arrival data of other vendor channels is displayed using the <code>pushOnlineUv</code> parameter of TPNS). <b>Note:</b> This parameter will be discontinued later. Therefore, you are advised to use <code>arrivalUv</code> for the arrival data.
callbackVerifySvcUv	Integer	Arrival receipt for vendor channels (for the Huawei and Meizu channels, you need to configure the arrival receipt manually. For more information, please see <a href="#">Acquisition of Vendor Channel Arrival Receipt</a> ). <b>Note:</b> This parameter will be discontinued later. Therefore, you are advised to use <code>arrivalUv</code> for the arrival data.
verifyUv	Integer	Displayed (this parameter has been discarded and will be discontinued later)
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 aggregate the data of the TPNS, ROG, and FCM channels.

In the aggregated statistics, `pushActiveUv` (scheduled delivery) and `pushOnlineUv` (actual delivery) aggregate 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` (reached devices).

**pushState (iOS)**

Parameter	Type	Description
pushActiveUv	Integer	Scheduled delivery
pushOnlineUv	Integer	Successfully received by APNs



verifySvcUv	Integer	Reached
clickUv	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
 }
 }
]
```

```
}
```

# Querying All Push Tasks on the Day by Token

Last updated : 2024-01-17 14:26:44

## API Calling Description

**Request method:** POST.

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

Service URL: v3/toolbox/getPushListByToken

API service URLs correspond to service access points one to one. Select the [service URL](#) corresponding to the service access point of your application.

**Feature:** Query the token-specific push data of the application on a day.

## Parameters

### Request parameters

Parameter	Required	Type	Description
token	Yes	String	Device token

### Response parameters

Parameter	Type	Description
retCode	Integer	Returned status code
errMsg	string	Error message
pushTaskList	pushTask	Push parameter

### pushTask

Parameter	Type	Description
pushId	Integer	Push task ID
pushTime	Integer	Push timestamp
pushTargetType	Array	Push type

# Samples

## Sample request

```
{
 "token": "05a305f6b71abb3a6b8c759fd1bc56b4bb44"
}
```

## Sample response

```
{
 "retCode": 0,
 "errMsg": "NO_ERROR",
 "pushTaskList": [
 {
 "pushId": 589840563,
 "pushTime": 1651662600,
 "pushTargetType": "TAG_PUSH"
 },
 {
 "pushId": 590166744,
 "pushTime": 1651741604,
 "pushTargetType": "TAG_PUSH"
 },
 {
 "pushId": 590178525,
 "pushTime": 1651742807,
 "pushTargetType": "TAG_PUSH"
 },
 {
 "pushId": 590179538,
 "pushTime": 1651742914,
 "pushTargetType": "TAG_PUSH"
 },
 {
 "pushId": 590179672,
 "pushTime": 1651742928,
 "pushTargetType": "TAG_PUSH"
 },
 {
 "pushId": 590235722,
 "pushTime": 1651750200,
 "pushTargetType": "TAG_PUSH"
 }
]
}
```

```
}
```

# User Attribute APIs

Last updated : 2024-01-17 14:26:44

## API Description

**Request method:** POST.

```
Service URL/v3/device/set_custom_attribute
```

API service URLs correspond to service access points one by one. Please select the [service URL](#) corresponding to the service access point of your application.

**Feature:** this API is used to manage personalized attributes at the token level, including creation, deletion, update, and query operations.

## Parameter Description

### Request parameters

Parameter	Required	Type	Description
cmd	Yes	Integer	Operation type: 1: adding an attribute. 2: updating an attribute. 3: deleting an attribute. 4: deleting all attributes. 5: querying attributes.
token	Yes	String	Unique ID assigned to each device by TPNS. <a href="#">Suggestions on getting TPNS token (Android)</a> <a href="#">Suggestions on getting TPNS token (iOS)</a>
attributeInfo	Yes (when cmd is 1, 2, or 3)	Map	Attribute details. See the description of the attributeMap parameter below.
attributeMap	Yes (when cmd is 1, 2, or 3)	Map	Attribute details: key indicates the attribute name and can contain up to 50 bytes. <b>Note:</b> you should have created an attribute in the <a href="#">TPNS console</a> > <b>Toolbox &gt; User Attribute Management</b> ; otherwise, this parameter will be filtered out, and invalidAttribute will be returned. value indicates the attribute value and can contain up to 50 bytes.



## Response parameters

Parameter	Returned	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	Yes (when <code>cmd</code> is <code>5</code> )	Map	Attribute details.
invalidAttribute	Yes (when the attribute is invalid)	Array	Details of the invalid attribute.

## Samples

### Adding an attribute

#### Sample request

Add three attributes to a token.

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

#### Sample response

```
{
 "retCode": 0,
 "errMsg": "success",
 "invalidAttribute": [
 "high" // The corresponding key value does not exist in the console
]
}
```

```
}
```

## Updating an attribute

### Sample request

Update the value of the `name` attribute to `workman` .

```
{
 "cmd": 2,
 "token": "04cac74a714f61bf089987a986363d88****",
 "attributeInfo": {
 "attributeMap": {
 "name": "workman"
 }
 }
}
```

### Sample response

```
{
 "retCode": 0,
 "errMsg": "success"
}
```

## Deleting an attribute

### Sample request

Delete the `workman` value of the `name` attribute.

```
{
 "cmd": 3,
 "token": "04cac74a714f61bf089987a986363d88****",
 "attributeInfo": {
 "attributeMap": {
 "name": "workman"
 }
 }
}
```

### Sample response

```
{
 "retCode": 0,
 "errMsg": "success"
}
```

## Deleting all attributes

### Sample request

Delete all attributes of a token.

```
{
 "cmd": 4,
 "token": "04cac74a714f61bf089987a986363d88****"
}
```

### Sample response

```
{
 "retCode": 0,
 "errMsg": "success"
}
```

## Querying attributes

### Sample request

Query the attribute details of a token.

```
{
 "cmd": 5,
 "token": "04cac74a714f61bf089987a986363d88****"
}
```

### Sample response

```
{
 "retCode": 0,
 "errMsg": "success",
}
```

```
"attributeInfo": {
 "attributeMap": {
 "nickname": "workman"
 }
}
```

# Server-Side Error Codes

Last updated : 2024-01-17 14:26:44

Error Code	Description	Feedback API
1008001	Parameter parsing error	General API
1008002	Parameter missing	General API
1008003	Authentication failed	General API
1008004	Failed to call internal service	General API
1008006	Invalid token	Tag operation and account operation APIs
1008007	Invalid parameter	General API
1008011	File upload failed	Certificate upload API and number package upload API
1008012	The uploaded file is empty	Certificate upload API and number package upload API
1008013	Failed to parse the certificate	Certificate upload API
1008015	The push task ID does not exist	Statistics query API
1008016	Incorrect date and time parameter format	Statistics query API
1008019	The content is considered non-compliant by the content security service	Push API
1008020	Failed to verify the certificate package name	Certificate upload API
1008021	Failed to verify the content and format of the .p12 certificate	Certificate upload API
1008022	Incorrect .p12 certificate password	Certificate upload API
1008023	iOS push cert was expired for the environment: PRODUCT	Push certificate verification error. Check whether the certificate is within its validity period.
1008025	Failed to create the application. The application already exists under the product	App creation API

1008026	Batch operation partially failed	Account binding and query API
1008027	Batch operation completely failed	Account binding and query API
1008028	Exceeded the frequency limit: Exceeded the call frequency limit of statistics APIs. Exceeded the push frequency limits of push to all devices, push to devices with specified tags, and push by account package. For related information, see "Note" in Push API.	Statistics API and push API
1008029	Invalid token	Single push API, account binding API, and tag binding API
1008030	The app is not paid	General error
1008031	Application resources are terminated	General error
10110008	The queried token or account does not exist	Account binding and query API
10010005	The push target does not exist	Push API
10010012	Invalid push time	Push API
10010018	Repeated push	Push APIs for full push and tag push
1008035	The application AccessID is incorrect, or the service access point of your application does not match the domain address entered. Refer to Service URL	General error

# Server SDKs

Last updated : 2024-01-17 14:26:44

In addition to RESTful APIs, TPNS also provides mainstream server API SDKs.

## Officially Maintained Versions

The officially maintained versions are released in **TGit - TPNS** as open-source tools.  
The source code and sample code for push are provided at the official URLs for your reference.

Project	URL
Java	<a href="#">Official URL</a>
Python	<a href="#">Official URL</a>
PHP	<a href="#">Official URL</a>
C#	<a href="#">Official URL</a>
C++	<a href="#">Official URL</a>
Golang	<a href="#">Official URL</a>

# API (Java)

Last updated : 2024-01-17 14:26:44

## SDK Description

This SDK is used to encapsulate the Tencent Push Notification Service server APIs and communicate with the Tencent Push Notification Service 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>io.github.tpnsPush</groupId>
 <artifactId>xinge</artifactId>
 <version>1.2.4.11</version>
</dependency>
```

### Note:

There are `groupId` changes since version 1.2.4.11.

## Usage

### XingeApp API description

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

Parameter	Type	Required	Default Value	Description
appld	String	Yes	Empty	Push target <code>AccessID</code> , which can be obtained on the <a href="#">Product Management</a> page in the console.
secretKey	String	Yes	Empty	Push key, which can be obtained on the <a href="#">Product Management &gt; Configuration Management &gt; Basic Configuration</a> page.



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	https://openapi.xg.qq.com/	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.

## Sample

```
XingeApp xingeApp = new XingeApp.Builder()
 .appId(appid)
 .secretKey(secretKey)
 .domainUrl("https://api.tpsns.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](#).

## Samples

### Note:

The `XingeAppSimple` class provides API examples related to message push, account binding and unbinding, etc.

## Push to a single Android device

```
public JSONObject pushTokenAndroid() {
 PushAppRequest pushAppRequest = new PushAppRequest();
 pushAppRequest.setAudience_type(AudienceType.token);
}
```

```
pushAppRequest.setMessage_type(MessageType.notify);
Message message = new Message();
message.setTitle("title");
message.setContent("content");
pushAppRequest.setMessage(message);
MessageAndroid messageAndroid = new MessageAndroid();
message.setAndroid(messageAndroid);
ArrayList<String> tokenList = new ArrayList();
tokenList.add("04cac74a714f61bf089*****63d880993");
pushAppRequest.setToken_list(tokenList);
return this.xingeApp.pushApp(pushAppRequest);
}
```

## Push to a single Android account

```
public JSONObject pushAccountAndroid() {
 PushAppRequest pushAppRequest = new PushAppRequest();
 pushAppRequest.setAudience_type(AudienceType.account);
 pushAppRequest.setPlatform(Platform.android);
 pushAppRequest.setMessage_type(MessageType.notify);
 pushAppRequest.setAccount_push_type(1);
 Message message = new Message();
 message.setTitle("title");
 message.setContent("content");
 MessageAndroid messageAndroid = new MessageAndroid();
 message.setAndroid(messageAndroid);
 pushAppRequest.setMessage(message);
 ArrayList<String> accountList = new ArrayList();
 accountList.add("123");
 pushAppRequest.setAccount_list(accountList);
 return this.xingeApp.pushApp(pushAppRequest);
}
```

## Push to Android devices with specified tags

```
public JSONObject pushTagAndroid() {
 PushAppRequest pushAppRequest = new PushAppRequest();
 pushAppRequest.setAudience_type(AudienceType.tag);
 pushAppRequest.setPlatform(Platform.android);
 pushAppRequest.setMessage_type(MessageType.notify);
 Message message = new Message();
 message.setTitle("title");
 message.setContent("content");
 MessageAndroid messageAndroid = new MessageAndroid();
 message.setAndroid(messageAndroid);
}
```

```
pushAppRequest.setMessage(message);
ArrayList<String> tagList = new ArrayList();
tagList.add("tag");
TagListObject tagListObject = new TagListObject();
tagListObject.setTags(tagList);
tagListObject.setOp(OpType.OR);
pushAppRequest.setTag_list(tagListObject);
return this.xingeApp.pushApp(pushAppRequest);
}
```

## Push to all Android devices

```
public JSONObject pushAllAndroid() {
 PushAppRequest pushAppRequest = new PushAppRequest();
 pushAppRequest.setAudience_type(AudienceType.all);
 pushAppRequest.setPlatform(Platform.android);
 pushAppRequest.setMessage_type(MessageType.notify);
 Message message = new Message();
 message.setTitle("title");
 message.setContent("content");
 MessageAndroid messageAndroid = new MessageAndroid();
 message.setAndroid(messageAndroid);
 pushAppRequest.setMessage(message);
 return this.xingeApp.pushApp(pushAppRequest);
}
```

## Push to a single iOS device

```
public JSONObject pushTokenIos(){
 PushAppRequest pushAppRequest = new PushAppRequest();
 pushAppRequest.setAudience_type(AudienceType.token);
 pushAppRequest.setEnvironment(Environment.valueOf("dev"));
 pushAppRequest.setMessage_type(MessageType.notify);
 Message message = new Message();
 message.setTitle("title");
 message.setContent("content");
 MessageIOS messageIOS = new MessageIOS();
 Alert alert = new Alert();
 Aps aps = new Aps();
 aps.setAlert(alert);
 messageIOS.setAps(aps);
 message.setIos(messageIOS);
 pushAppRequest.setMessage(message);
 ArrayList<String> tokenList = new ArrayList<String>();
 tokenList.add("0250df875c93c55*****536b54fc1c49f");
}
```

```
pushAppRequest.setToken_list(tokenList);
return this.xingApp.pushApp(pushAppRequest);
}
```

## Push to a single iOS account

```
public JSONObject pushAccountIos() {
 PushAppRequest pushAppRequest = new PushAppRequest();
 pushAppRequest.setAudience_type(AudienceType.account);
 pushAppRequest.setEnvironment(Environment.valueOf("dev"));
 pushAppRequest.setMessage_type(MessageType.notify);
 Message message = new Message();
 message.setTitle("Account push");
 message.setContent("content");
 MessageIOS messageIOS = new MessageIOS();
 Alert alert = new Alert();
 Aps aps = new Aps();
 aps.setAlert(alert);
 messageIOS.setAps(aps);
 message.setIos(messageIOS);
 pushAppRequest.setMessage(message);
 ArrayList<String> accountList = new ArrayList();
 accountList.add("1122");
 pushAppRequest.setAccount_list(accountList);
 return this.xingApp.pushApp(pushAppRequest);
}
```

## Push to iOS devices with specified tags

```
public JSONObject pushTagIos() {
 PushAppRequest pushAppRequest = new PushAppRequest();
 pushAppRequest.setAudience_type(AudienceType.tag);
 pushAppRequest.setEnvironment(Environment.valueOf("dev"));
 pushAppRequest.setMessage_type(MessageType.notify);
 Message message = new Message();
 message.setTitle("Tag push");
 message.setContent("content");
 MessageIOS messageIOS = new MessageIOS();
 Alert alert = new Alert();
 Aps aps = new Aps();
 aps.setAlert(alert);
 messageIOS.setAps(aps);
 message.setIos(messageIOS);
}
```

```
pushAppRequest.setMessage(message);
ArrayList<String> tagList = new ArrayList();
tagList.add("1122");
TagListObject tagListObject = new TagListObject();
tagListObject.setTags(tagList);
tagListObject.setOp(OpType.OR);
pushAppRequest.setMessage(message);
pushAppRequest.setTag_list(tagListObject);
return this.xingeApp.pushApp(pushAppRequest);
}
```

## Push to all iOS devices

```
public JSONObject pushAllIos() {
 PushAppRequest pushAppRequest = new PushAppRequest();
 pushAppRequest.setAudience_type(AudienceType.all);
 pushAppRequest.setEnvironment(Environment.valueOf("dev"));
 pushAppRequest.setMessage_type(MessageType.notify);
 Message message = new Message();
 message.setTitle("Full push");
 message.setContent("content");
 MessageIOS messageIOS = new MessageIOS();
 Alert alert = new Alert();
 Aps aps = new Aps();
 aps.setAlert(alert);
 messageIOS.setAps(aps);
 message.setIos(messageIOS);
 pushAppRequest.setMessage(message);
 return this.xingeApp.pushApp(pushAppRequest);
}
```

## Push response sample

```
{"result":{"{}", "environment":"","push_id":"1328245138690125824", "err_msg":"NO_ERROR"}
```

## Server Return Codes

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

## FAQs

### 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.

### What should I do if the API returns the error code 1008007 that indicates invalid parameters?

Refer to [Push API](#) and enter all the parameters correctly.

### Are there SDKs in other programming languages?

You can find more server SDKs in other programming languages on the [SDK Download](#) page.

### What should I do if the push API returns `Peer certificate cannot be authenticated with given CA certificates` ?

This is because the CA certificate has expired. Enter the certificate directory and run the `openssl` command to view the expiration time:

```
openssl x509 -in signed.crt -noout -dates
```

Replace `signed.crt` with the name of the certificate on your server.