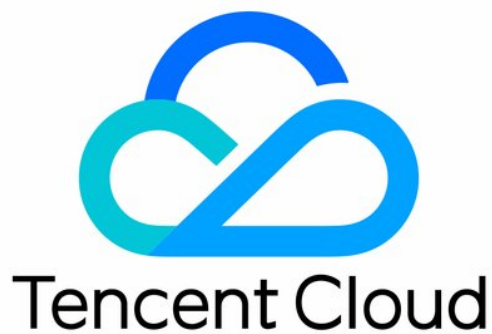


# **Tencent Push Notification Service**

## **Operation Guide**

### **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

## Operation Guide

### Push Management

Troubleshooting Tool

Conversion Funnel

Push Plan

Push Settings

Message Recall

### Advanced Push Features

Tags

Multi-Package Name Push

Notification Tap-to-Redirect

Custom Push Speed

Channel Policies

Message Collapse

Rich Media Notification

Custom Notification

Push to Accounts

# Operation Guide

## Push Management

## Troubleshooting Tool

Last updated : 2024-01-16 17:34:39

### Overview

Tencent Push Notification Service provides you with a troubleshooting tool for message push exceptions such as message delivery or push failures. You can use this tool to troubleshoot issues on your own or query device details (including the bound accounts and tags) via an account or token.

### Use Cases

1. A developer successfully pushes a message to the testing device via the console, but this message is not displayed in the notification bar of the device. Using a token to query the device details detects that the notification bar is disabled. Enable it on the device, trigger registration again, and send a message to the device again. The message is now displayed in the notification bar.
2. An operator fails to push a message to a specified account. Using the account and bound account type to query the token in the troubleshooting tool detects that the account hasn't been associated with a token yet. Please submit a ticket to check whether the API for [Android](#) or [iOS](#) was called to bind an account.

### Query device detail

Query Method ☒ Query by account ☐ Query through token

#### Step 1: Enter the user account type and account to search for Token

Account Type ⓘ

Default

Account

111

Query

The account does not belong to this application. Please confirm that the input is correct.

#### Step 2: Select token to view details

Token (displayed in the reverse order of the binding time with the account) ⓘ

Device Model

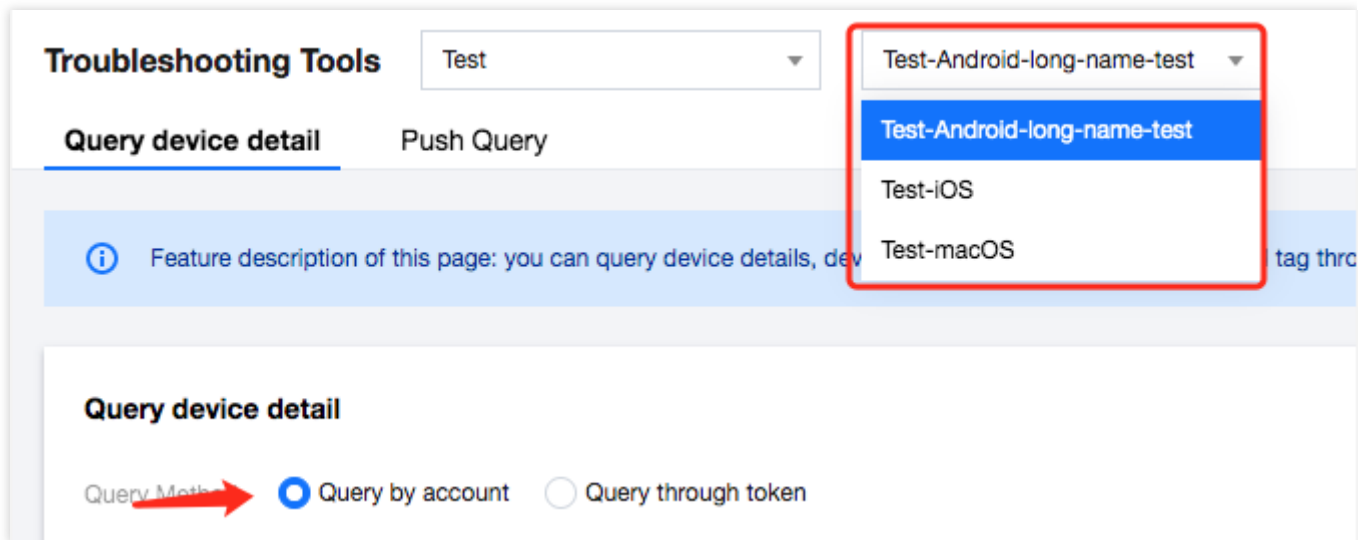
Please enter account to search

3. After completing the Tencent Push Notification Service integration, a developer pushes a message to a certain batch of device tokens, but an online user reports that the push message hasn't been reported. Querying the obtained token and `pushID` of the user in the troubleshooting tool detects that the token is not in the push list.

## Directions

### Querying via an account

1. Log in to the [Tencent Push Notification Service console](#).
2. In the left sidebar, choose **Message Management** > **Troubleshooting Tools**.
3. Select the product and application to be searched from the drop-down list, and select **Query by account** for the **Query Method**.



4. Enter the user account and click **Query**.
5. Select the token associated with the account to view device details.

**Note:**

Account: Refers to the unique ID of the user bound to the token, including OpenID and UID.

Token list: Displays tokens in reverse chronological order by the binding time. A maximum of 10 tokens are listed in a page. If the account is bound to multiple devices, the message will be pushed to the last device bound to this account.

To push the message to all devices bound to this account, change the push settings via the [console](#) or the [API](#).

## Querying via a token

1. Select **Query through token** for the **Query Method** on the **Troubleshooting Tools** page.
2. Enter a token and click **Query**. You will see the device details on the right.

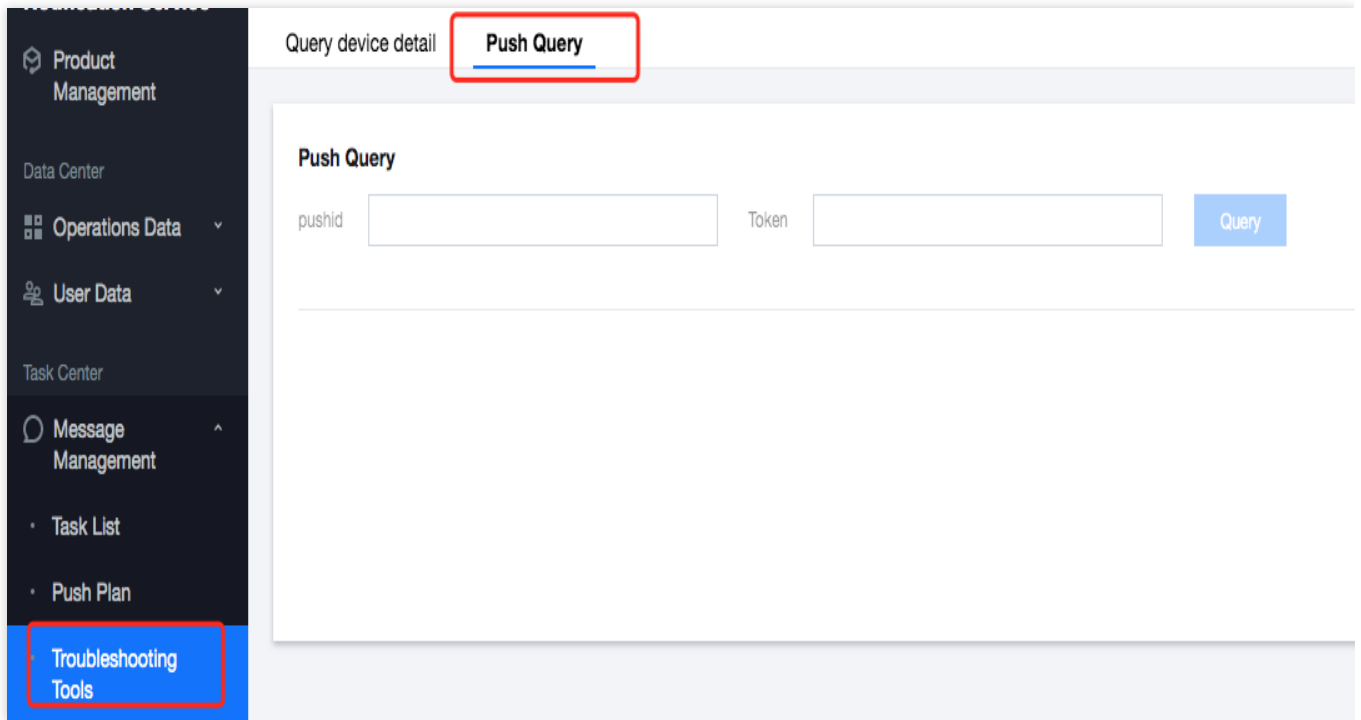
**Note:**

Token: refers to the unique 36-character ID assigned to each device by Tencent Push Notification Service.

If you want to obtain the device token, depending on your device type, please see the [Android](#) or [iOS](#) SDK documentation.

## Push query

1. On the **Troubleshooting Tools** page, click the **Push Query** tab.

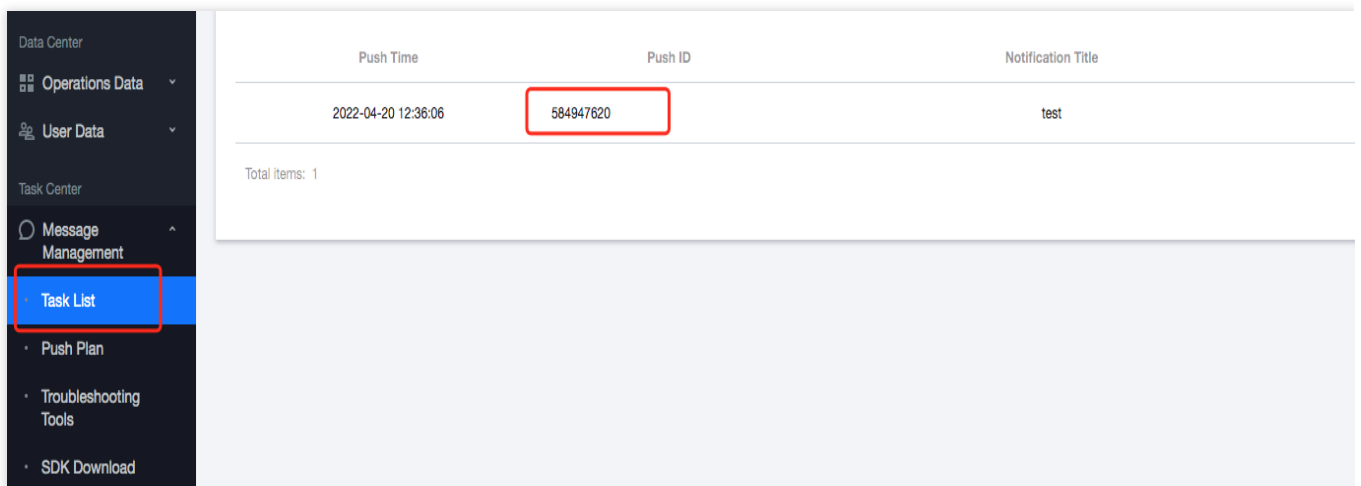


2. Enter the `pushid` (required) and the device token (required) to be queried, and click **Query** to view the troubleshooting results.

**Note:**

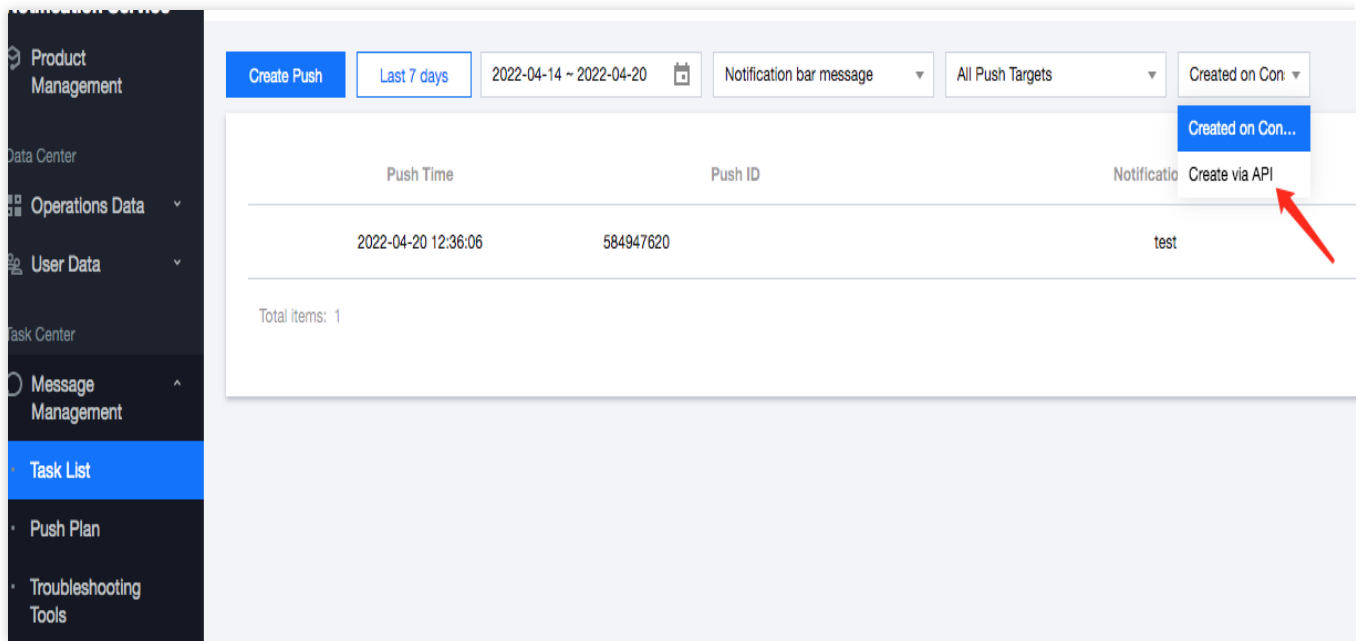
Obtain `pushid` as follows:

1. Choose **Message Management > Task List** in the left sidebar to obtain the `PushID` to be queried.



2. Obtain it from the response parameter of the push API.

3. Choose **Message Management > Task List** in the left sidebar, and switch from **Created on Console** to **Create via API**. Then you can view API push data.



4. If the query result does not match the actual situation, or the problem persists, you can view the [FAQs about push](#) or [contact our online customer service](#) with the `pushID` and token for assistance.

## FAQs

### Device query

1. In what situations will the device token expire?

The token is valid for 90 days. If the device hasn't been connected to the Tencent Push Notification Service server for 90 consecutive days, the device will be considered unavailable.

If the application has been uninstalled from the device, the token will be considered as an invalid token.

2. I have enabled the notification bar, but why is "notification bar status: disabled" displayed in the device details?

After the notification bar is enabled, send the registration request from the client again to sync the notification bar status to the Tencent Push Notification Service server.

### Push

1. Push by account or tag fails but push by token succeeds. Why does this happen?

On the **Query device detail** page, verify that the token is associated with the target account or tag to implement push by account or tag.

2. The push message is in the completed status and the device is normal. Why can't I receive the message on my mobile phone?

Check that you select the same environment for iOS and token. For more information, see [Push Environment Selection Description](#).



Check whether you entered the same app package name as in the [Tencent Push Notification Service console](#) > **Message Management** > **Basic Config**. If different, check whether the [Multi-Package Name Push](#) feature is enabled.

If you're using Android version P or later, add and use the Apache HTTP client library. To do this, add the following configuration to the AndroidManifest application node.

```
<uses-library android:name="org.apache.http.legacy" android:required="false"/>
```

If you're using a Mi phone, check whether the message is included in the ordinary notifications.

If you're using a Meizu phone, check whether the message is included in the message box.

If you're using a vivo phone, check whether the notification bar is manually enabled.

If you're using a vivo or Huawei phone, check whether the notification banners and sounds are manually enabled.

Do not contain "test" and other sensitive words in your notifications because such notifications may be blocked by vendor channels.

Note that the vivo and OPPO push channels must be approved by vendors before being used.

# Conversion Funnel

Last updated : 2024-01-16 17:34:39

You can go to the [Conversion Funnel](#) page to view and analyze the push conversion data of an application in different dimensions. We strongly recommend that you visit the conversion funnel page on a daily basis so that you can identify push conversion opportunities and stuck points in time and constantly optimize the push strategy to maximize the benefits of push.

## Metrics

**Planned Sends:** the number of non-unique devices available for the push tasks on the day

**Actual Sends:** the number of non-unique devices to which messages were pushed on the day

**Total Reaches:** the number of non-unique messages pushed on the day **with Tencent Push Notification Service or vendor channel arrival receipt**

**Total Clicks:** the number of non-unique clicks on pushed messages

**Send Rate (PV):** Actual sends/Planned sends x 100%

**Reach Rate (PV):** Total reaches/Actual sends x 100%

**Click Rate (PV):** Total clicks/Total reaches x 100%

**Planned Target Users:** the number of unique devices available for the push tasks on the day

**Actual Target Users:** the number of unique devices to which messages were pushed on the day

**Unique Reaches:** the number of unique devices that the pushed messages reached on the day

**Unique Clicks:** the number of unique devices on which messages were clicked on the day

**Send Rate (UV):** Actual target users/Planned target users x 100%

**Reach Rate (UV):** Unique reaches/Actual target users x 100%

**Click Rate (UV):** Unique clicks/Unique reaches x 100%

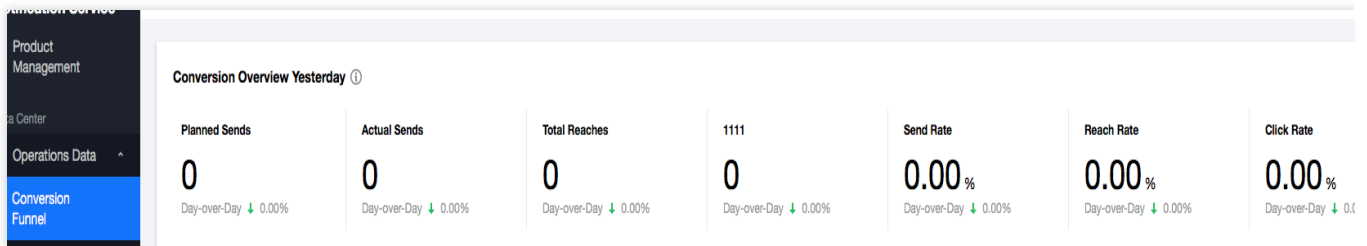
### Note:

Before scheduled delivery, filter out invalid devices or devices with the notification bar disabled.

Before actual delivery, filter out notifications that cannot be delivered via vendor channels, devices where the app is offline, and other conditions that do not support the delivery.

## Conversion Overview Yesterday

You can quickly view yesterday's overall conversion effect in the **Conversion Overview Yesterday** module and view the day-over-day fluctuations of metrics in the day-over-day comparison module.



## Statistical Dimension

**Filters**

Today Yesterday Last 7 days Last 30 days 2023-02-14 ~ 2023-02-14

Count (PV) Users (UV) All push plans All channels All messages

1. You can view conversion data by **a specified time range**, which can be the last 90 days at maximum.

### Note:

If you select a single day (for example, **Today**, **Yesterday**, or **2021-06-06 to 2021-06-06**), the data granularity for the statistics area at the bottom is hour. Otherwise, the data granularity is day.

2. You can choose to view conversion data by **Count (PV)** or **Users (UV)**. For **Count (PV)**, data is not deduplicated. For **Users (UV)**, data is deduplicated in the device dimension.

For example, if 10 messages were sent to the same device one day, and all the messages successfully reached the device and were clicked, the **Total Clicks** is 10 in the **Count (PV)** dimension and is 1 in the **Users (UV)** dimension. The same principle applies to other metrics.

3. You can view conversion data by **push plan**. We strongly recommend that you divide your business into different **push plans** by scenario and analyze conversion rates by scenario.

For example, if custom A's application has three push plans **Operational activities**, **User engagement**, and **Gifts for new users**, customer A can view the conversion effect of each or all of these push plans, facilitating constant growth strategy optimization.

4. You can view conversion data by **push channel**.

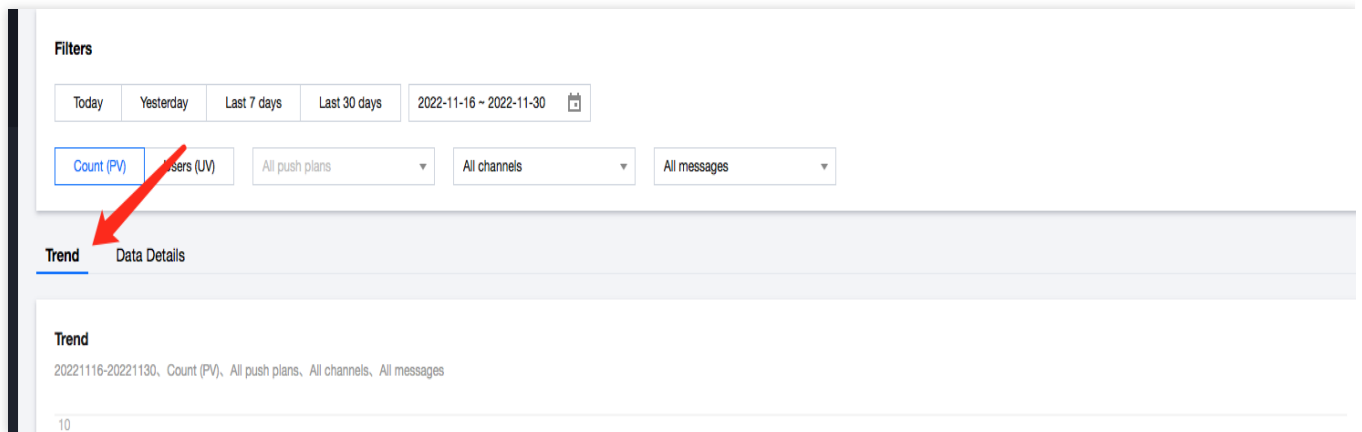
5. You can view conversion data by **message type**. If you use Tencent Push Notification Service to send notification bar messages, silent messages, and in-app messages, you can view the conversion effect of each or all of these message types.

# Data Analysis

In the data analysis area, you can view data trends and details, and download data. The statistics displayed in this area change dynamically with the filters you specify.

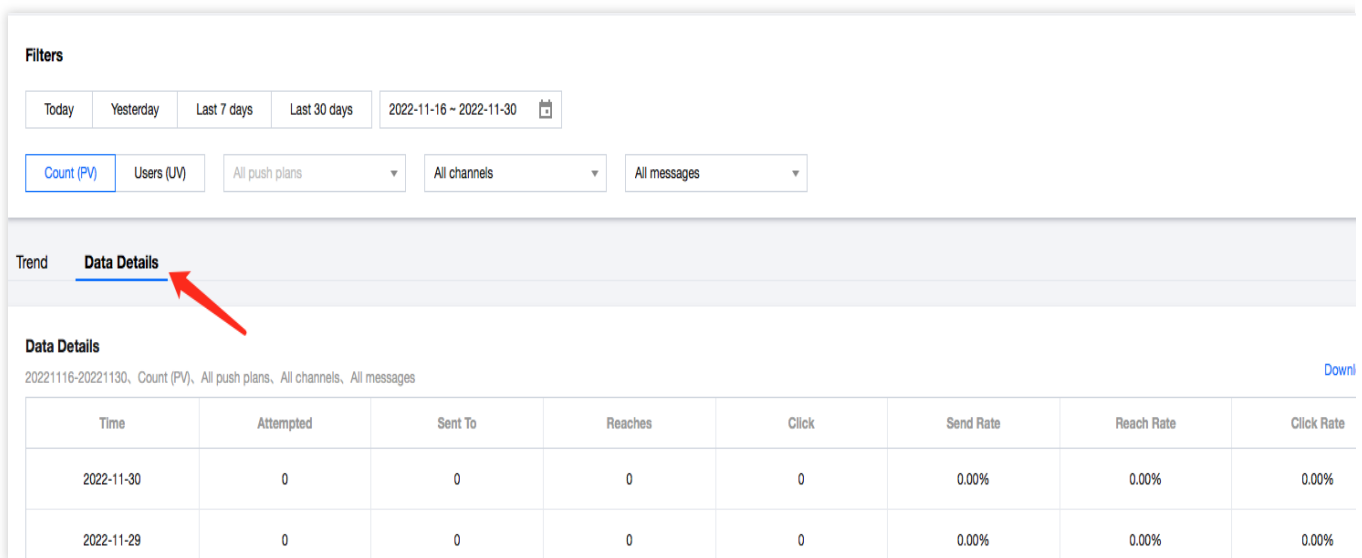
## Trend

In the **Trend** area, you can view push conversion trends in the time dimension.



## Details

1. Click the **Details** tab to view the detailed data displayed according to specified filters.
2. Click **Download Data** to export data tables to CSV files to analyze the data conversion effect in more detail.



### Note:

If the statistical period spans days, data is classified according to the time when the data is reported. For example, if a message was delivered and reached device A on March 1, but it was clicked on March 2, the corresponding **Planned**

**Sends**, **Actual Sends**, and **Total Reaches** will be calculated for March 1, but the corresponding **Total Clicks** will be calculated for March 2.

# Push Plan

Last updated : 2024-01-16 17:34:39

## Overview

A push plan helps operational personnel and developers manage multiple push tasks for different push targets in the form of combinations and allows them to view the overall effect of pushes by plan and evaluate the achievement of overall goals.

## Use Cases

### Managing push tasks for different operational goals

Multiple push tasks may need to be created for phased events such as 6/18 and 11/11 shopping festivals or for the same operational goal such as improving the retention rate of new users. In such cases, you can create a push plan to manage them and view the number of messages reached and user clicks at the plan level, so as to understand the conversion effect and achievement of marketing goals.

### Notifications and pushes triggered by user

For common push types such as like, comment, share, and private message, you can group all single-user push tasks of the same type into the same plan and view data such as the number of push messages and user clicks in the type by day, so as to evaluate the promotional effect of the message type on user activity.

## Directions

### Creating a push plan in the console

#### Method 1:

1. Log in to the [Tencent Push Notification Service console](#).
2. In the left sidebar, choose **Message Management** > **Push Plan** and click **Create Plan**.
3. In the **Create Plan** dialog box, enter the plan name and description and click **OK**.

After the plan is created, you can view its name, description, and number of push tasks and edit or delete it at any time on the push plan page. Note that the default plan cannot be edited or deleted.

#### Method 2:

1. Log in to the [Tencent Push Notification Service console](#).
2. In the left sidebar, choose **Message Management** > **Task List** and click **Create Push**.

3. Click **Create New Plan**.

### Set Up Push Task

Push Platform

Test

Test-Android-long-name-test

Push Plan ⓘ

Choose Plan

+Create New Plan

Push Type

Notification bar message

In-app message

Notification Title

Enter notification title

Notification Content

Enter notification content

Push Time

Immediate

Scheduled

Loop

Push Target

Please select

Advanced Settings ▾

4. In the pop-up dialog box, enter the plan name and description and click **OK**.

After the plan is created, it will be automatically selected for the current push.

## Viewing the push tasks under a push plan

1. Log in to the [Tencent Push Notification Service console](#).

2. In the left sidebar, choose **Message Management** > **Push Plan**, select the target push plan, and click **Details** to go to the push task list page.

New Plan					Creation Method	TPNS Console	API	Search Plan
Plan Name	Plan Descriptions	Created Time	Tasks Counts	Operation				
Default Plan	System default plan, cannot be deleted.	2020-07-10 10:48:22	0	<a href="#">Details</a>				
Total Items: 1					10 / page			

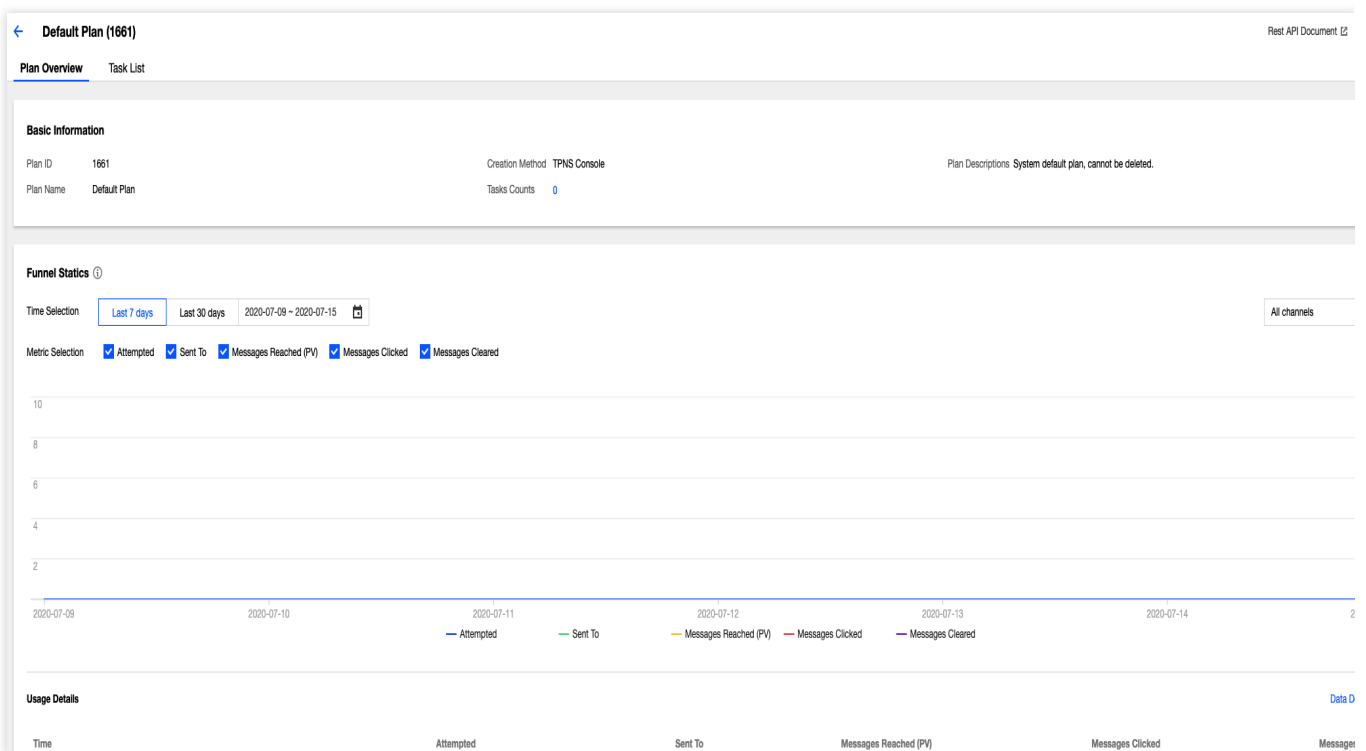
## Creating a push task in a push plan

On the push plan details page, click the **Task List** tab, and click **Create Push**.

Create Push	Last 7 days	2020-07-09 ~ 2020-07-15	Notification bar message	All	TPNS Console	Enter PushID	Q	Aggregat
Delivery Time	PushID	Notification Title	Notification Content	Push Target	Status	Y	Operation	
No data yet								
Total Items: 0								10 / page 1 / 1 page

## Viewing the aggregated statistics of a push plan

On the push plan details page, click the **Plan Overview** tab, and you can view the aggregated statistics of the push plan.



## Using RESTful APIs (to create a push plan or specify a push plan for push)

### Creating a push plan

Create a push plan as instructed [here](#). Then, you can specify it for message push.

### Sample creation

```
{
  "planName": "VIP_Level 15",
  "planDescribe": "VIP member benefits"
}
```



## Specifying a push plan for push

When calling the push API, you can set the `plan_id` to specify the ID of the push plan for the target recipients. For more information, please see [Optional Parameter Description](#) in the RESTful API document.

Sample push:

```
{
  "audience_type": "token",
  "environment": "dev",
  "token_list": [ "05da87c0ae*****fa9e08d884aada5bb2" ],
  "message_type": "notify",
  "plan_id": "20200704",
  "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\\"}"
    }
  }
}
```

# Push Settings

Last updated : 2024-01-16 17:34:39

## Overview

Proper message push can quickly improve user activity, increase conversion rate, and bring other benefits at a very low cost. However, improper message push will adversely affect product reputation, and even sharply increase the application uninstalls.

Tencent Push Notification Service provides the following capabilities for you to implement proper push in a secure and efficient manner:

Frequency capping: limits the pushes to avoid annoying users

Message deduplication: blocks identical pushes caused by misoperation at the system level

Specified time range: sets a time range that allows pushes to avoid disturbing users late at night.

## Use Instructions

### Frequency Capping

#### Background

Ecommerce applications often need to push promotion and marketing messages to targeted groups that are selected according to their browsing habits, shopping frequency, and other information. As a result, the same user may receive multiple push notifications in a short period of time, which seriously affects user experience and even causes the user to turn off notifications or uninstall the application.

#### Directions

1. Log in to the [Tencent Push Notification Service console](#).
2. Select **Message Management** > **Push Settings** on the left sidebar.
3. Click **Enable** in the **Frequency Capping** section.
4. Enter an integer within 1-100 or keep the default value 3 in the text box of **Max Pushes to a Device per Day**. Select a [push plan](#) to which the setting applies, and click **Save** to complete the configuration.

### Message Deduplication

#### Background

News and video applications usually need to preemptively reach out to users with breaking news or hot issues for users to quickly get the latest and most concerned information. However, receiving repeated pushes in a short period

may irritate users. This feature avoids duplicate pushes to users.

### Directions

1. Log in to the [Tencent Push Notification Service console](#).
2. Select **Message Management** > **Push Settings** on the left sidebar.
3. Click **Enable** in the **Message Deduplication** section.

After this feature is enabled, the system will block identical pushes to the same recipients within 1 hour to prevent users from receiving repeated pushes in a short period.

### Note:

This applies only to pushes to all devices, to multiple accounts, by tag combination, or by user group.

## Specified Time Range

### Background

Tools or applications with system notification requirements usually need to notify users when the user or system function status changes. However, pushing notifications late at night will disturb users, thus damaging the reputation of the applications and even causing uninstallation. This feature sets a time range that allows pushes to avoid disturbing users late at night.

### Directions

1. Log in to the [Tencent Push Notification Service console](#).
2. Select **Message Management** > **Push Settings** on the left sidebar.
3. Click **Enable** in the **Specified Time Range** section to set a time range that allows pushes, and click **Save**. The pushes outside the specified time range will not be delivered.

### Note:

The **Specified Time Range** is a period that allows pushes to deliver rather than to reach devices. For example, if the time range is set to 15:00-16:00, only push tasks created within this period will be delivered, and other push tasks will be blocked.

# Message Recall

Last updated : 2024-01-16 17:34:39

## Use Cases

If a notification message sent by an application contains an error or incorrect redirect link, it will give end users a negative impression of the involved product/service after they view or click it. In this case, you should fix the problem in time. You can choose to terminate, recall, or override the message, or cancel the related scheduled task if there is any. This document describes how to do so through the [console](#) and [RESTful APIs](#).

## Overview

Message termination: terminates all offline messages within the offline storage period of the task.

Message recall: terminates the current message and makes it disappear from the notification center on devices where it has arrived but has not been clicked. A recall is imperceptible to users.

Message override: terminates the current message and overrides it with a new one. After a successful override, only the new message will be displayed in the device notification center.

Scheduled task cancellation: Cancels a scheduled task that has not yet been scheduled. After successful cancellation, the scheduled push task will not be delivered any more.

## Use Limits

Platform	Termination	Recall	Override	Scheduled task cancellation
Android	Supported	This feature is still being upgraded and not available yet.	Supported only for the Tencent Push Notification Service channel, Mi channel, Meizu channel, and Huawei devices on EMUI 10 and later	Supported
iOS	Not supported	This feature is still being upgraded and not available yet.	Supported	Supported

**Note:**

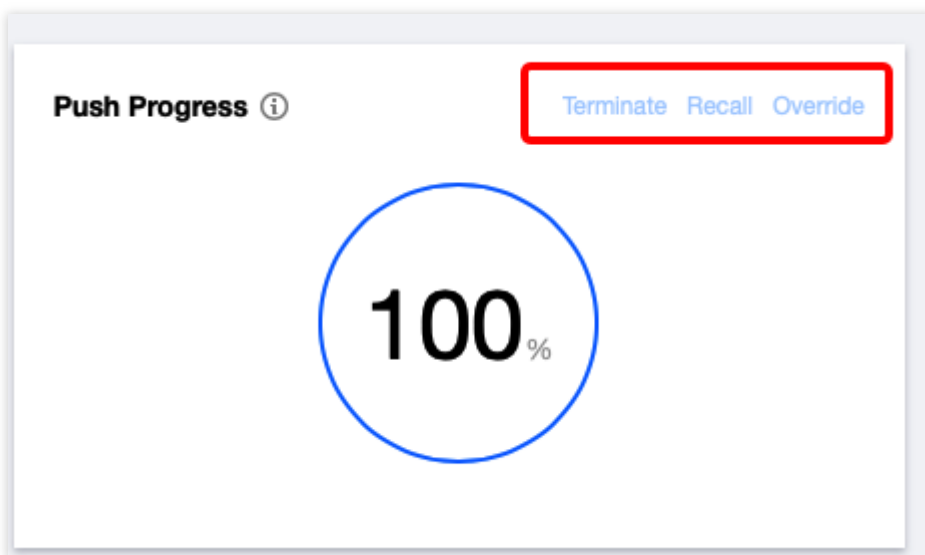
Message termination, recall, and override are supported only for notifications whose push target is **push to all devices**, **push to a list of accounts**, or **push to devices with specific tags**.

Scheduled task cancellation is supported only for notifications whose push target is **push to all devices**, **push by account package**, and **push to devices with specific tags**.

## Using the Console

### Terminating, overriding, or recalling a push task

1. Log in to the [Tencent Push Notification Service console](#).
2. Click **Message Management** > **Task List** in the left sidebar.
3. Click **View Details** for the push task to be terminated, overridden, or recalled.
4. In the **Push Progress** column in the top-right corner of the push details page, terminate, override, or recall the task.

**Note:**

When you select message override, for vivo, OPPO, and Huawei (below EMUI 10) devices that do not support message override, you can choose whether to continue the message delivery:

If you choose to continue the delivery, the notification message will be overridden, and both the new and original messages will appear in the device notification center at the same time.

If you choose to abort the delivery, no new notification messages will be sent to the above devices that do not support message override.

5. After the operation succeeds, you can return to the **Task List** page and view the current status of the task in the **Status** column.

### Cancelling a scheduled task

1. Log in to the [Tencent Push Notification Service console](#).
2. Click **Message Management** > **Task List** in the left sidebar.
3. Locate the push task whose scheduled task is to be canceled, and click **Cancel Push**.
4. After the operation succeeds, you can see that the status of the push task becomes **Canceled** in the **Status** column.

## Using RESTful APIs

### Message termination

#### API description

Request method: POST

Request Address

```
Service URL/v3/push/stop_push_msg
```

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](#).

#### Request parameters

Parameter	Type	Required	Description
pushId	String	Yes	Push task ID

#### Sample request

```
{
  "pushid": "43214535"
}
```

### Recalling a message (This feature is still being upgraded and not available yet.)

#### API description

Request method: POST

Request Address

```
Service URL/v3/push/revoke_push_msg
```

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

**Note:**

Message recall will terminate the offline message delivery of the current push task by default.

**Request parameters**

Parameter	Type	Required	Description
pushId	String	Yes	Push task ID

**Sample request**

```
{
  "pushid": "150032"
}
```

**Message override****Step 1. Query the `collapse_id` of the push task**

Call the [API for querying push information for one task](#) and get the `collapse_id`, for example, 0001, from the corresponding response parameter.

**Step 2. Call the push API to override the original push content**

When you call the [push API](#), add the `collapse_id`, for example, 0001, obtained in **step 1**. You can also set the `force_collapse` field to decide whether to deliver the message to devices that do not support message override.

**Note:**

Message override will terminate the offline message delivery of the original push task by default.

**Sample push**

```
{
  "audience_type": "all",
  "collapse_id": 0001,
  "force_collapse": false,
  "message_type": "notify",
  "message": {
    "title": "Override message 0001",
    "content": "It's a nice day today"
  },
  "platform": "android"
}
```

**Canceling a scheduled task**

## API description

Request method: POST

Request Address

```
Service URL/v3/push/cancel_timing_task
```

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

### Note:

Cancel a scheduled task that has not yet been scheduled.

## Request parameters

Parameter	Type	Required	Description
pushId	String	Yes	Push task ID

## Sample request

```
{
  "pushid": "15003211"
}
```



# Advanced Push Features

## Tags

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

Tag is a feature used in targeted push where you can call Tencent Push Notification Service SDKs or server APIs to bind one or more tags to devices. After that, you can push messages based on the tags, which makes lean operations easier.

### Note:

Tag push is a batch push feature and has frequency limits. It's not recommended for scenarios where a push tag is bound to only a small number of devices or frequent pushes are required. For such scenarios, you can use the feature of push to an account or a list of accounts accordingly.

## Tag Push Scenarios

### User engagement and reactivation

Application operation often requires message reminders for new users, which is an important part of the new user experience and improves the retention rate of new users. By selecting the `new devices` tag provided by Tencent Push Notification Service during push, you can push messages to users registered on the specified date with ease. In addition, we also provide the `inactive users` tag, with which you can specify users who have not been active for N days as the push target and push messages to them for user reactivation, thus increasing the number of active application users.

### Event subscription notification

Your live streaming application will stream a football match at 18:00 on October 24, and the live stream will be available for reservation on October 20. You want to push a message about the upcoming start to users who subscribe to this program before the live stream starts.

If a user subscribes to this program, the title `10241800 Football` can be used as a tag to bind to the user device token. When the live stream is about to start, you can select the `Football` tag to push a notification to inform the user of the match start. After the match ends, you can call the tag unbinding API of Tencent Push Notification Service to unbind the `10241800 Football` tag from the device token.

### Renewal notification

You want to push a renewal notification in application A to users whose membership will expire in three days. Assume that a device token is bound to tags `football` and `deadline:20200210`. If a user renews the membership for a month on February 9, 2020, you need to replace the tag `deadline:20200210` with

`deadline:20200310` , i.e., the tag `deadline` can have only one value (the latest value). In this case, you can call the key-value overriding API provided by Tencent Push Notification Service to unbind the tag `deadline:20200210` and then bind the tag `deadline:20200310` without affecting other tags. When pushing a renewal notification (scheduled push is supported), set the tag to the current date plus three days; for example, if the current date is March 7, 2020, you can push the renewal notification to devices with the tag `deadline:20200310` .

## Tag Overview

Tencent Push Notification Service provides two types of tags: custom tags and preset tags. Tag categories are as follows:

Tag Type	Scenario	tag_type Built in Tencent Push Notification Service	Constraints	Example
Custom tag	Custom tag, such as meeting ID, class ID, and user hobbies (like basketball and digital products)	xg_user_define	Up to 10,000 custom tags are allowed (to increase the quota, <a href="#">contact our online customer service</a> ) One device token can be bound to up to 100 custom tags (to increase the quota, <a href="#">contact our online customer service</a> ) One custom tag can be bound to an unlimited number of device tokens	love_basketball, love_shopping, male
Preset tag	Application version	xg_auto_version	Preset in Tencent Push Notification Service, unlimited	1.0.1, 1.0.2
	Province	xg_auto_province		guangdong, hunan, shanghai
	City	xg_auto_city		beijing, tianjin, chongqing, etc.
	Active	xg_auto_active		20200521,

	information			20200522
	Tencent Push Notification Service SDK version	xg_auto_sdkversion		1.1.5.4, 1.1.6.1
	System version	xg_auto_systemversion		10.0.0, 12.4.5
	System language	xg_auto_systemlanguage		zh, en, ja
	Country/Region	xg_auto_country		CN, US (uppercase letters)
	Phone brand	xg_auto_devicebrand		xiaomi, huawei
	Model	xg_auto_deviceversion	Samsung Note4, Vivo Y75A	-
	Continuously active	Does not support API call currently	Devices active in the last N days. Value range: [1,30]. Format: string	Devices active in the last "10" days
	Continuously inactive	Does not support API call currently	Devices inactive in the last N days. Value range: [1,30]. Format: string	Devices inactive in the last "10" days
Recently registered	Does not support API call currently	Devices recently registered. The tag value is [startDate, endDate] in the format of [YYYYmmdd,YYYYmmdd] The range between <code>startDate</code> and <code>endDate</code> cannot exceed 30 days <code>endDate</code> cannot be the current date <code>startDate</code> cannot be more than 90 days ago <code>startDate</code> and <code>endDate</code> can be the	Devices registered within [20200901,20200910]	-

		same		
--	--	------	--	--

**Note:**

When you push by tag through API, you need to use the `tag_type` built in Tencent Push Notification Service to set the tag type.

## Preparations

### Managing custom tags

You can customize device tag names. Currently, Tencent Push Notification Service allows you to set tags through RESTful APIs and device SDKs.

#### Method 1. Set tags through RESTful APIs

Bind and unbind a custom tag:

See [Tag Binding and Unbinding](#).

#### Method 2. Set tags through device SDKs

For the iOS SDK, see [Setting custom tag](#).

For the Android SDK, see [Setting custom tag](#).

**Note:**

One device can be bound to up to 100 tags (to increase the quota, [contact our online customer service](#))

One app can be bound to up to 10,000 tags (to increase the quota, [contact our online customer service](#))

One tag can contain up to 50 bytes.

Up to 500 tags can be bound or unbound in one request.

### Custom tag use cases and keywords

Tag push is suitable for scenarios where more than 10 devices are bound to a tag and less than 10 pushes are required per day. For other scenarios, account push (binding an account instead of a tag to multiple devices for push) is recommended.

**Keyword**

A colon (:) is the keyword for separating the key and value in a key-value pair for user tag binding. For example, if you assign the tag `level:3` to a device token, the Tencent Push Notification Service backend will take `level` as the tag key and `3` as the tag value, while the original tag `level:3` is pushed. Storage based on key-value is mainly to facilitate subsequent overwriting of tags of the same type.

### Binding/Unbinding tags

Tencent Push Notification Service provides APIs for binding/unbinding a single tag to/from a single device, a single tag to/from multiple devices, multiple tags to/from a single device, and multiple tags to/from multiple devices.

## Binding/Unbinding a single tag to/from a single device

### Recommended scenarios

1. Call the device SDK API, for example, to get a user's subscribed channel in the application and bind/unbind the channel tag to/from the device token.
2. Call the RESTful API occasionally, for example, to perform integration testing.

### Tag binding method

```
{
  "operator_type": 1,
  "tag_list": ["tag"],
  "token_list": ["token"]
}
```

### Tag unbinding method

```
{
  "operator_type": 2,
  "tag_list": ["tag"],
  "token_list": ["token"]
}
```

### Use limits

A tag can contain up to 50 bytes.

The API is called synchronously.

## Binding/Unbinding multiple tags to/from a single device

### Recommended scenarios

1. Call the device SDK API, for example, to get a user's characteristics tags such as age, province, and gender in the application and bind/unbind them to/from the device token in batches.
2. Call the RESTful API, for example, to get a device's user subscription information tags such as marital status and hobbies (football, movies, etc.) through other internal channels, bind/unbind them to/from the device token in batches.

### Tag binding method

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

### Tag unbinding method

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

```
"token_list": ["token"]
}
```

### Use limits

A tag can contain up to 50 bytes.

Up to 100 tags can be included in one call.

The API is called synchronously.

### Binding/Unbinding a single tag to/from multiple devices

#### Recommended scenarios

Call the RESTful API, for example, to bind/unbind the `football` tag to/from all users who like/dislike football in batches.

#### Tag binding method

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

#### Tag unbinding method

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

### Use limits

A tag can contain up to 50 bytes.

Up to 500 device tokens can be included in one call.

The API is called synchronously.

### Binding/Unbinding multiple tags to/from multiple devices

#### Recommended scenarios

Call the RESTful API, for example, to bind/unbind the `football` and `basketball` tags to/from all users who like/dislike football and basketball in batches.

#### Tag binding method

```
{
  "operator_type": 9,
  "tag_token_list": [{"tag": "tag1", "token": "token1"}, {"tag": "tag2", "token": "token"}
}
```

## Tag unbinding method

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

### Use limits

A tag can contain up to 50 bytes.

Up to 500 device tokens can be included in one call.

The API is called synchronously.

## Tag overriding

Tencent Push Notification Service provides two tag overriding methods: general overriding and overriding by tag category (key-value overriding). In key-value overriding, a colon (:) is used to separate the key and value in a key-value pair.

### General overriding

#### Recommended scenarios

1. Call the device SDK API. For example, if all channel information subscribed by a device has expired, you need to unbind all the channel tags from the device. However, traversing all tags to unbind them one by one is inconvenient. In this case, you can call this API to overwrite the tags.
2. Call the RESTful API. For example, to set new tags for a device so that it will not be affected by legacy tags, this API can be called to overwrite them.

## Tag overriding method

```
{
  "operator_type": 6,
  "tag_list": ["test", "level:1", "level:2"],
  "token_list": ["token"]
}
```

### Use limits

A tag can contain up to 50 bytes.

The API is called asynchronously. You are recommended to set the call interval to longer than 1 second.

### Key-value overriding

#### Recommended scenarios

1. Call the device SDK API.
2. Call the RESTful API.

**Instructions:** a colon (:) is used for separating the key and value in a key-value pair. For example, `level:2` indicates that the device level is 2. Assume that the device level is upgraded to 3, then you need to delete the tag

`level:2` before adding the tag `level:3`. If you know that the device has only the tag `level:2`, you can call the general overriding API to overwrite it. However, a device usually has multiple tags. If the device has another tag `test` and you want to delete only the tag `level:2`, you need to manipulate all tags of the device or call the relevant Tencent Push Notification Service API to find the corresponding legacy tag and delete it, which is inconvenient. In this case, you can call this API to overwrite only tags with the key `level`.

### Tag overriding method

```
{
  "operator_type": 6,
  "tag_list": ["test:2", "level:3"],
  "token_list": ["token"]
}
```

**API description:** key-value overwriting can be performed properly only when all tags in the `tag_list` have a colon (:). For example, if a token has the tags `test` and `level:1`, after this API is called, the tag list of the token will include `test`, `test:2`, and `level:3`.

### Use limits

A tag can contain up to 50 bytes.

Up to 100 tags can be included in one call.

The API is called asynchronously. You are recommended to set the call interval to longer than 1 second.

### Tag deletion scenarios

Tencent Push Notification Service provides two tag deletion methods: deleting all tags of a single device and deleting specific tags of an application.

#### Deleting all tags of a single device

### Recommended scenarios

1. Call the device SDK API.
2. Call the RESTful API.

**Instructions:** you can use this API to delete all legacy tags of a device. This API is generally used to delete expired tags or set tags again after misoperations.

### Tag deletion method

```
{
  "operator_type": 5,
  "token_list": ["token"]
}
```

### Use limits

A tag can contain up to 50 bytes.

The API is called asynchronously. You are recommended to set the call interval to longer than 1 second.



## Deleting specific tags of an application

### Recommended scenarios

Call the RESTful API.

**Instructions:** you can use this API to delete specific tags of an application, i.e., removing them from the tag list of the application after unbinding them from bound devices. This API is generally used to delete disused tags. For example, you can call this API to delete testing tags added during test after the application is officially released.

### Tag deletion method

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

### Use limits

A tag can contain up to 50 bytes.

Up to 100 tags can be included in one call.

The API is called asynchronously. You are recommended to set the call interval to longer than 1 second.

## Managing preset tags

Preset tags are tags maintained on the Tencent Push Notification Service platform, i.e., tags automatically collected by the SDK when user devices are registered with or connected to the Tencent Push Notification Service server. Currently, Tencent Push Notification Service preset tags include application version, system version, province, active information, system language, SDK version, country/region, phone brand, and phone model.

All devices are bound to the latest preset tags, which will automatically replace the corresponding legacy ones. For example, if the current application version of a device is 1.0.1, when the application is upgraded to 1.0.2, the device will be automatically unbound from the `v1.0.1` tag and then bound to the `v1.0.2` tag.

## Querying tags bound to devices

Log in to the [Tencent Push Notification Service console](#), and click **Message Management > Troubleshooting Tools** in the left sidebar. Then you can query preset or custom tags by device token.

See the figure below:

Token ⓘ

001ca4b76ae2a8cef2f25dfaabd092583037

Query

Validity ⓘ

Yes

Application AccessId ⓘ

1500007822

Application Package Name/BundleID

com.qq.xgdemo1122

First Connection to the Server

2020-05-07 13:00:25

Last Connection to the Server ⓘ

2020-05-10 21:58:18

Push Channel ⓘ

TPNS channel

Third-Party Platform Token ⓘ

-

Device Brand

OPPO (PCAM00)

System Version

10

App Version

3.0

SDK version

1.1.5.4

Registration environment ⓘ

Failed to obtain

Notification Bar Permissions ⓘ

Enable

Associated Accounts

Serial No.	Account Type	Account Details
No associated accounts yet		

Associated Tags

Serial No.	Tag Details
No associated tags yet	

## Directions

Tag push allows you to push messages by preset or custom tags or combinations of preset and custom tags (combined by the "AND" or "OR" relationship) according to your operational requirements.

### Setting the policy in the console

You can push a message by tag in the Tencent Cloud console as follows.

1. Select the tag combination type, e.g., custom tag or a category of preset tags.

**Set Up Push Task****Preview**

Push Platform

DuoDuo

DuoDuo-iOS(正式)

Push Type

Notification bar message

Silent message

(Optional) Notification Title

Enter notification title

(Optional) Notification Subtitle

Enter notification subtitle

Notification Content

Enter notification content

Push Time

Immediate

Scheduled

Loop

Push Target

Tag

Operator logic priority between different rules:"AND" higher than "OR"

Province

Yes

shanxi(356)

guizhou(231)

Custom Tag

System Version

App Version

Province

Active Information

System Language

You

tag(s)

Push Environment

Production environment

mutable-content

Advanced Settings

2. Select the tags for which you want to push a message after selecting the tag type.

Setup Push Task

Preview

Push Platform

DuoDuo

DuoDuo-iOS(正式)

Push Type

Notification bar message

Silent message

(Optional) Notification Title

Enter notification title

(Optional) Notification Subtitle

Enter notification subtitle

Notification Content

Enter notification content

Push Time

Immediate

Scheduled

Loop

Push Target

Tag

Operator logic priority between different rules:"AND" higher than "OR"

System Version

Yes

13.4(489) × 13.2.2(65) ×

+ ▾

+ ▾

13.1  
12.3.2  
13.4.1  
11.1  
Version 13.1.3 (Build 17A878)  
Version 10.2 (Build 14C82)

You can add 9 more rule(s) and 48 more tag(s)

Push Environment

Development environment

Production env

mutable-content ⓘ

Advanced Settings ▾

After a tag is selected, the number of devices bound to the selected tag will also be displayed. The tag combination in the figure above indicates to push a message to users in Guangdong who were active within two days. Then, click **Test Preview** to push the message to the corresponding target devices.

## Calling the push API for tag push

Set the `audience_type` (push target) in the push API request parameter to `tag` to enable tag push. For more information, see [Push API](#).

API example: push a message to male users in Guangdong and Jiangsu who were active on 2020-04-23, 2020-04-22, or 2020-04-21.

```
{
  "audience_type": "tag",
  "tag_rules": [
    {
      "tag_items": [
        {
          "tags": [
            "guangdong",
            "jiangsu"
          ],
          "is_not": false,
          "tags_operator": "OR",
          "items_operator": "OR",
          "tag_type": "xg_auto_province"
        },
        {
          "tags": [
            "20200421",
            "20200422",
            "20200423"
          ],
          "is_not": false,
          "tags_operator": "OR",
          "items_operator": "AND",
          "tag_type": "xg_auto_active"
        }
      ],
      "operator": "OR",
      "is_not": false
    },
    {
      "tag_items": [
        {
          "tags": [
            "male"
          ],
          "is_not": false,
          "tags_operator": "OR",
          "items_operator": "OR",
          "tag_type": "xg_user_define"
        }
      ],
      "operator": "AND",
      "is_not": false
    }
  ]
}
```

```
    }
  ],
  "message_type": "notify",
  "message": {
    "title": "Test title",
    "content": "Test content",
    "android": {
      "ring": 1,
      "ring_raw": "ring",
      "vibrate": 1,
      "lights": 1,
      "clearable": 1,
    },
    "custom_content": "{\\"key\\":\\"value\\"}"
  }
}
```

## FAQs

### 1. Will the API for querying bound devices by tag be made publicly available?

The API is currently for internal use only and will not be made publicly available in the future for the sake of system stability. You can select a tag on the push page in the console to query its bound devices.

### 2. Will the API for querying bound tags by device token be made publicly available?

The API is currently for internal use only and will not be made publicly available in the future for the sake of system stability. You can enter a token in the toolkit in the console to query its bound tags.

# Multi-Package Name Push

Last updated : 2024-01-16 17:34:39

For various types of Android apps, especially Android games, it is a common practice in business operations to customize versions for different channels and use different package names; however, this will lead to excessive workload for subsequent message pushes, as each package requires separate push, which is time-consuming and laborious and makes it difficult to achieve precise push and high efficiency.

The new "multi-package name push" feature of Tencent Push Notification Service is a convenient solution to this problem. After this feature is enabled, you can easily add a multi-package name for different channels, and then one message push is enough to reach all channels.

## Scenarios

For marketing promotions, a game vendor needs to publish an event announcement to all game players. The game is released in multiple app markets (such as CoolAPK, Anzhi, Wandoujia, and 360) with different package names. In this case, Multi-Package Name Push can be used to send the event announcement to all packages at a time.

## Preparations

### Configuring in console

1. Log in to the [Tencent Push Notification Service console](#) and go to the **Product Management** page.
2. Select the application for which to configure multiple package names and click **Configuration Management**.

Test <span>Guangzhou</span>						
Platform	Application Name	Access ID	Service Status	Operation		Service Management
Android	Test-Android-long-name-test	1500003223	Waiting for payment	Create Push	Push Record Configuration Management	Delete Contact sales Apply for Free Trial
iOS	Test-iOS	1600003224	Waiting for payment	Create Push	Push Record Configuration Management	Delete Contact sales Apply for Free Trial

3. If the application does not have a main package name, you need to enter one and then click **Add another package name** to enter a channel-specific package name.

**Application Information**

Package Name

The package name above will be used as the main package name of the application. If there are multiple package names, you can [click here to add](#) or add them later on the configuration management page

Save

4. If the application already has a main package name, click **Edit** to open the **Package Name Management** section, and click **click here to add** to configure a package name.

Quick Integration

Manual Integration

Download

**Application Information**

Application Name

test-Android

Package Name

com.test

⚠

Modifying the package name will cause the vendor channel configuration to be cleared. Please

Main package name

The package name above will be used as the main package name of the application. If there are multiple package names, you can [click here to add](#) or add them later on the configuration management page

**Note:**

Up to 50 package names can be configured.

**Configuring vendor channel for multi-package name**

If your application has multiple package names, when you need to deliver messages to package names through vendor channels, you need to apply for a vendor key for each package name and configure it in the [Tencent Push Notification Service console](#) > **Message Management** > **Basic Config**.

Taking Huawei as an example, if the application has multiple package names configured, multiple key configurations will appear after the Huawei channel is enabled, and you need to complete the configuration for each package name. Otherwise, messages to devices under a package name with incomplete configuration will be sent through the Tencent Push Notification Service channel after multi-package name push is enabled.



Quick Integration

Manual Integration

Download

### Application Information

Application Name test-Android

Package Name com.test.demo and 2 more

### Vendor Channel

[How to Integrate with Vendor Channel](#)

#### Xiaomi Official Push Channel

After integration, system-level push delivery can be implemented on Xiaomi devices.

You need to install the push SDK for Xiaomi and publish

Please go to the vendor's platform to apply for the key for each package name. If not configured, the push for the package name will be sent through the self-built channel when multi-package name push is enabled

Package Name com.test.demo

Appld

SecretKey

Package Name com.test.demo2

Appld

SecretKey

## Integrating the SDK

After configuring a package name, get the `AccessID` and `AccessKey` , and configure as instructed in [Android SDK Integration Guide](#) or the **Quick Integration** process in the console.

The screenshot displays the Tencent Push Notification Service console interface on the left and an IDE window on the right. In the console, the 'Quick Integration' tab is selected and highlighted with a red arrow. Below it, the 'Application Information' section shows 'test-Android' as the Application Name and 'com.test.demo and 2 more' as the Package Name. The 'Vendor Channel' section includes a link 'How to Integrate with Vendor Channel' and a toggle for 'Xiaomi Official Push Channel'. On the right, a blue box labeled '1' indicates the first step: 'Download and add the configuration file'. Below this, a blue button says 'Download tpns-configs.json'. A text instruction states: 'Add the downloaded tpns-configs.json file to the root directory of the application module (E)'. The IDE window shows a project structure with 'tpns-configs.json' highlighted in the 'app' directory.

**Quick Integration** Manual Integration Download

**Application Information**

Application Name test-Android

Package Name com.test.demo and 2 more

**Vendor Channel** [How to Integrate with Vendor Channel](#)

**Xiaomi Official Push Channel**

After integration, system-level push delivery can be implemented on Xiaomi devices.

You need to install the push SDK for Xiaomi and publish

**1 Download and add the configuration file**

**Download tpns-configs.json**

Add the downloaded tpns-configs.json file to the root directory of the application module (E)

Project tpns app tpns-configs.json

Project ~\workspace\android\tpns

.gradle

.idea

app

libs

src

.gitignore

app.iml

build.gradle

proguard-rules.pro

tpns-configs.json

gradle

.gitignore

build.gradle

gradle.properties

gradlew

gradlew.bat

local.properties

settings.gradle

tpns.iml

External Libraries

Scratches and Consoles

## Directions

### Setting the policy in the console

After the above configuration is completed and confirmed, you can enable multi-package name push in the [Tencent Push Notification Service console](#) > **Message Management** > **Task List** > **Create Push** > **Advanced Settings**, as shown below:

Advanced Configuration ▲

Custom Push Speed ⓘ

GroupID ⓘ

tpns\_20200415

Additional Parameter(s)

Rich Media Content ⓘ

Offline Storage

−

24

+

The offline storage period is 0-72 hours. Users who connect to internet within this period can receive this push message.

Open Location

Application

In-APP Page

URL

Custom Intent

Reminder

☒ Sound

Default Sound

Custom

☒ Vibration

☒ Breathing Light

Multi-Package Push

After multi-package name push is enabled, the push will be delivered to all devices that

After enabling Multi-Package Name Push, a push will be delivered to devices that match the push target under all package names.

**Note:**

The multi-package name push feature is available only to Android. If a vendor channel corresponding to a package name is not configured, messages to devices registered under the package name will be delivered through the Tencent Push Notification Service channel.

## Setting the policy with RESTful APIs

Set `multi_pkg` to `true` in the optional parameter of the Rest API to enable multi-package name push. For more information, please see [Push API Parameter Description](#).

Sample push:

```
{
  "audience_type": "token",
  "token_list": [
    "05da87c0ae5973*****9e08d884aada5bb2"
  ],
  "message_type": "notify",
  "multi_pkg": true,
  "message": {
    "title": "Push title",
    "content": "Push content",
    "android": {
      "custom_content": "{\"key\":\"value\"}"
    }
  }
}
```

# Notification Tap-to-Redirect

Last updated : 2024-11-19 11:31:42

## Use Cases

You can redirect subscribers who tap your notification to the specified in-app page, HTML5 page, or deep link to meet your needs in different use cases.

## Application Scope

Platform	Type
Android	Intent redirect: jumps to the specified in-app page. You can also pass in custom parameters. Open application: directly goes to the application's homepage. URL: opens the browser and accesses the specified webpage. In-app activity: jumps to the specified in-app page.
iOS	Opens the app by default. Implements the business logic based on the delivered custom key and value.

## Android Applications

### Note:

In the SDK, a tap on a message can trigger a click event by default, which opens the target page. If a redirect action is configured in `onNotificationClickedResult`, it will conflict with the custom redirect rule specified in the console or API, and the custom redirect rule will fail.

### Configuring SDK

To use Intent redirect, first configure the page to be redirected to in the client application's `AndroidManifest` file. If you want to redirect to the page specified by `AboutActivity`, use the following sample code:

```
<activity
    android:name="com.qq.xg>AboutActivity"
    android:theme="@android:style/Theme.NoTitleBar.Fullscreen" >

    <!-- Other intent-filter -->
```

```
<!-- <intent-filter> ... </intent-filter> -->

<!-- AndroidManifest supports configuring multiple intent-filters for an Android
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.DEFAULT"/>
    <!-- Customize the content of the data block to specify your complete scheme
    <!-- According to your configuration, a URL in the `scheme name://host name
    <!-- To avoid conflicts with the redirection destination pages of other app
    <data
        android:scheme="Scheme name"
        android:host="Host name"
        android:path="/Path name" />
</intent-filter>
</activity>
```

## Using the console

### Intent redirect (recommended)

To set the Intent redirect in the [Tencent Push Notification Service console](#), go to **Task List > Create Push > Advanced Settings > Open Location** and enter the following information:

Application	Custom Intent(Recommended)	URL	In-App Activity
<input type="text" value="Enter the Intent info configured on the AndroidManifest of APP"/>			
<a href="#">Configuration Guide</a>			

### Open app

Tapping on the notification that is pushed via the console will open the application by default.

### URL redirect

To set the URL redirect in the [Tencent Push Notification Service console](#), go to **Advanced Settings > Open Location** and enter the following information:

Application	Custom Intent(Recommended)	<b>URL</b>	In-App Activity
<input type="text" value="Enter URL, e.g. http://tpns.qq.com"/>			

### In-app activity redirect

This method will be disused, so we don't recommend using it. Go to **Advanced Settings > Open Location** in the console and enter the following information:

Open Location	Application	Custom Intent(Recommended)	URL	<b>In-App Activity</b>
<input type="text" value="This method is gradually not supported and not recommended. Fill in the a"/>				

### Using RESTful APIs

Add the `action` and `action_type` fields under `body.message.android` of the push message body.

Field	Type	Parent Project	Default Value	Required	Description
action	Object	Android	1	No	This sets the action after the notification bar is tapped; the default action is to open an application.
action_type	Integer	Action	1	No	One-tap action. Valid values: <div> 1 : opens activity or app  2 : opens browser  3 : opens Intent (recommended; for more information, see <a href="#">Configuring SDK</a>) </div>

### Intent redirect (recommended)

Below is a sample of a complete message:

```
{
  "audience_type": "token",
  "token_list": [
    "04xxx993"
  ],
  "message_type": "notify",
```

```
"message":{
  "title": "xxx",
  "content": "xxx",
  "android": {
    "action": {
      "action_type": 3, // Action type. `1`: opens activity or app; `2`: open
      "intent": "xgscheme://com.tpns.push/notify_detail" // The SDK must be v
    }
  }
}
```

If you want to pass in custom parameters such as `param1` and `param2` , use the sample code below:

```
{
  "audience_type": "token",
  "token_list": [
    "04xxx993"
  ],
  "message_type": "notify",
  "message":{
    "title": "xxx",
    "content": "xxx",
    "android": {
      "action": {
        "action_type": 3, // Action type. `1`: opens activity or app; `2`: open
        "intent": "xgscheme://com.tpns.push/notify_detail?param1=aa&param2=bb"
      }
    }
  }
}
```

**Note:**

If the client requires more parameters for other responses, refer to [Getting parameters on the client](#).

**Open app**

Below is a sample of a complete message:

```
{
  "audience_type": "token",
  "token_list": [
    "04xxx993"
  ],
  "message_type": "notify",
  "message":{
    "title": "xxx",
```



```
"content": "xxx",
"android": {
  "action": {
    "action_type": 1 // Action type. `1`: opens activity or app; `2`: opens
  }
}
}
```

## URL redirect

Below is a sample of a complete message:

```
{
  "audience_type": "token",
  "token_list": [
    "04xxx993"
  ],
  "message_type": "notify",
  "message": {
    "title": "xxx",
    "content": "xxx",
    "android": {
      "action": {
        "action_type": 2, // Action type. `1`: opens activity or app; `2`: opens
        "browser": {
          "url": "http://tpns.qq.com", // Only HTTP and HTTPS URLs are support
          "confirm": 1 // Whether user's confirmation is required
        }
      }
    }
  }
}
```

## Opening the in-app activity

Below is a sample of a complete message:

```
{
  "audience_type": "token",
  "token_list": [
    "04xxx993"
  ],
  "message_type": "notify",
  "message": {
    "title": "xxx",
    "content": "xxx",
```

```
"android": {
  "action": {
    "action_type": 1, // Action type. `1`: opens activity or app; `2`: opens
    "activity": "com.x.y.MainActivity",
    "aty_attr": { // Activity attribute, only for action_type=1
      "if": 0, // Intent's Flag attribute
      "pf": 0 // PendingIntent's Flag attribute
    }
  }
}
```

## Getting parameters on the client

1. In the `onCreate` method of the page you specify for redirect, add the following intent URI code:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_temp);

    // ...

    // Getting parameters via `onCreate`

    // If the `intent uri` (client customization) mode is used for push, and
    parameters are added to the end of `intent`, you can use this code snippet to
    get the parameter values.
    // For example, if the value of the pushed `intent` is
    `xgscheme://com.tpns.push/notify_detail?param1=aa&param2=bb`,
    // you can use this code snippet to get the values of `param1` and `param2`,
    which are `aa` and `bb` respectively.
    Uri uri = getIntent().getData();
    Log.i(TAG, "onCreate get data uri: " + uri);
    if (uri != null) {
        String url = uri.toString();
        String p1 = uri.getQueryParameter("param1");
        String p2 = uri.getQueryParameter("param2");
    }

    // This is new from SDK 1.3.2.0.
    // If `custom_content` is set when the push task is created, you can use this
    API to get the content of the `custom_content` string.
    String customContent = XGPushManager.getCustomContentFromIntent(this,
    this.getIntent());
```

```
// ...  
}
```

2. If your activity page is the application's resident page (for example, `launchMode` is set to `singleTop` or `singleTask` ), the intent content for tap-to-redirect will be triggered by the `onNewIntent` method of the activity page. Please add the following to the `onNewIntent` method to get the intent URI code:

```
@Override  
protected void onNewIntent(Intent intent) {  
    super.onNewIntent(intent);  
  
    // ...  
  
    // Getting parameters via `onNewIntent`  
  
    // If the `intent uri` (client customization) mode is used for push, and parameter  
    // For example, if the value of the pushed `intent` is `xgscheme://com.tpns.push/n`  
    // you can use this code snippet to get the values of `param1` and `param2`, which  
    Uri uri = intent.getData();  
    Log.i(TAG, "onNewIntent get data uri: " + uri);  
    if (uri != null) {  
        String url = uri.toString();  
        String p1 = uri.getQueryParameter("param1");  
        String p2 = uri.getQueryParameter("param2");  
    }  
  
    // This is new from SDK 1.3.2.0.  
    // If `custom_content` is set when the push task is created, you can use this API  
    String customContent = XGPushManager.getCustomContentFromIntent(this, intent);  
  
    // ...  
}
```

3. If the parameters passed in contain special characters, you can use `URLEncoder` to encode the parameter values when creating the push and use `URLDecoder` to decode them at the terminal. The following is a sample:

```
Uri uri = getIntent().getData();  
if (uri != null) {  
    String p1 = uri.getQueryParameter("param1");  
    String value1 = "";  
    try {  
        // The value of the custom parameter `param1` contains special characters. You  
        value1 = URLDecoder.decode(p1, "UTF-8");  
    } catch (UnsupportedEncodingException e) {  
        Log.w("TPNS", "URLDecode param failed: " + e.toString());  
    }  
}
```

```
// The custom parameter `param2` is not encoded with URLEncode and can be obtained
String value2 = uri.getQueryParameter("param2");
Log.i("TPNS" , "value1 = " + value1);
}
```

## iOS Applications

You can pass in custom parameters in the notification for delivery, and implement the redirect or other business logic by parsing the parameters obtained on the client.

### Using the console

To set parameters in the [Tencent Push Notification Service console](#), go to **Advanced Settings > Extra Parameter(s)** and enter the following information:

Extra Parameter(s)

☒ Enable

In order to adapt to the advanced features of vendor, TPNS has upgraded the Huawei push protocol to V4. The Huawei V4 protocol does not support carrying service custom parameters through this field. If you have integrated a Huawei vendor channel, it is recommended that you use the **Intent method** to carry custom parameters, otherwise the custom parameters cannot be successfully delivered through the Huawei push channel. If you need to continue to use the original V2 protocol, you can click "Scan Code for Consultation" in the upper right corner of the console to be added to the allowlist.

Key	Value
-----	-------

### Using RESTful APIs

Add the following `custom_content` field under `body.message.ios` of the push message body.

Field	Type	Parent Project	Default Value	Required	Description
custom_content	String	ios	Empty	No	Custom parameter for delivery, which must be serialized to a JSON string.

Below is a sample of a complete message:

```
{
  "audience_type": "token",
```

```

    "environment": "dev",
    "token_list": [
        "0250df875c93c555dd3a2ba536b54fc1xxxx"
    ],
    "message_type": "notify",
    "message": {
        "title": "xxx",
        "content": "xxxxxxxxxx",
        "ios": {
            "aps": {
                "alert": {
                    "subtitle": "xxx"
                }
            },
            "custom_content": "{\\"key\\":\\"value\\"}"
        }
    }
}

```

## Getting parameters on the client

If you use iOS SDK integration, you can obtain custom parameters using click callback. This callback applies to the notification messages of the app in foreground, background, and shutdown status.

```

/// Unified message click callback
/// @param response will be `UNNotificationResponse` for iOS 10+/macOS 10.14+, or `
/// @note TPNS SDK1.2.7.1+
- (void)xgPushDidReceiveNotificationResponse:(nonnull id)response withCompletionHan
    NSLog(@"[XGDemo] click notification");
    if ([response isKindOfClass:[UNNotificationResponse class]]) {
        /// Getting messages on iOS 10 or later versions
        NSLog(@"notification dic: %@", ((UNNotificationResponse *)response).notific
    } else if ([response isKindOfClass:[NSDictionary class]]) {
        /// Getting messages on iOS versions earlier than 10
        NSLog(@"notification dic: %@", response);
    }
    completionHandler();
}

```

# Custom Push Speed

Last updated : 2024-01-16 17:34:39

The Custom Push Speed feature is to solve the problem where Tencent Push Notification Service may push messages so fast that some customer servers experience too much connection pressure. Tencent Push Notification Service provides API settings to allow you to control the push speed according to your own server conditions.

## Scenarios

Scenario 1:

You want to push a promotional message to all users, but there is a limit on the number of concurrent visitors on the event page; therefore, you want to control the push speed in order to reduce the connection pressure on your server. In this case, you can set a push speed to limit the number of users allowed to open the event page simultaneously.

Scenario 2:

You have tagged a group of users as "lost users" and want to push a "benefit claim" message to them, so that they will be attracted to open your app; however, you don't want too many users to concurrently visit the event page. In this case, you can set a push speed to limit the number of users allowed to open the event page simultaneously.

## Directions

### Using the console

1. Log in to the [Tencent Push Notification Service console](#).
2. Go to **Push Management > Task List**.
3. Click **Create Push**, expand the **Advanced** section, enable **Custom Push Speed**, and set a speed.

After Custom Push Speed is enabled, the message will be pushed to devices that match the push target at the set speed.

#### Note:

Only push to all devices and push by tag support Custom Push Speed.

The push speed can range from 1,000 to 50,000 pushes per second.

### Using RESTful APIs

When calling the RESTful API, you can set the `push_speed` parameter to push messages at a custom speed. For more information, see the **Optional Parameters** section in [Push API](#).

Below is a sample push:

```
{
```

```
"audience_type": "tag",
"tag_list": {
  "tags": [
    "tag1",
    "tag2"
  ],
  "op": "AND"
},
"push_speed": 50000,
"message_type": "notify",
"message": {
  "Title": "Push title",
  "content": "Push content",
  "android": {
    "custom_content": "{\\"key\\":\\"value\\"}"
  }
}
```

# Channel Policies

Last updated : 2024-01-16 17:34:39

With tightening restrictions over the push limit and frequency of vendor channels, the push reach rate and delivery speed are also affected. For specific restrictions, see:

[Vendor Channel Limit Description](#)

[Vendor Channel QPS Limit Description](#)

Tencent Push Notification Service provides two channel assignment policies: smart assignment and custom. These policies can help improve the overall push reach rate and delivery speed against the restrictions of the vendor channels.

## Channel Overview

Channel Type	Applicable Condition	Supported Mobile Brand
Tencent Push Notification Service channel	The application process is online	All brands
Android vendor channels (Huawei, Mi, Meizu, vivo, OPPO, and FCM)	Application process online or offline	Huawei, Mi, Meizu, vivo, OPPO, OnePlus, Black Shark, realme, iQOO, Honor, and other mobile phones with Google Services Framework
iOS vendor channel (APNs)	The application process is online or offline	Apple

## Channel Policy Overview

### Smart assignment

Tencent Push Notification Service will, based on the device status, segment active status, and push channel status, intelligently assign the optimal delivery channel for each device to achieve the following effects:

1. Improve the overall push reach rate.
2. Speed up the message arrival overall.
3. Save the resources of the vendor channels.

### Custom channel policy



Currently, vendor channels limit the daily number of pushes. You can choose the channels through which this push can be delivered according to business needs and personalize the push channel delivery policy to save vendor channel resources and maximize the value of pushes.

The table below specifies the delivery rules of a custom policy.

Channel	Enabled	Disabled	Supported Messages
Android vendor channels (Huawei, Mi, Meizu, vivo, OPPO, and FCM)	Both the vendor channel and the Tencent Push Notification Service channel are available for this push. Note: If the <b>Tencent Push Notification Service preference</b> is enabled, this push is preferentially delivered through the Tencent Push Notification Service channel when the device is online. Otherwise, the vendor channel will be preferred. If the vendor channel push fails, the Tencent Push Notification Service channel will be used to retry the push. If Tencent Push Notification Service channel is disabled, the push can only be delivered with the vendor channel.	You can disable the Tencent Push Notification Service channel or the vendor channel.	Notification bar messages
iOS vendor channels (APNs)	Both the APNs and Tencent Push Notification Service channels are available for the push. Note: If the <b>Tencent Push Notification Service preference</b> feature is enabled, this push is preferentially delivered through the Tencent Push Notification Service channel when the device is online. Otherwise, the APNs channel is preferred. If the APNs channel push fails, the Tencent Push Notification Service channel will be used to retry the push. If Tencent Push Notification Service channel is disabled, the push can only be delivered with the vendor channel.	You can disable the Tencent Push Notification Service channel or the vendor channel.	Notification bar messages and silent messages. <b>Note:</b> The APNs channel supports delivering up to 3 silent messages to a device per hour.
Tencent Push Notification Service	The Tencent Push Notification Service channel is available for the push.	The Tencent Push Notification Service channel can be disabled.	Notification bar messages, in-app messages, and silent messages. <b>Note:</b> The Tencent Push Notification

Service channel for iOS only takes effect in iOS SDK 1.2.8.0 or later.

## Directions

### Setting the policy in the console

You can select a channel policy for a push in the following path when creating it in the console:

[Tencent Push Notification Service console](#) > **Message Management** > **Task List** > **Create Push** > **Advanced settings** > **Channel Policy**

#### Smart assignment

Select **Smart assignment**. The system will intelligently assign the delivery channel for each device. For more information, see the rules given in [Smart assignment](#).

Channel Policy ⓘ



Smart assignment



Custom

#### Custom policy for Android channels

Select **Custom**. Click **View details** to check vendor-specific quota information.

char	Package Name	Used Today	Last Exceeded Time	Last Exceeded ...
<input type="checkbox"/>	com.tencent.android.duoduo	7	Never	-
<input checked="" type="checkbox"/>	com.tencent.weishi.alpha	Not configured	-	-
<input checked="" type="checkbox"/>	Mi vendor channel (With limit on push quota. <a href="#">View details</a> )			

You can choose the channel used for push according to the remaining quota of the current vendor channels and the priority of the push task. For more information, see the rules given in [Custom channel policy](#). We recommend selecting **It is preferentially delivered through Tencent Push Notification Service channel when the device is online** to save vendor channel resources.

Channel Policy ⓘ

☐ Smart assignment ☒ Custom

Select the available channels to deliver this notifications

☒ TPNS channel (Required)

☐ Huawei vendor channel (This vendor channel has not been enabled for the application. [Enabled Now](#))

☐ Meizu vendor channel (This vendor channel has not been enabled for the application. [Enabled Now](#))

☐ Mi vendor channel (This vendor channel has not been enabled for the application. [Enabled Now](#))

☐ OPPO vendor channel (This vendor channel has not been enabled for the application. [Enabled Now](#))

☐ Vivo vendor channel (This vendor channel has not been enabled for the application. [Enabled Now](#))

☒ It is preferentially delivered through TPNS channel when the device is online

**Caution:**

The Tencent Push Notification Service channel can be disabled.

**Custom policy for iOS channels**

You can choose the channel used for push based on the priority of the push task. For more information, see the rules given in [Custom channel policy](#). We recommend selecting **It is preferentially delivered through Tencent Push Notification Service channel when the device is online** to ensure the fastest notification delivery to the device.

Channel Policy ⓘ

☐ Smart assignment
☒ Custom

Select the available channels to deliver this notifications

☒ TPNS channel (Required)
☒ APNs channel (Apple system-level channel, recommended to enable)

☒ It is preferentially delivered through TPNS channel when the device is online

Note:

- When the APNs channel fails to be issued, the system automatically supplements the push through the TPNS channel

TPNS notification channel only takes effect in iOS SDK 1.2.8.0 and above

## Setting the policy with RESTful APIs

Set the optional `channel_rules` parameter for the RESTful API. For more information, please see [channel\\_rules parameter description](#) in the Push API documentation.

Below is a sample push on Android:

```
{
  "audience_type": "token",
  "token_list": [
    "05da87c0ae*****8d884aada5bb2"
  ],
  "message_type": "notify",
  "channel_rules": [
    {
      "channel": "mz",
      "disable": true //Disable the Meizu channel
    },
    {
      "channel": "xm",
      "disable": false //Enable the Mi channel
    }
  ],
  "tpns_online_push_type": 0, //The push will be delivered through the Tencent Push
  "message": {
    "title": "The push will be delivered through the Mi channel, and the Meizu",
    "content": "Push content",
    "android": {
      "custom_content": "{\\"key\\":\\"value\\"}"
    }
  }
}
```

```
}  
}
```

Below is a sample push on iOS:

```
{  
  "audience_type": "token",  
  "environment": "dev",  
  "token_list": ["05da87c0ae*****fa9e08d884aada5bb2"],  
  "message_type": "notify",  
  "channel_rules": [{  
    "channel": "apns",  
    "disable": true  
  }],  
  "tpns_online_push_type": 0,  
  "message": {  
    "title": "The push will be delivered through the Tencent Push  
Notification Service channel only",  
    "content": "Push content",  
    "ios": {  
      "aps": {  
        "alert": {  
          "subtitle": "Push subtitle"  
        },  
        "badge_type": -2,  
        "sound": "Tassel.wav"  
      },  
      "custom_content": "{\\"key\\":\\"value\\"}"  
    }  
  }  
}
```

# Message Collapse

Last updated : 2024-01-16 17:34:39

## Overview

If a mobile phone receives multiple messages from the same application, these messages will be collapsed in the notification center according to the corresponding rule to avoid disturbing the user. However, this mechanism will reduce the exposure of a push and thus lower the operating profits.

Both the native Android and iOS systems provide the corresponding settings. You can use the message collapse feature to collapse messages of the same type based on your operation needs so that the pushed messages can be read more easily.

## Application Scope

Mobile Phone OS	OS Version	SDK Version	Push Channel
Android	Android 7.0 or above	TPNS SDK v1.2.0.1 or above	Tencent Push Notification Service channel
iOS	iOS 10 or above	TPNS SDK v1.2.0.1 or above	APNs channel

## Directions

### Using the console

1. Log in to the [Tencent Push Notification Service console](#). In the left sidebar, click **Message Management** > **Task List**. Then, click **Create Push** > **Advanced Settings**.

### Set Up Push Task

Push Platform

Test ▼

Test-Android-long-name-test ▼

Push Plan ⓘ

Choose Push Plan ▼

+Create New Plan

Push Type

Notification bar message

In-app message

Notification Title

Enter notification title

Notification Content

Enter notification content

Push Time

Immediate

Scheduled

Loop

Push Target

Please select ▼

Advanced Settings ▼

2. Set a message collapse rule, as shown in the following figure:

Grouping and Collapsing ⓘ

System Default

Uncollapsed

Custom

The following three options are provided for you to set whether and how to collapse messages in the notification center:

System Default: uses the default collapse rule of the system.

The following are system default rules of some vendors:

Native Android: If there are 5 or more messages of the same application, these messages will be collapsed into a group. The number on the right of the collapsed messages is the number of unread messages. When they are expanded, up to 8 messages can be displayed.

Huawei: If there are 2 or more messages of the same application, these messages will be collapsed into one group. The number on the right of the collapsed messages is the number of unread messages. When they are expanded, up to 8 messages can be displayed.

Mi: If there are 4 or more messages of the same application, these messages will be collapsed into one group. The number on the right of the collapsed messages is the number of unread messages (up to 7). When they are expanded, up to 10 messages can be displayed.

Meizu: If there are 4 or more messages of the same application, these messages will be collapsed into one group. The number at the top of the collapsed messages is the total number of messages. When they are expanded, up to 35 messages can be displayed.

**Note:**

Meizu phones provide the unimportant notification feature, which allows users to place excessive notifications in the notification drawer in the upper-right corner, where the notifications will not be collapsed. You can enable "Priority display" in the application notification settings to disable this feature.

OPPO: If there are 4 or more messages of the same application, these messages will be collapsed into one group. The number on the right of the collapsed messages is the number of unread messages. When they are expanded, up to 8 notifications can be displayed.

Vivo: If there are 2 or more messages of the same application, these messages will be collapsed into one group. The number of collapsed messages will not be displayed. When they are expanded, up to 8 notifications can be displayed.

Uncollapsed: The message will not be collapsed with other messages of the same application.

Custom: Messages with the same `thread_id` will be collapsed.

**Note:**

You can set this parameter for low-priority messages to collapse them. On the contrary, you can set **Uncollapsed** for important messages.

## Using RESTful APIs

If you want to implement the "Uncollapsed" and "Custom" effects of the console, you need to customize

`thread_id` in the `message` field of the `Push` API. For more information, please see [Push API - Optional Parameters](#).

Below is a sample push:

```
{
  "audience_type": "token",
  "token_list": [
    "05da87c0ae*****2dfa9e08d884aada5bb2"
  ],
  "message_type": "notify",
  "multi_pkg": true,
  "message": {
    "Title": "Push title",
    "content": "Push content",
    "android": {
      "custom_content": "{\\"key\\":\\"value\\"}"
    },
    "thread_id": "Activity_id",
  }
}
```



```
    "thread_sumtext": "Operation activity"  
  
  }  
}
```

# Rich Media Notification

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

## Overview

Rich media push allows you to push rich media contents such as image, audio, and video in addition to text, which can effectively increase the notification click rate. You can use this feature to push a wider variety of contents such as news, coupons, and event information, satisfying your personalized push needs.

## Application Scope

Currently, Tencent Push Notification Service rich media push supports the following types of rich media contents:

Android: image and audio

iOS: image, audio, and video

**Note:**

The Tencent Push Notification Service channel for iOS supports rich media push by default. For the APNs channel, applications need to integrate with the [notification service extension plugin](#).

The supported types of rich media and use requirements for each push channel are as follows:

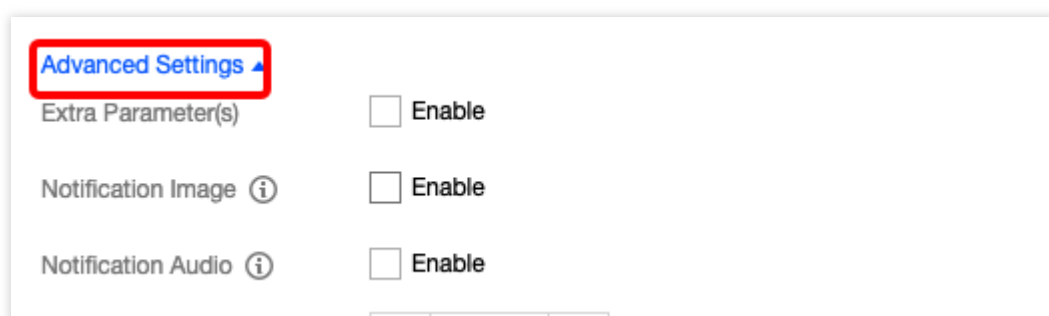
Push Channel	Supported Type	Use Requirements
Huawei	Thumbnail	Only HTTPS URLs are supported. Format requirements: PNG , JPG , and JPEG 120 x 120 px. The image will be automatically resized if its width or height exceeds 120 px. Less than 200 KB.
Mi	Large image	Only HTTPS URLs are supported. Format requirements: PNG , JPG , and JPEG Fixed at 876 x 324 px. Less than 1 MB. <b>Note:</b> to use the large image notification feature in the Mi channel, you need to first call the image upload API of Mi to upload the image file, get the image address <code>pic_url</code> specified by Mi, and enter it in the corresponding Tencent Push Notification Service push parameter <code>xg_media_resources</code> . For more information, see the large image upload API description in <a href="#">here</a> .

Tencent Push Notification Service	Large image, thumbnail, and audio	<p>Android platform:</p> <p>Only HTTPS URLs are supported.</p> <p>Image format requirements:</p> <p>JPEG , JPG , and PNG .</p> <p>The height of a large image cannot exceed 324 px and the width is adaptive. The size of a thumbnail should be 120 x 120 px. If its height or width exceeds 120 px, it will be cropped into a square.</p> <p>Audio file format requirements: an audio file cannot exceed 5 MB.</p> <p>iOS platform: the requirements are the same as that of the APNs channel.</p>
APNs	Thumbnail and audio/video	<p>Only HTTPS URLs are supported.</p> <p>Image format requirements:</p> <p>JPEG , PNG , and GIF formats.</p> <p>Less than 10 MB.</p> <p>Audio/Video file format requirements:</p> <p>Video encoding formats: MPEG , MPEG-2 Video , MPEG-4 , and AVI</p> <p>Audio file extensions: AIFF , WAV , and CAF</p> <p>Audio encoding formats: Linear PCM , MA4 (IMA/ADPCM) , alaw , and μLaw</p> <p>An audio/video file should be less than 5 MB.</p> <p>An audio file should be less than 30s long.</p> <p><b>Note:</b> image and audio/video cannot be enabled at the same time.</p>

## Operation Directions

### Console

1. Log in to the [Tencent Push Notification Service console](#). Go to **Message Management > Task List**.
2. Click **Create Push > Advanced Settings**.



3. Enable **Notification Image** or **Notification Audio/Video**, and enter the rich media message URL. The detailed configuration instructions are as follows:

When you enable **Notification Image**:

Android:

**Note:**

Thumbnail:

The URL must be in HTTPS format.

The thumbnail can be displayed in pushes delivered through the Tencent Push Notification Service and Huawei channels.

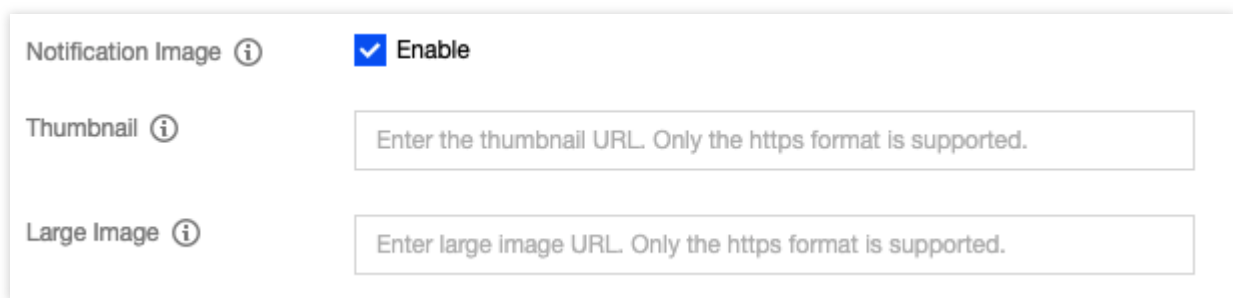
Format requirements: PNG , JPG , or JPEG ; 120 × 120 px; less than 200 KB

Large image:

The URL must be in HTTPS format.

The large image can be displayed in pushes delivered through the Tencent Push Notification Service and Mi channels.

Format requirements: PNG , JPG , or JPEG ; 876 x 324 px; less than 1 MB.



Notification Image ⓘ ☒ Enable

Thumbnail ⓘ

Large Image ⓘ

iOS:

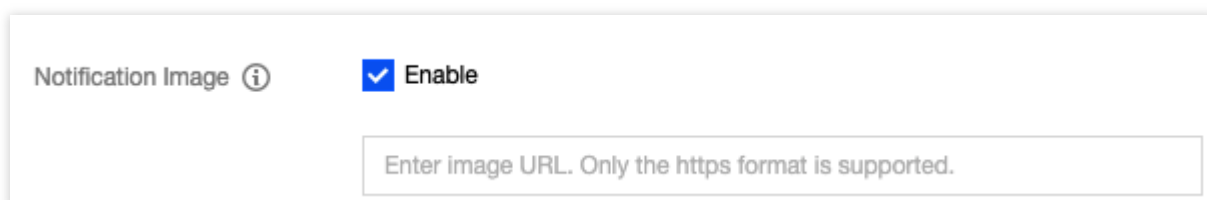
**Note:**

After the image URL is entered, the image will be displayed in the notification. Format requirements:

The file size cannot exceed 10 MB.

The file must be in PNG , JPG , JPEG , or GIF format.

The URL must be in HTTPS format.



Notification Image ⓘ ☒ Enable

When you enable **Notification Audio** or **Notification Audio/Video**:

Android:

**Note:**

After the audio URL is entered, notifications delivered through the Tencent Push Notification Service channel can contain audio.

The audio file size cannot exceed 5 MB.

The URL must be in HTTPS format.

Notification Audio ⓘ

☒ Enable

Enter audio URL. Only the https format is supported.

iOS:

#### Note:

After the audio/video URL is entered, notifications can contain audio or video and be played back through Apple's native component.

The audio/video file size cannot exceed 5 MB.

The file must be in MPEG , MPEG-2 Video , MPEG-4 , or AVI format.

The URL must be in HTTPS format.

Notification Audio and Video ⓘ

☒ Enable

Enter audio or video URL. Only the https format is supported.

## RESTful API

To send a rich media message via an API call, you can set the following parameters in the Android or iOS message body in the `Push` API:

Platform	Parameter	Notes
Android	Thumbnail: icon_res, icon_type Large image: xg_media_resources Audio: xg_media_audio_resources	Thumbnails are supported only for the Tencent Push Notification Service and Huawei channels. Large images are supported only for the Tencent Push Notification Service and Mi channels. Audios are supported only for the Tencent Push Notification Service channel.
iOS	Image and audio/video: xg_media_resources	Notification Image and Notification Audio/Video cannot be enabled at the same time.

For more information, see the **Required Parameters** section in [Push API](#).

**Below is a sample push on Android:**

```
{
  "audience_type": "token",
  "token_list": [
    "05da87c0ae*****2dfa9e08d884aada5bb2"
  ],
  "message_type": "notify",
  "multi_pkg": true,
  "message": {
    "Title": "Push title",
    "content": "Push content",
    "xg_media_resources": "xxx1" , // Enter the URL of the rich media
    element, such as `https://www.xx.com/img/bd_logo1.png?qua=high`.
    "xg_media_audio_resources": "xxx", // Enter the URL of the audio, such
    as `https://sc1.111ttt.cn/2018/1/03/13/396131227447.mp3`.
    "android": {
      "icon_type": 1,
      "icon_res": "xxx", // Enter the URL of the thumbnail.
      "custom_content": "{\\"key\\":\\"value\\"}"
    }
  }
}
```

## FAQs

### **The Meizu, OPPO, and vivo channels do not support rich media. How do I deliver notifications to all devices?**

Notifications delivered through the Tencent Push Notification Service, Huawei, Mi, and FCM channels contain images, while those delivered through the Meizu, OPPO, and vivo channels are in plain text format without images by default.

### **How can I include an image in a push message?**

Whether you call an API or use the console to deliver a message, you need to generate a URL for the image to include it in the message.

### **What is the policy for delivering audios on Android?**

Audio pushes delivered through the Tencent Push Notification Service channel can display the audios normally, while those delivered through other channels will be in plain text format without audios by default.

### **Can I enable both Notification Image and Notification Audio/Video for iOS?**

No. You can only enable either of them.

# Custom Notification

Last updated : 2024-01-16 17:34:39

## Overview

Tencent Push Notification Service supports adding custom parameters to the push text. After you bind custom parameters to the device and create a push, the device will display the message with custom parameters. This makes the push more attractive, so users are more likely to click on it.

## Use Cases

### Ecommerce

To increase the payment rate for items in cart, you can use the following template:

```
Hi, @{{nickname}}, there are only {{productnum}} left in stock for the
{{productname}} in your cart.
Wanna buy it now?
```

**The push message Tommy receives is as follows:**

```
Hi, @Tommy, there are only 6 left in stock for the penguin doll in your cart. Wanna
```

### Gaming

To reactivate inactive gamers through push, you can use the following template:

```
Hi, @{{nickname}}, you have not logged in to the game for {{offline_days}}
days. We have prepared {{gift_num}} gift packages for you.
Come claim them. >>>
```

**The push message Tommy (who has been inactive for 3 consecutive days) receives is as follows:**

```
Hi, @Tommy, you have not logged in to the game for 3 days. We have prepared 6 gift
```

### Social networking

To reactivate users who have not opened the application for 3 consecutive days through push, you can use the following template:



```
Hi, @{{nickname}}, {{friend_num}} friends of yours posted {{story_num}} updates
while you were away.
Come check them out. >>>
```

The push message Tommy receives is as follows:

```
Hi, @Tommy, 8 friends of yours posted 20 updates while you were away. Come check th
```

## Prerequisites

### Creating and managing user attributes

1. Log in to the [Tencent Push Notification Service console](#).
2. Go to **Message Management > Attributes Management** and click **Add User Attribute**.
3. In the **Add User Attribute** dialog box, enter the attribute name and description, and click **OK**. Then you can view the created time, attribute name, attribute description and number of devices on the **User Attribute Management** page. You can edit or delete an attribute at any time.

**User attributes:** A unified identification of a certain type of user information. Such as: user nickname, city where user resides, etc.

**Use scenario:** When the notification bar message is pushed, user attributes can be added to the push copy to enhance push personality and increase click rate.

**Operation steps:** ① Add user attributes; ② Call client/server API to bind user information to the device; ③ Add user attributes to the copy when creating a push ;

**Help Documentation:** [Personalized push](#)

Add User Attribute

Search

Created Time	Attribute Name	Attribute Description	Number of Devices	Operation
2020-10-22 17:51:56	nickname	usernickname	0	<a href="#">Edit</a> <a href="#">Delete</a>

Total items: 1

10 / page

1

/ 1 page

### Binding user attributes

Before pushing a custom message, you need to bind the user attributes to devices using either of the following methods:

#### Method 1: using client APIs:

For the iOS SDK, see [here](#).

For the Android SDK, see [here](#).

## Method 2: using the RESTful API

To bind user attributes via the RESTful API, see [here](#).

## Directions

### Setting the policy in the console

1. Go to **Message Management > Task List** and click **Create Push**.
2. Insert the user attributes on the right of the **Notification Title** or **Notification Content** field.

#### Note:

One push supports adding up to 5 attributes at a time.

**Set Up Push Task** Pre

Push Platform: Test Test-Android-long-name-test [Cross Platform Push](#)

Push Plan ⓘ: Choose Push Plan [+Create New Plan](#)

Push Type: Notification bar message In-app message

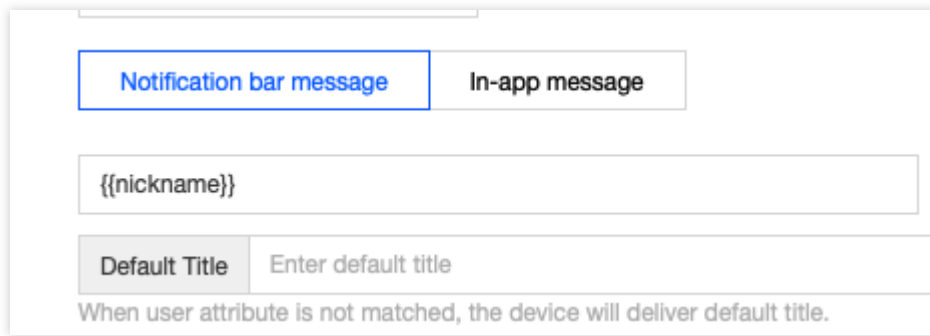
Notification Title: Enter notification title

Notification Content: Enter notification content

**User Attribute Selection (highlighted in red box):**

- nickname(0)
- username
- [User Attribute Management](#)

3. Set to deliver the default notification or content if no user attribute is matched.



4. Click **Test Preview**, double check the information, and click **Confirm**.

## Setting the policy with RESTful APIs

To enable the custom notification via the API, set `ntf_wt_attrs` to `true` and add the following fields to `message`.

Parameter	Type	Required	Description
<code>default_content</code>	string	Yes	The default message content will be sent to devices if no user attribute is matched.
<code>default_title</code>	string	<code>Yes</code> for Android, and <code>No</code> for iOS	The default message title will be sent to devices if no user attribute is matched.
<code>default_subtitle</code>	string	No	The default message subtitle will be sent to devices if no user attribute is matched.

For more information about other message fields, see the “message: message body” section in [Push API](#).

Sample push:

```
{
  "audience_type": "token",
  "expire_time": 3600,
  "message_type": "notify",
  "environment": "dev",
  "message": {
    "title": "Hi, {{name}}",
    "content": "You have earned {{score}} points",
    "default_content": "Default content",
    "default_title": "Default title",
    "default_subtitle": "Default subtitle"
  },
  "token_list": [
    "086f959c7aefc3****add2ccf0cd539c1edd"
  ],
  "ntf_wt_attrs": true
}
```

```
}
```

# Push to Accounts

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

## Overview

After you get a batch of user accounts from a third-party data platform or business backend, when launching a marketing campaign for such users, you can use the **push to accounts** feature to push messages to a single or multiple accounts at a time.

### Note:

The abovementioned **accounts** must be bound with a Tencent Push Notification Service token. For detailed directions, please see [Binding an account \(Android\)](#) or [Adding account \(iOS\)](#).

## Directions

### Using the console

1. Log in to the [Tencent Push Notification Service console](#).
2. Find the application for which to configure batch push and select **Create Push** in its **Operation** column to enter the **Create Push** page.
3. In the **Push Target** field, select **Account** and upload an account package file or manually enter accounts.

### Note:

Requirements for the uploaded account package:

Account package filename: [1, 100] characters.

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

`.zip` file requirements: can contain a single `.txt` or `.csv` file 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.

4. Select the account type. You can obtain the account type from service developers. If no account type is specified, the default type is used.
5. Click **Preview**. After confirming that the push configuration is correct, click **Confirm**.

### Using RESTful APIs

#### Push to a single or multiple accounts

When you call the [push API](#), set `audience_type` (push target) to `account` (single account) or `account_list` (a list of accounts) and enter a proper account type as instructed in [Account Type Value Table](#).

## Sample push

```
{
  "audience_type": "account",
  "account_list": [
    "123456"
  ],
  "account_type": 1,
  "account_push_type": 0,
  "message_type": "notify",
  "message": {
    "title": "Congrats on winning in the campaign",
    "content": "Get online to claim your prize!"
  }
}
```

## Uploading an account package file for push

### Step 1. Call the API for uploading an account package file

Upload your account package file as instructed in [Account Package Upload API](#). After the call succeeds, an

`upload_id` will be returned, such as `11231`.

### Step 2. Call the push API

1. When you call the [push API](#), set `audience_type` (push target) to `package_account_push` (push to accounts in the package).
2. Enter the `upload_id` obtained in [Step 1](#), such as `11231`.
3. Enter a proper account type as instructed in [Account Type Value Table](#).
4. Set `account_push_type` to specify whether to push to the **recent** or **all** devices bound to each account.

## Sample push

The following sample pushes a message to users who have won prizes in a marketing campaign:

```
{
  "audience_type": "package_account_push",
  "upload_id": 11231,
  "account_type": 1,
  "account_push_type": 0,
  "message_type": "notify",
  "message": {
    "title": "Congrats on winning in the campaign",
    "content": "Get online to claim your prize!"
  }
}
```