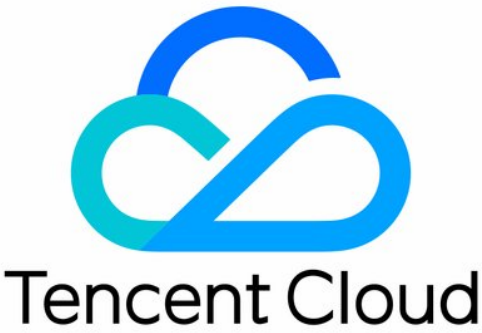
Tencent Push Notification Service

Getting Started

Product Documentation



Contents

Getting Started
Creating Products and Applications

Last updated：2024-01-16 17:31:40

Overview

This document shows you how to create products and applications in the Tencent Push Notification Service console and how to configure applications.

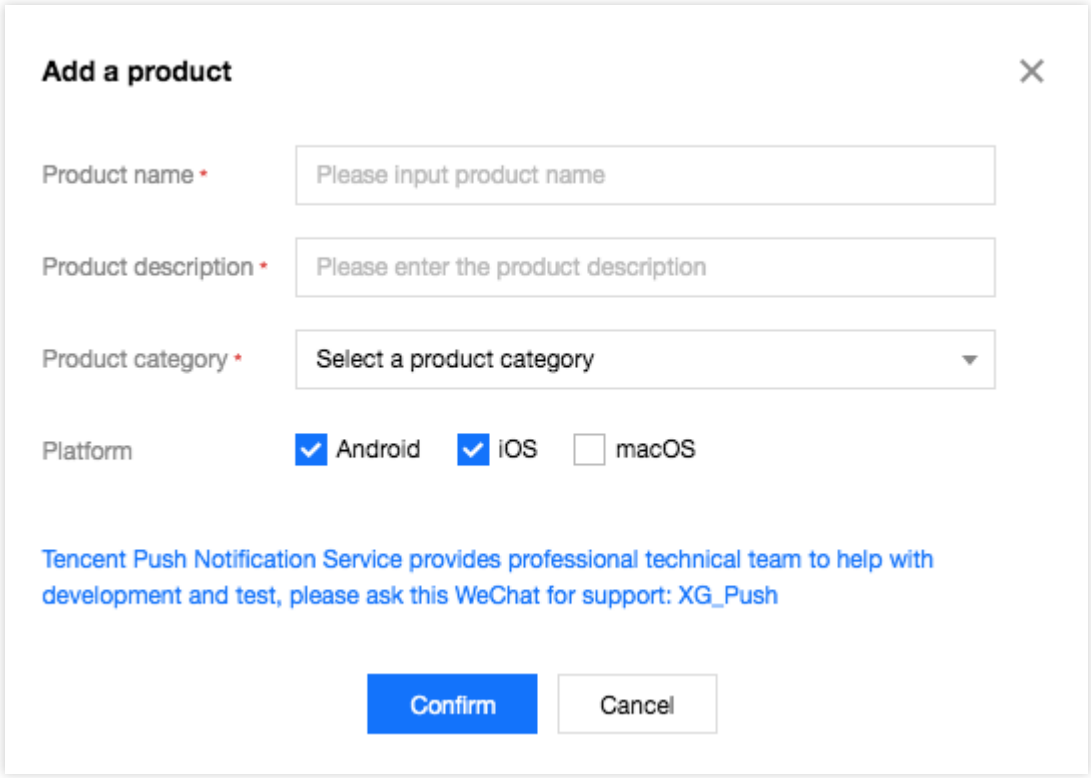
Prerequisites

You have a Tencent Cloud account. For details, see the Sign Up for a Tencent Cloud Account tutorial.

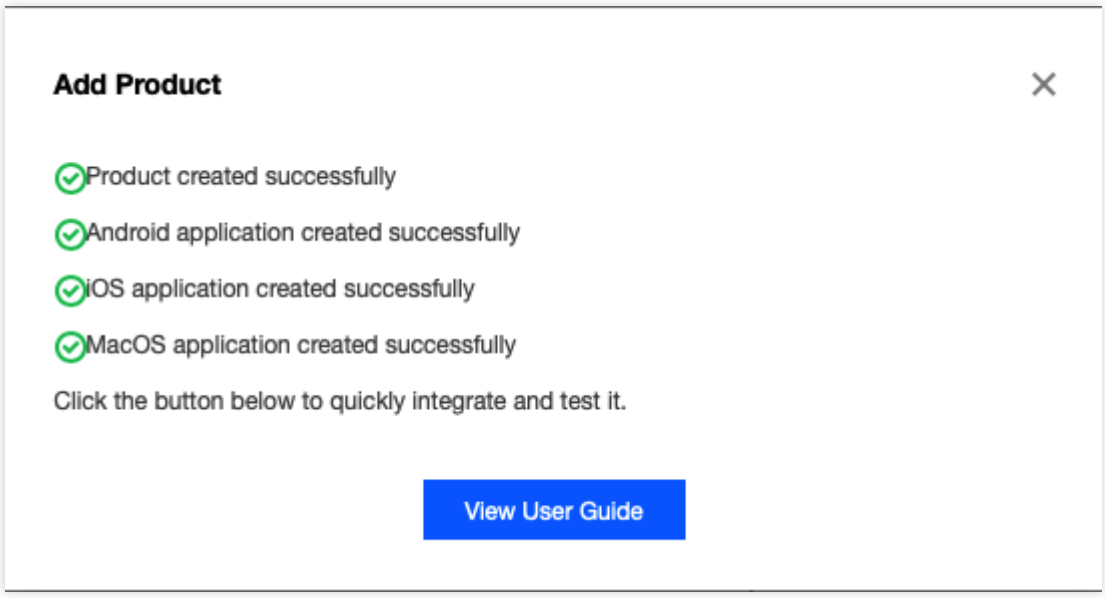
Directions

Creating a product

1. Log in to the Tencent Push Notification Service console, and click **Product Management** in the left sidebar.
2. On the **Product Management** page, click **Add Product**.
3. In the pop-up window, enter the product name and product description, and select a product category and service access point. For more information on service access points, see Global Deployment.
4. When you put a check mark next to Android, iOS, or macOS below, the system will by default create applications for the selected platform.



5. Click **OK** to complete the product creation, and click **View User Guide** to integrate the product as instructed.

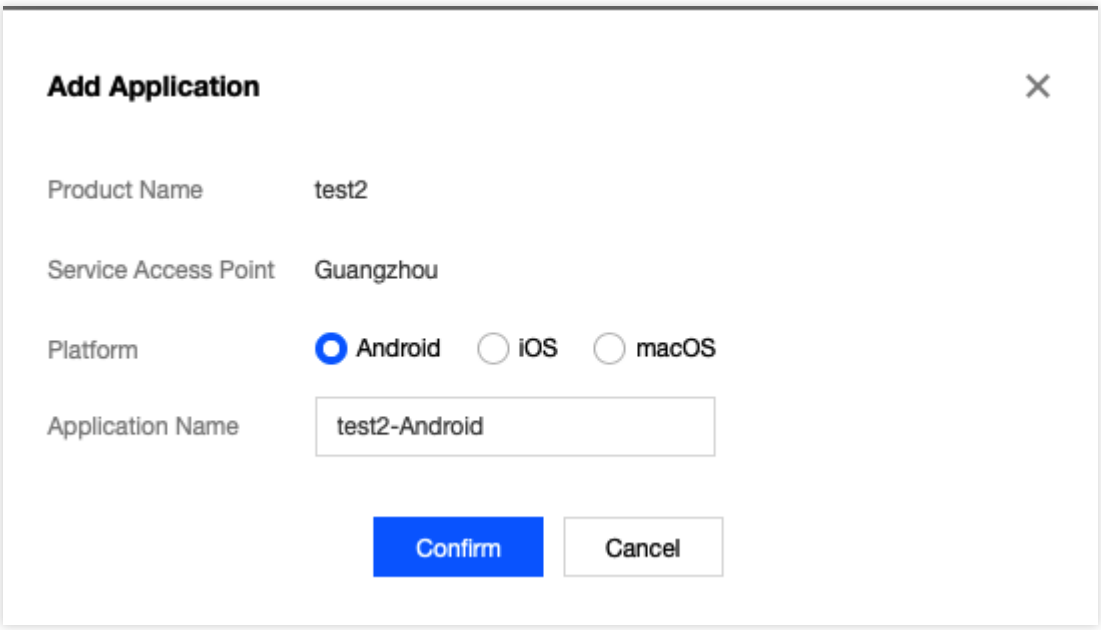


Creating an application

After you create a product, if you have not yet added a default application, you can follow these instructions to create an application. Only one application can be created per platform. After applications have been created for all three platforms, no new applications can be created.

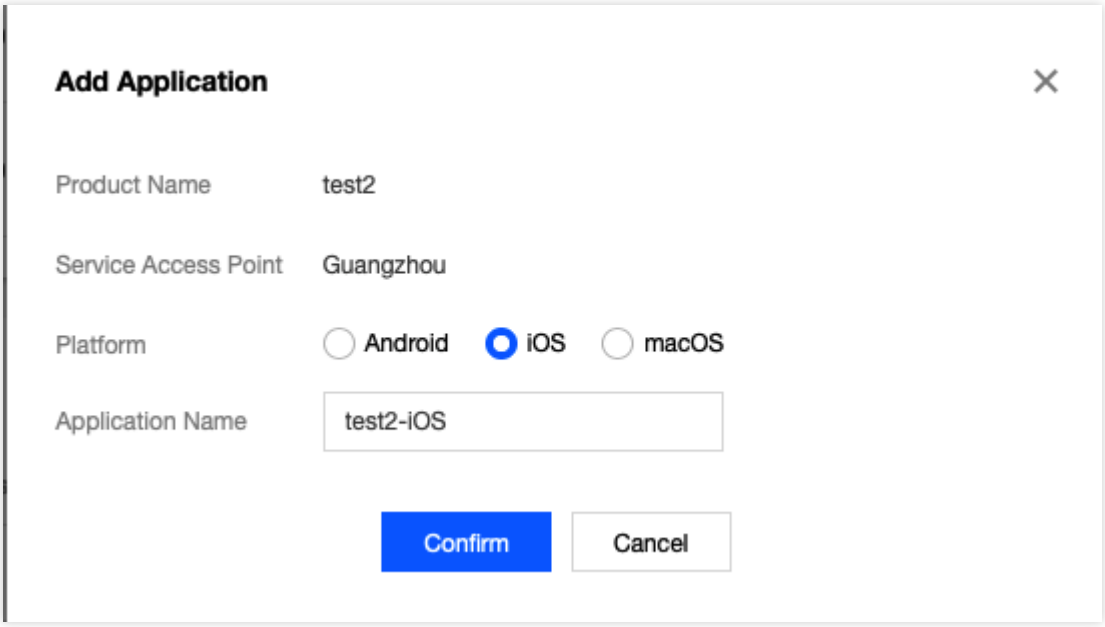
Android applications

1. Log in to the Tencent Push Notification Service console, and click **Product Management** in the left sidebar.
2. On the product list page, select the created product, click **Add Application**, and select **Android** as the platform.
3. Enter the application name, and click **OK** to create the application.



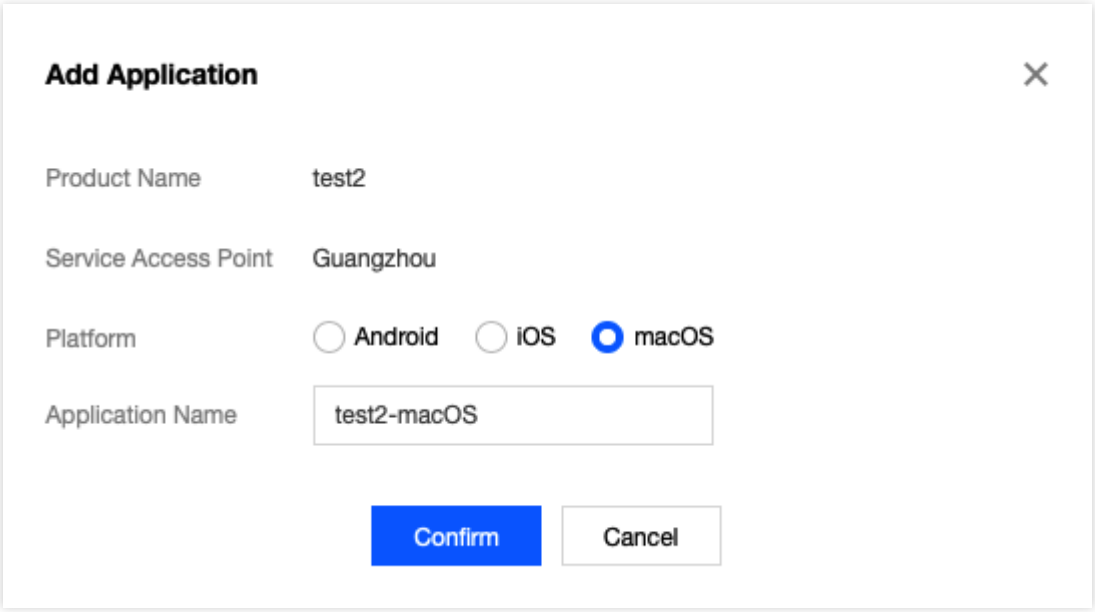
iOS applications

1. Log in to the Tencent Push Notification Service console, and click **Product Management** in the left sidebar.
2. On the product list page, select the created product, click **Add Application**, and select **iOS** as the platform.
3. Enter the application name, and click **OK** to create the application.



macOS applications

1. Log in to the Tencent Push Notification Service console, and click **Product Management** in the left sidebar.
2. On the product list page, select the created product, click **Add Application**, and select **macOS** as the platform.
3. Enter the application name, and click **OK** to create the application.



Configuring applications

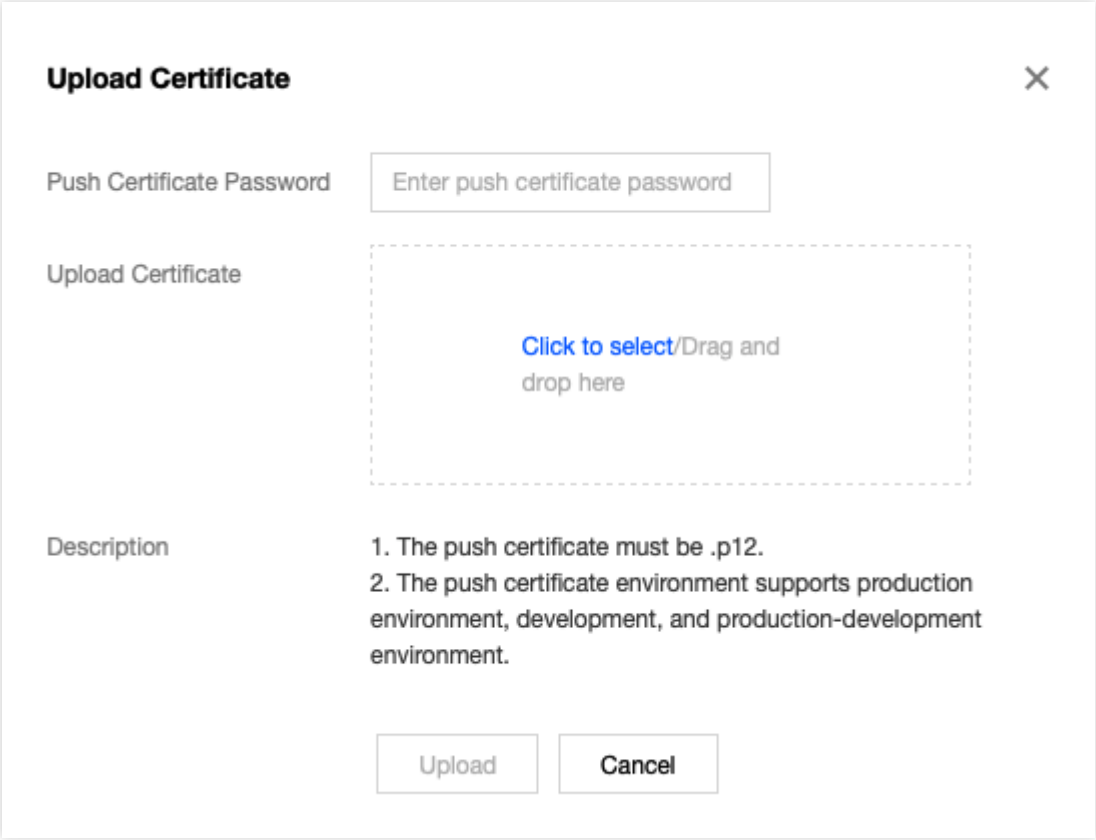
After you create an application, you can follow the steps below to configure it.

Android

1. Go to the **Product List** page, select Android platform applications, and click **Configuration Management**.
2. Enter the name of the Android platform application, and click **Save** to complete basic configuration.
3. Vendor channel configurations can be enabled according to your requirements.

iOS

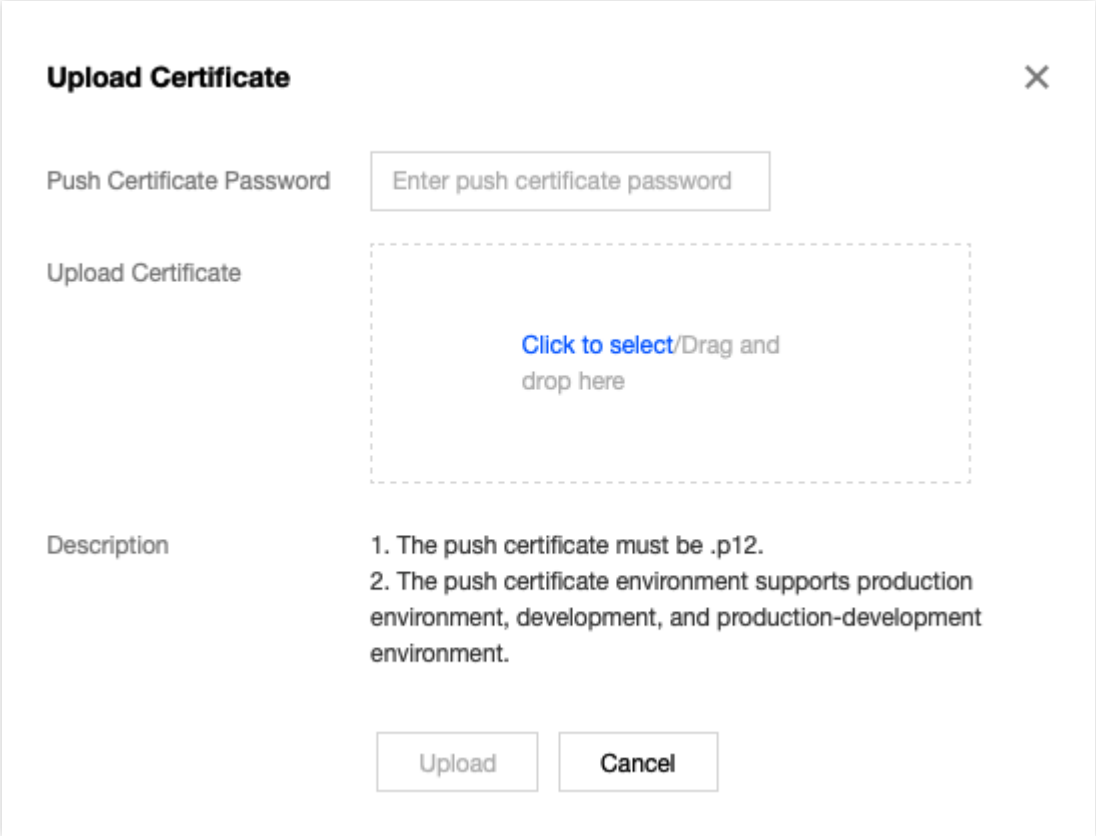
1. Go to the **Product List** page, select iOS platform applications, and click **Configuration Management**.
2. Enter the BundleID for the iOS platform, and click **Save** to complete basic configuration.
3. Go to the configuration management page, click **Upload Certificate**, enter the certificate password and select a certificate to upload.
4. Click **Submit** to upload your iOS push certificate to the management platform, and complete iOS application configuration.



macOS configuration

1. Go to the **Product List** page, select macOS platform applications, and click **Configuration Management**.
2. Enter the BundleID for the macOS platform, and click **Save** to complete basic configuration.
3. Go to the configuration management page, click **Upload Certificate**, enter the certificate password and select a certificate to upload.

4. Click **Submit** to upload your iOS push certificate to the management platform, and complete iOS application configuration.



Obtaining the application information in the console

After successful configuration, you can obtain the application's **AccessID**, **AccessKey**, and **SecretKey**, which are described as follows.

AccessID: Unique identifier of the application. Use cases:

1. SDK integration
2. Authentication signature generation during a RESTful API call

AccessKey: Client authentication key of the application. Use case:

SDK Integration

SecretKey: Server authentication key of the application. Use case:

Authentication signature generation during a RESTful API call

Quick Integration with Android

Last updated：2024-01-16 17:31:40

Overview

This document describes how to quickly integrate the Tencent Push Notification Service SDK into your Android application. You can use the Tencent Push Notification Service on your application after performing the following steps.

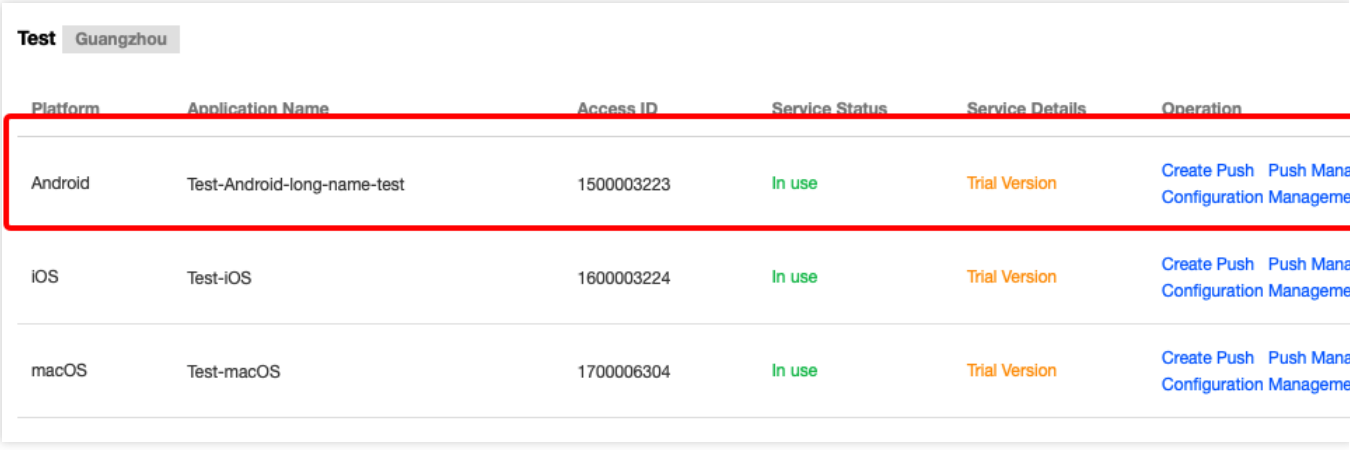
Note:

To avoid your app from being criticized in a circulated notice or removed from the market by regulatory authorities, be sure to add mobile push related instructions in the Privacy Policy according to the Android Compliance Guide before integrating the Tencent Push Notification Service SDK and initialize the SDK after the user agrees to the Privacy Policy.

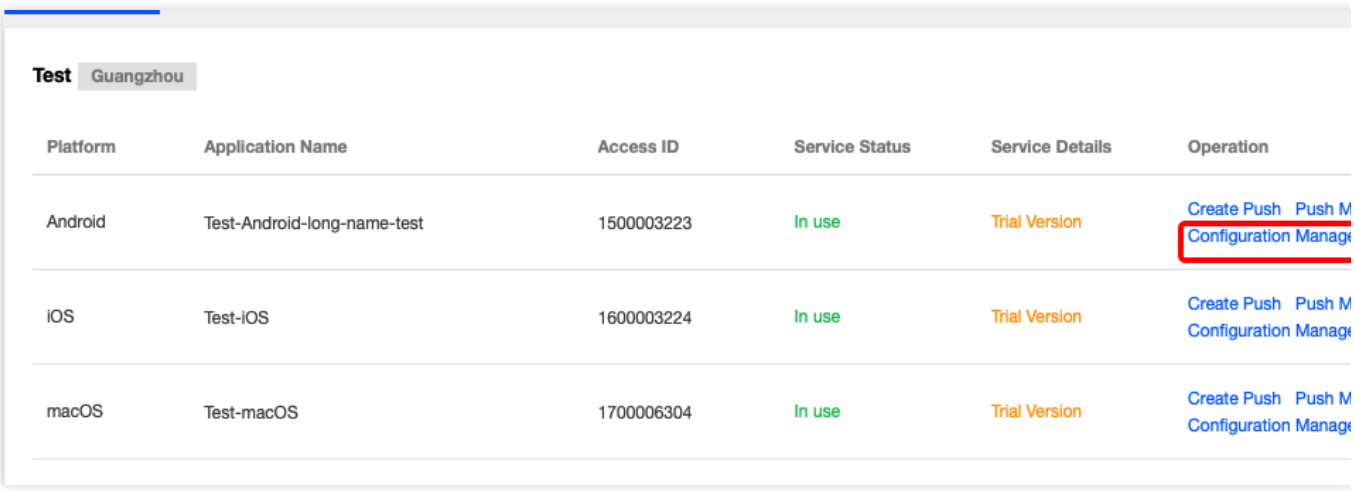
Preparing for Integration

Creating an Android platform application

1. Before integrating the SDK, you need to log in to the Tencent Push Notification Service console and create the product and Android application. For detailed directions, please see Creating Products and Applications.

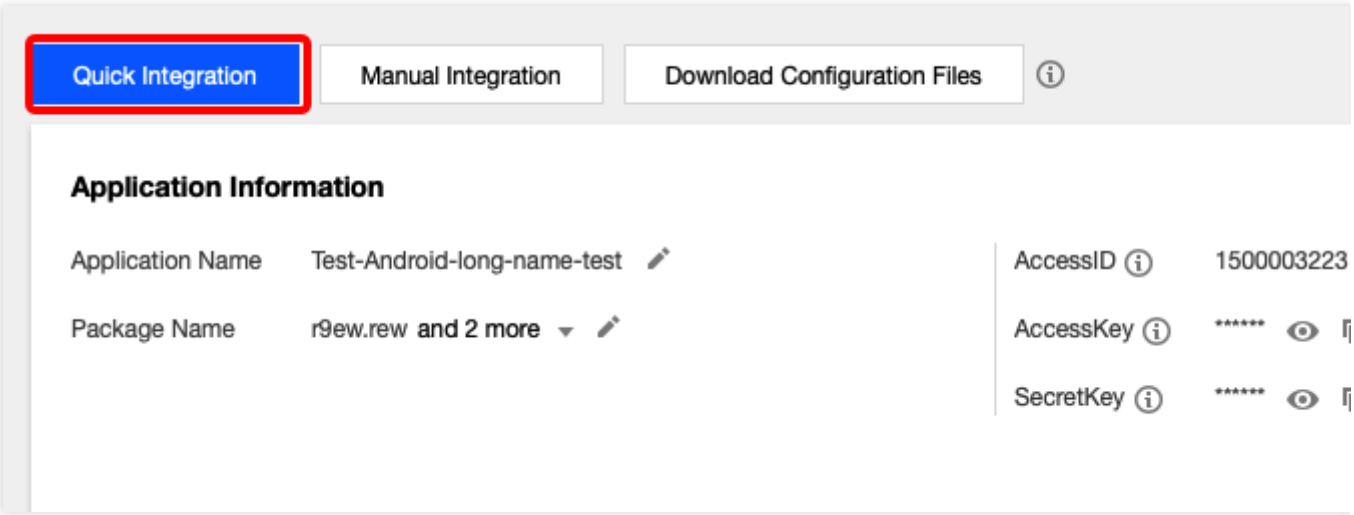


2. Go to the **Configuration Management** page of the application to prepare for the integration.

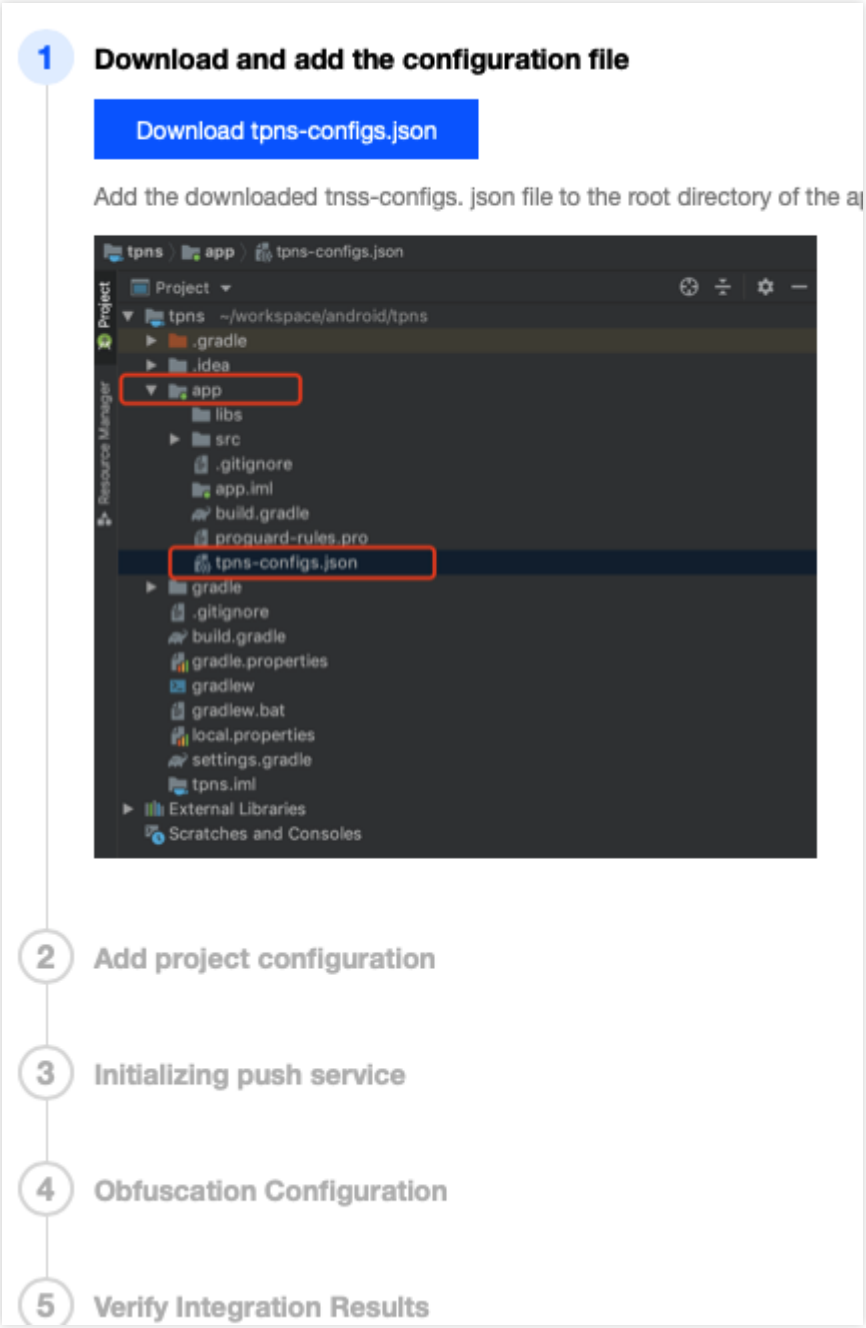


Starting Integration

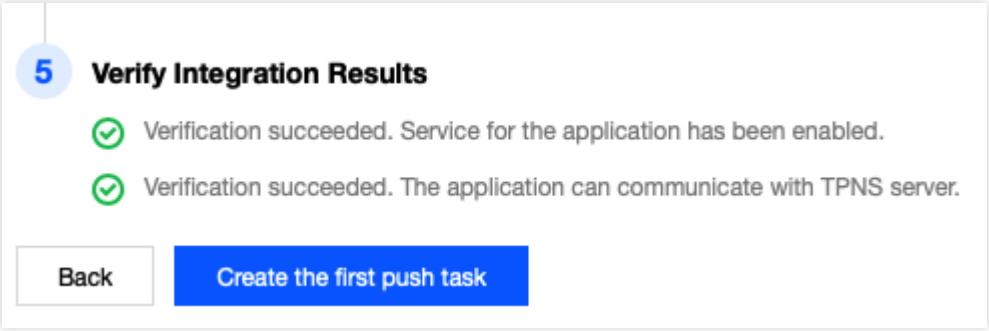
1. On the **Configuration Management** page, click **Quick Integration**.



2. Complete the configuration as instructed and click **Click to verify**.



3. If the following prompt is displayed, the Tencent Push Notification Service SDK is successfully integrated.



If verification failure information as shown in the figure below is reported, check whether the application has been successfully registered with the Tencent Push Notification Service as instructed in Verifying the Integration Result.

Caution:

To increase the offline reach rate, the Tencent Push Notification Service SDK enables the session keep-alive feature by default. To disable the feature, see here.

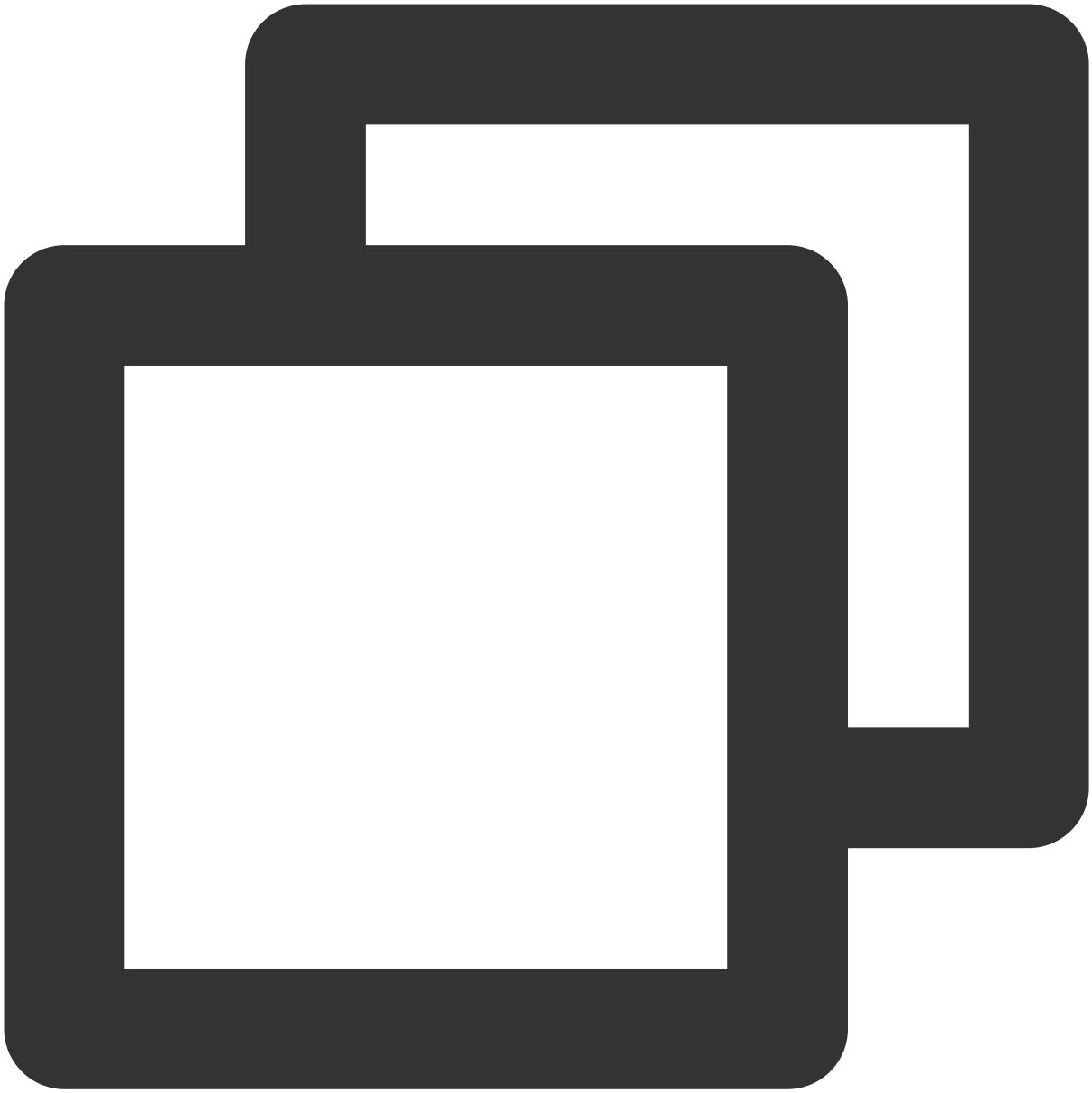
Verifying the Integration Result

1. Run the application, filter logs by the "TPush" keyword, and view the displayed logs.

```
I TPush    : [XGVipPushService] Service onCreate() : com.example.tpns
I TPush    : [XGVipPushService] Service onBind()
I TPush    : [XGVipPushService] Service onStartCommand() : com.example.tpns
D TPush    : [PushServiceManager] Service's running at com.example.tpns,version : 1.1.5.2
```

If a log similar to that shown in the preceding figure is displayed, the Tencent Push Notification Service SDK has been successfully integrated as a plugin.

2. Check whether the registration with the push service is successful. The following log indicates successful registration.

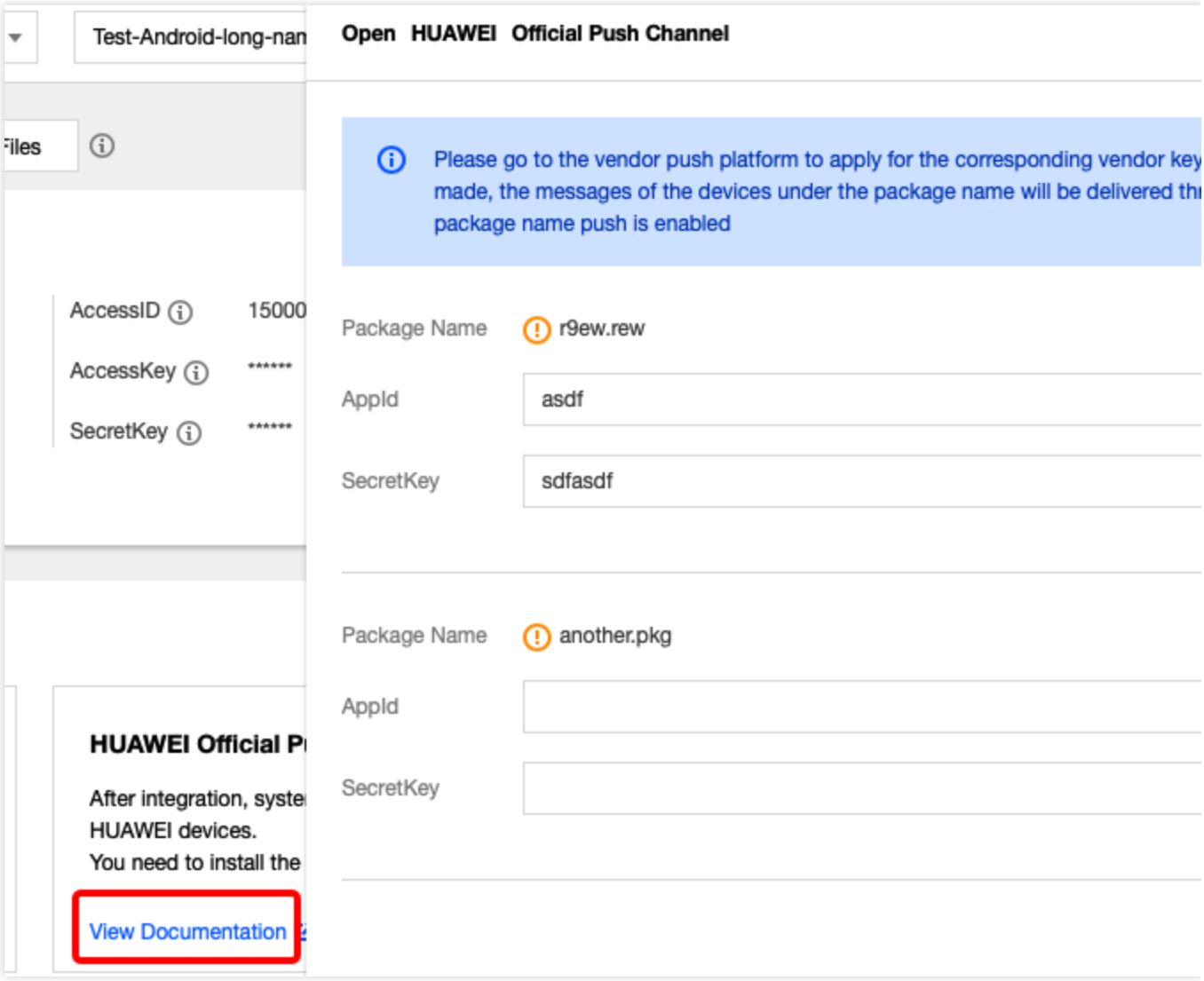


```
XG register push success with token:6ed8af8d7b18049d9fed116a9db9c71ab4aabb65
```

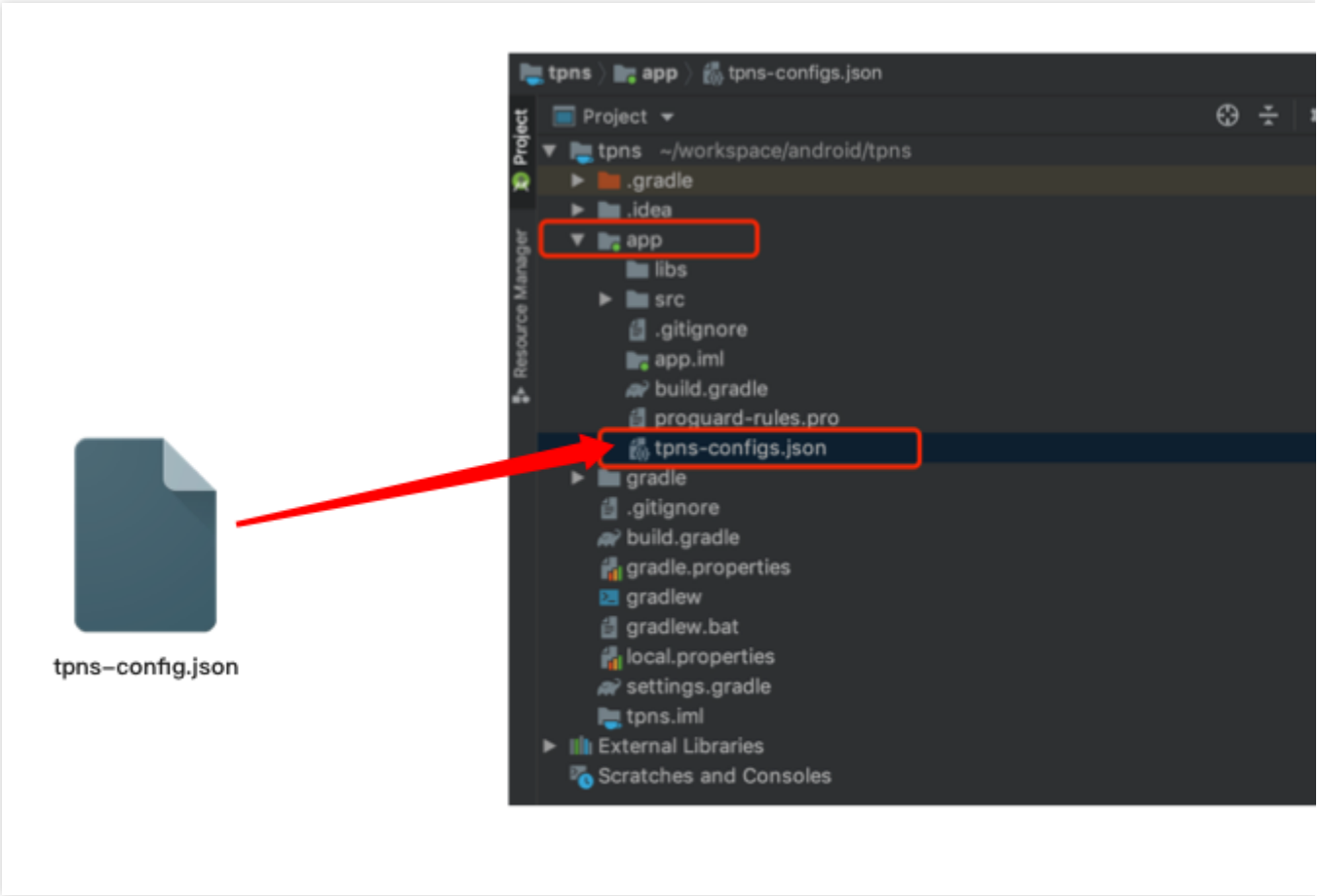
If the token cannot be found, please check the error code returned by the registration API and troubleshoot as instructed in Error Code.

Quickly Integrating with a Vendor Channel

1. On the **Configuration Management** page, enable the vendor push channel and configure the application information such as `AppId` and `SecretKey` . For more information about how to apply for such information, see the documentation of the vendor channel.
Click **View Documentation** to see the vendor channel description.
Configure `AppId` , `AppKey` , and `AppSecret` for the vendor channel.

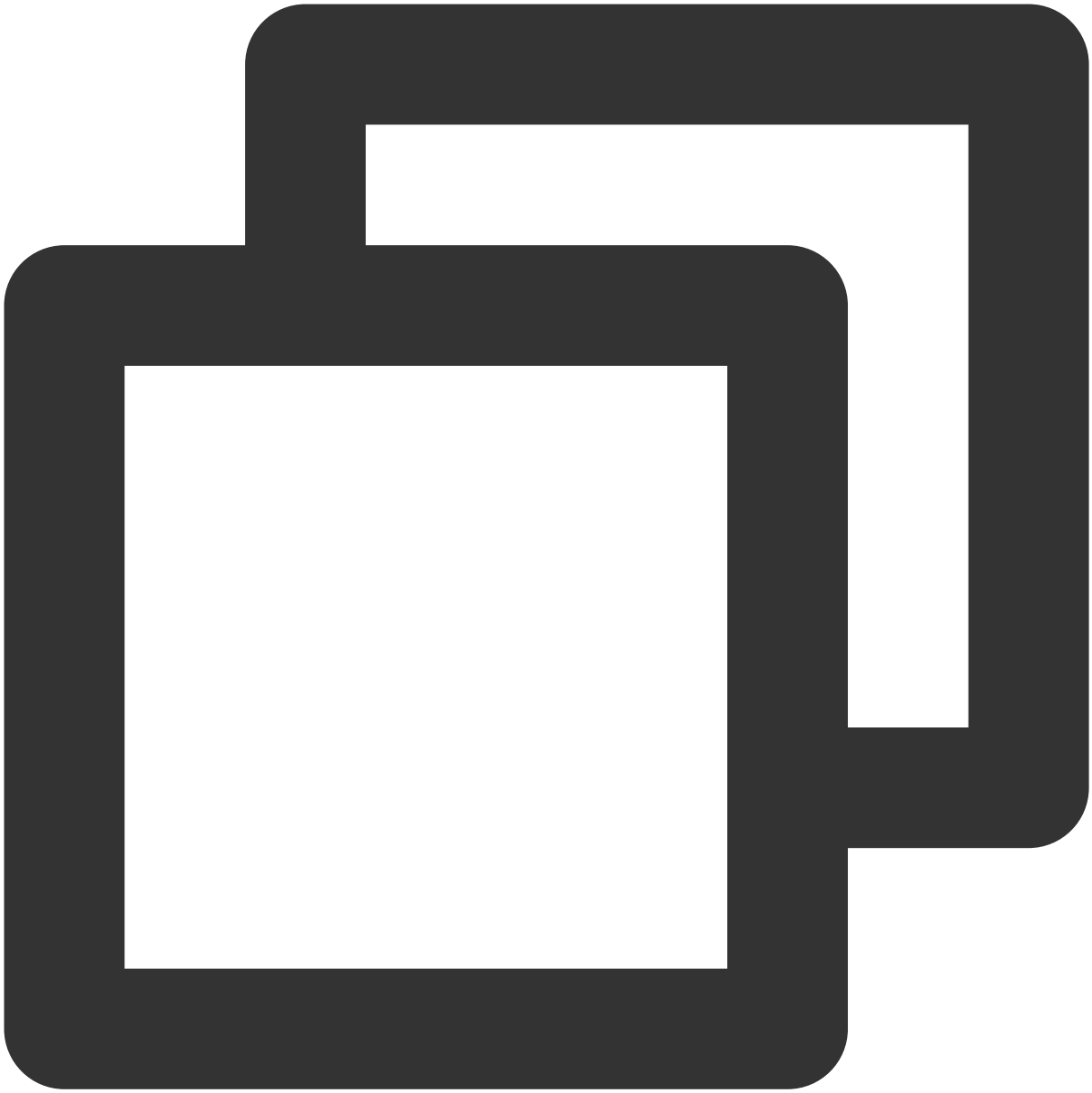


2. After configuring the vendor channel information, click **Download Configuration File** at the top of the page to download the vendor channel configuration file and use it to replace the legacy one in the project file.



Troubleshooting

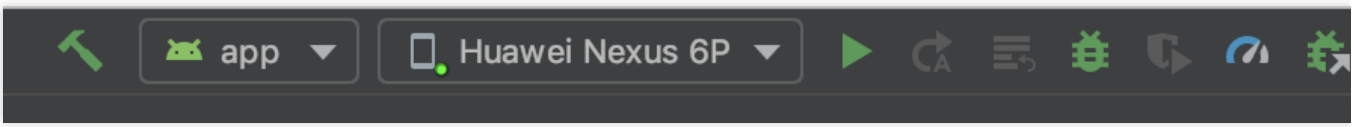
1. View the Android Gradle plugin logs.
If an exception occurs during integration, set the `debug` field in the `tpns-configs.json` file to `true` and run the following command:



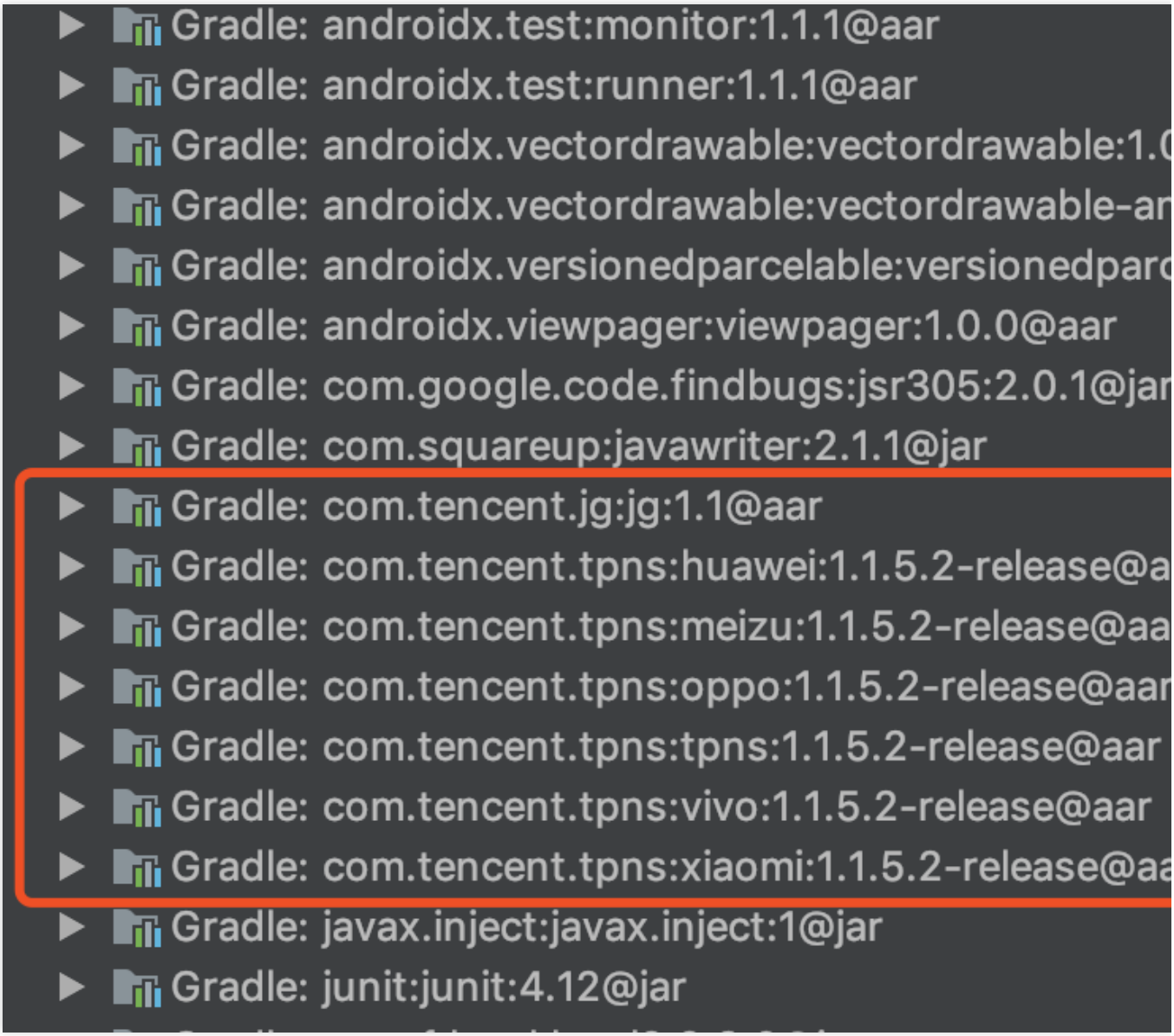
```
./gradlew --rerun-tasks :app:processReleaseManifest
```

Then, use the `TpnsPlugin` keyword for analysis.

2. Click the **sync projects** icon.



3. Check whether relevant dependencies exist in **External Libraries** of the project.



4. If the log displays `Execution failed for task ':Paracraft:checkTPNS'` , the Tencent Push Notification Service Android SDK can be updated to a later version. If you do not want to check for updates, add `"upgrade": false` to the `tpns-configs.json` file, as shown below:

5. If you encounter the version mismatch between the Android Gradle plugin and Gradle version when using the plugin, upgrade the Gradle version by referring to Android Gradle plugin release notes. The table in following figure lists which version of Gradle is required for each version of the Android Gradle plugin.

[redacted]	[redacted]
1.0.0 - 1.1.3	2.2.1 - 2.3
1.2.0 - 1.3.1	2.2.1 - 2.9
1.5.0	2.2.1 - 2.13
2.0.0 - 2.1.2	2.10 - 2.13
2.1.3 - 2.2.3	2.14.1+
2.3.0+	3.3+
3.0.0+	4.1+
3.1.0+	4.4+
3.2.0 - 3.2.1	4.6+
3.3.0 - 3.3.3	4.10.1+
3.4.0 - 3.4.3	5.1.1+
3.5.0 - 3.5.4	5.4.1+
3.6.0 - 3.6.4	5.6.4+
4.0.0+	6.1.1+
4.1.0+	6.5+

Quick Integration with iOS

Last updated：2024-01-16 17:31:40

Overview

This document describes how to quickly integrate Tencent Push Notification Service into your iOS application. You can use the local tool to configure Tencent Push Notification Service for your application in just a few clicks with no code integration required.

Preparing for Integration

Creating an iOS application

1. Before integrating the SDK, you need to log in to the Tencent Push Notification Service Console and create the product and iOS application. For detailed directions, please see Creating Products and Applications.
2. Upload the push certificate on the **Configuration Management** page. You can get the push certificate as instructed in Acquisition of Push Certificate.
3. After completing the above steps, click "Quick Integration" to download the quick integration tool.
4. Decompress the package and double-click TPNS Smart Tool.
5. The message "Unable to open TPNS Smart Tool" will be displayed.
6. Go to **System Preferences** > **Security & Privacy** > **General** and click **Open Anyway**.
7. Enter the system password as prompted to confirm the operation and click **Open Anyway** again after the confirmation. The **Open** button will be displayed. Then, click it.

Starting Integration

1. After starting the quick integration tool, go to the homepage and click **Start Integration**.
2. Enter the configuration page and configure the following 6 items as detailed below.

Configuration items 1 and 2: AccessID and AccessKey

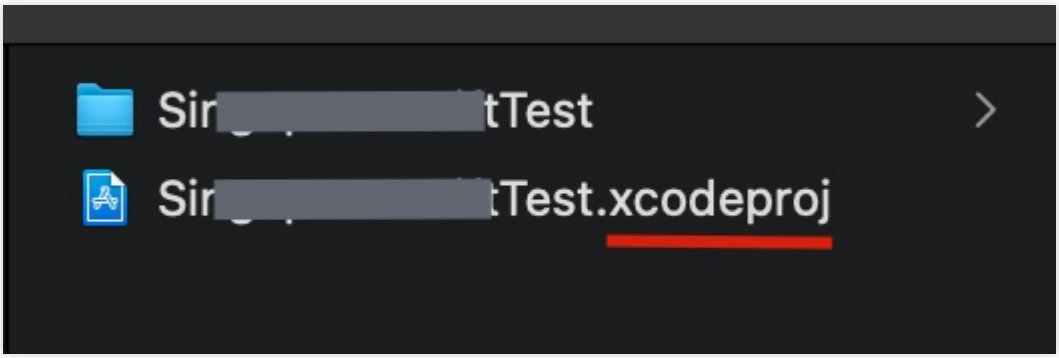
Log in to the Tencent Push Notification Service console.
1. On the "Product Management" page, find the product for which to configure the push capability and select configuration management for iOS or macOS.
2. Enter the product configuration management details page, copy `AccessID` and `AccessKey`, and paste them into the corresponding input boxes in the quick integration tool.

Configuration item 3: project language

Please select according to the language used by the `AppDelegate` file:
For `AppDelegate.m` , please select `Objective-C`
For `AppDelegate.swift` , please select `Swift`

Configuration item 4: project file

Please select the project file with the extension of `.xcodeproj` :

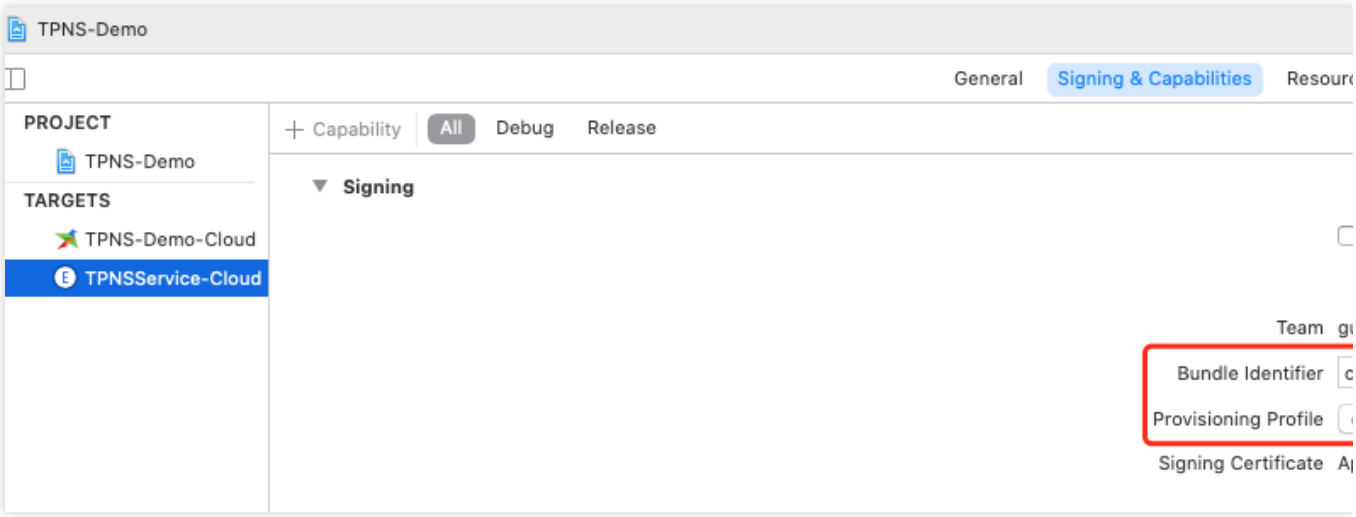


Configuration item 5: basic push capability

Basic push capability: normal push notification capability, which does not contain features such as push data reach statistics and rich media push.

Configuration item 6: notification service extension plugin

Notification service extension plugin is mainly used to calculate the reach rate of pushes and implement features such as rich media push.
If you select automatic signing for `Xcode` , `Xcode` will generate a provisioning file for your notification service extension plugin on the Apple Developer platform.
If you select manual signing for `Xcode` , you need to manually generate a provisioning file on the Apple Developer platform. Otherwise, the application cannot be installed on a real device for debugging. The steps are as follows:
1. Go to the Apple Developer platform to apply for a bundle identifier for the notification service extension plugin.
Note:
Bundle identifier naming rule: (main target Bundle Identifier).TPNSService.
2. Apply for a provisioning file containing the bundle identifier.
3. Specify `Bundle Identifier` and `Provisioning Profile` of the extension plugin as the bundle identifier and provisioning file you applied for respectively.



Note:

If you are **integrating Tencent Push Notification Service for the first time**, we recommend you check items 5 and 6 at the same time; otherwise, the push reach data cannot be obtained, and rich media pushes cannot be delivered.

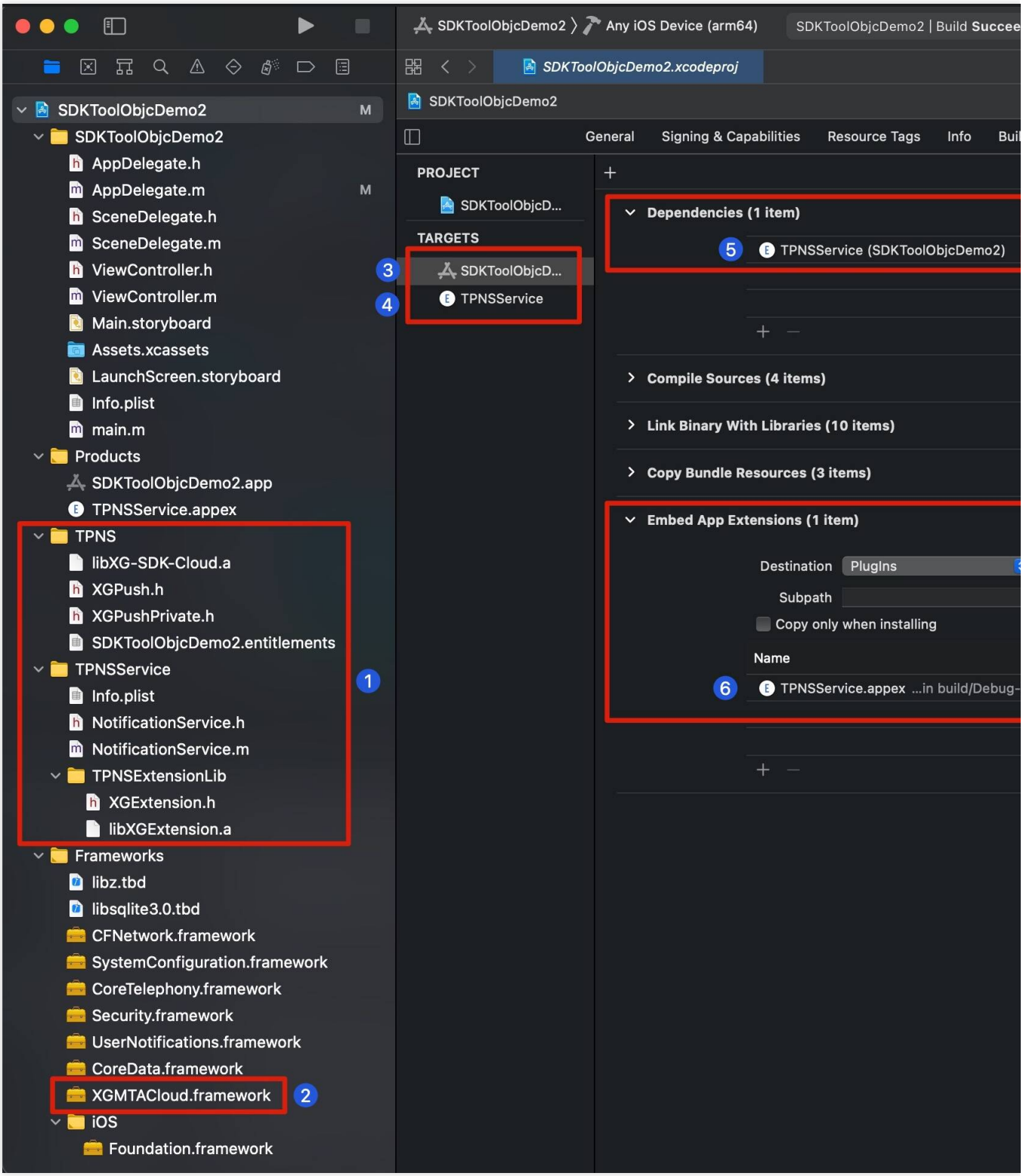
You can integrate configuration item 5 or 6 separately or integrate both of them at the same time as needed by your project.

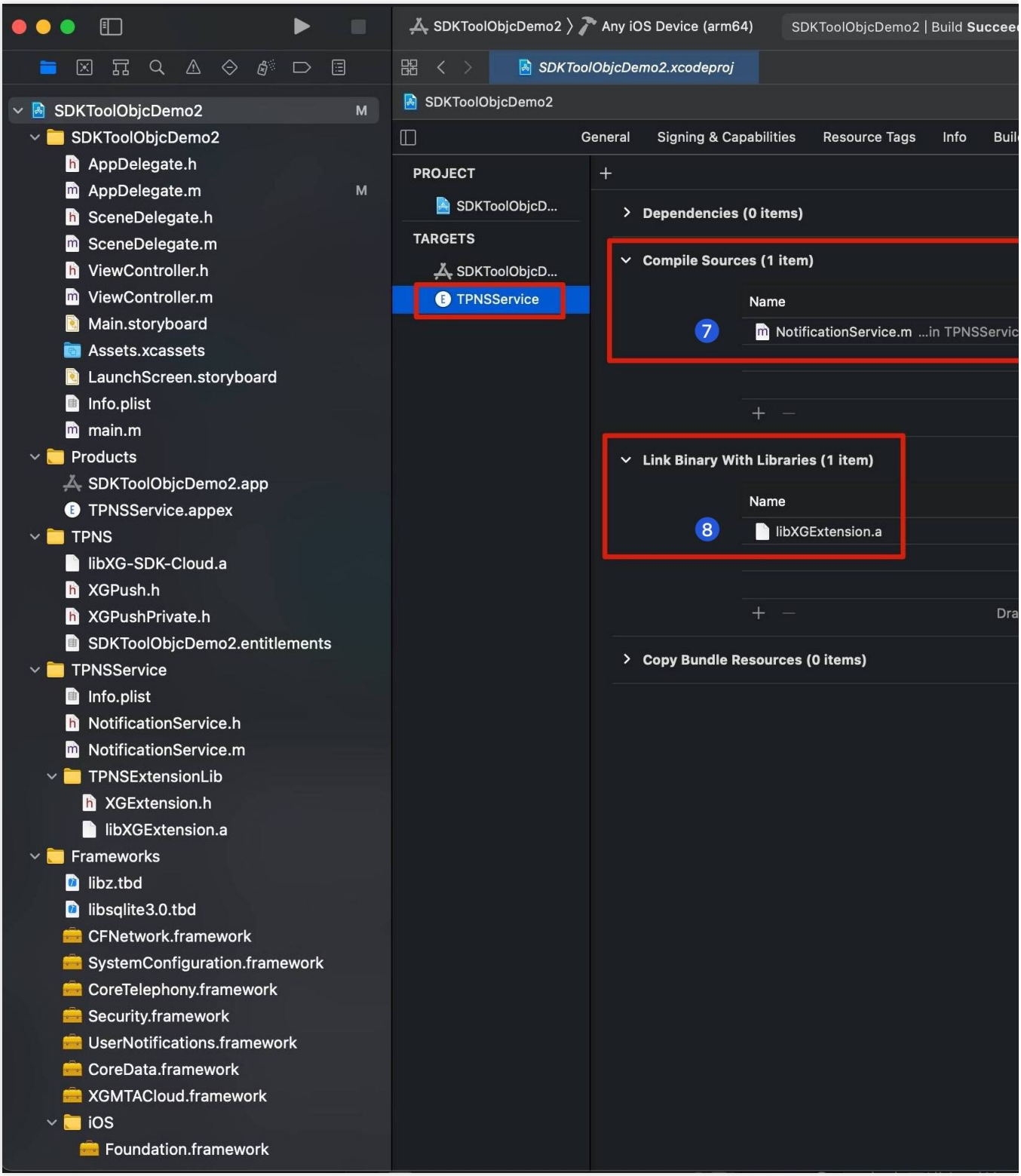
Integrating Tencent Push Notification Service SDK

1. After the above 6 configuration items are set, the **Quick Integration** button will become blue and clickable. Then, click it.
2. After successful integration, the following pop-up window will be displayed.

Project Structure and Configuration After Successful Integration

If the integration is successful, the project structure and configuration will be as shown below:

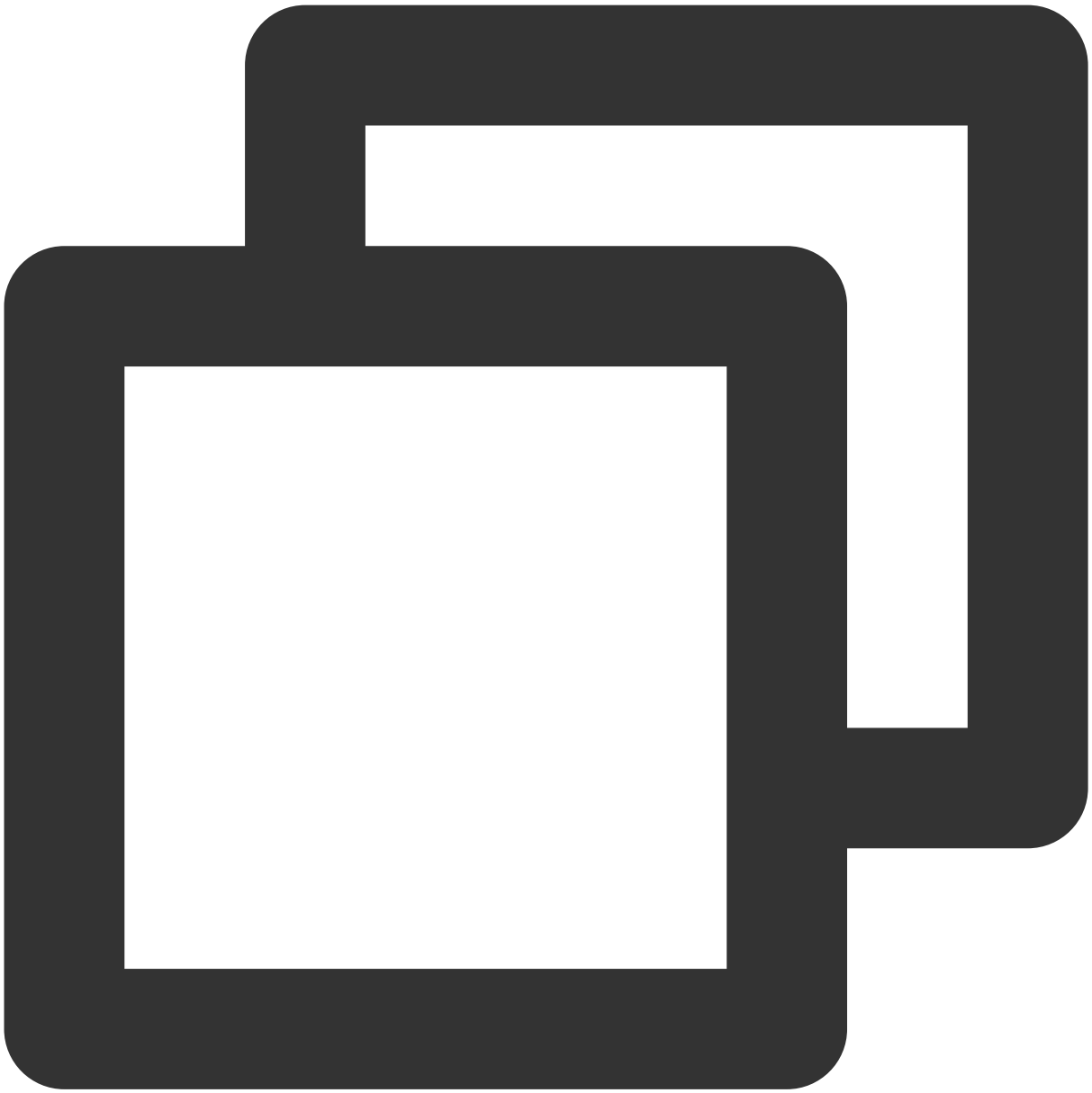




If there are exceptions such as compilation failure, no push notifications, or no reach rate statistics, please check the configuration of your project against the above figure to locate the integration error or submit a ticket for assistance.

Verifying Integration Result

Connect your iPhone to Xcode, install the application, and view the logs in the console. If a log similar to the one below is displayed, the client has properly integrated the SDK.



```
[TPNS] Current device token is 9298da5605c3b242261b57****376e409f826c2caf87aa0e6112
[TPNS] Current TPNS token is 00c30e0aeddff1270d8****dc594606dc184
```

If the token cannot be found, please check the error code returned by the registration API and troubleshoot as instructed in Error Codes.

Creating a Push Task

Last updated：2024-01-16 17:31:40

Overview

This document guides you how to create a push notification in the Tencent Push Notification Service console.

Prerequisites

You have purchased the Tencent Push Notification Service. For more information, please see Purchase Directions.

Directions

SDK download

1. Log in to the Tencent Push Notification Service console.
2. In the left sidebar, click **Message Management** > **SDK Download**.
3. On the **Download SDK** page, select the version you need to download, click **Download**, and configure according to the instructions.

Configuring push parameters

1. Log in to the Tencent Push Notification Service console.
2. In the left sidebar, click **Message Management** > **Task List**.
3. On the **Task List** page, click **Create Push**.
4. On the **Create Push** page, select **Notification bar message** for **Push Type** and enter the notification title and content. The preview of the notification will be displayed on the right. Other push settings are as follows:

Push Plan: if there are many push tasks, you are advised to categorize them and collect their statistics by push target.

Push Time: you can select **Immediate**, **Scheduled**, or **Loop**.

Push Target: you can select **All devices**, **Tag combination**, **Batch accounts**, **Account**, or **Token**.

Advanced Settings: Tencent Push Notification Service also supports the following advanced settings:

Grouping and Collapsing: you can control whether and how to collapse notifications in the device notification center.

Badge Number: this refers to the number of messages that have arrived but not been clicked. Only the Tencent Push Notification Service, Huawei, and Mi channels support automatically increasing the number by 1.

Custom Push Speed: after Custom Push Speed is enabled, messages will be pushed at the specified push speed. This feature is suitable for scenarios with limited server bandwidth resources, where a large number of messages pushed in a short period of time may result in a large number of users opening the application at the same time, leading to server resource overload.
Extra Parameter(s): you can select extra parameters in the data format of key:value for more diverse push content.
Notification Display: whether to display notification when the application is running in the foreground.
Notification Image: after you enter the image URL, notifications delivered through certain channels will display the image.
Notification Audio: after you enter the audio URL, notifications delivered through the Tencent Push Notification Service channel will contain the audio.
Offline Storage: set the offline retention time between 0 and 72 hours. The default value is 24 hours. Up to three latest pushes can be retained.
Open Location: set the click action for the push, supporting application, custom intent, URL, and in-app activity.
Reminder: set the reminder method, custom ringtone, etc.
Multi-Package Name Push: Tencent Push Notification Service supports push with multiple package names.
Channel Policy: select channels through which the push can be delivered. Smart assignment policy and custom policy are supported.
5. After selecting the push content, you can select a test device to verify the push effect before officially sending the push.
5.1 Click **Test Preview**.
5.2 Select a test device on the test preview page.
5.3 Click **Confirm** to push to the test device. The push status will be returned on the current page.
6. Click **Confirm**.
7. Click **Confirm** to push officially. The push status will be returned on the current page.
8. Click **Confirm** to go to the push record page, where the push records of both the test push and official push are displayed.

Querying Push Records

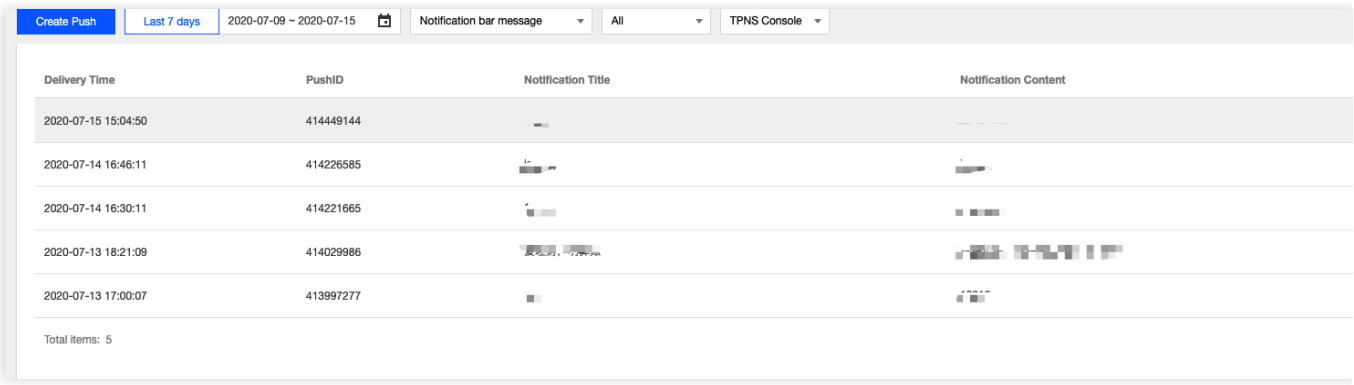
Last updated：2024-01-16 17:31:40

Overview

This document describes how to query push records (containing the message ID, title, content, sending time, and operation information) in the Tencent Push Notification Service console.

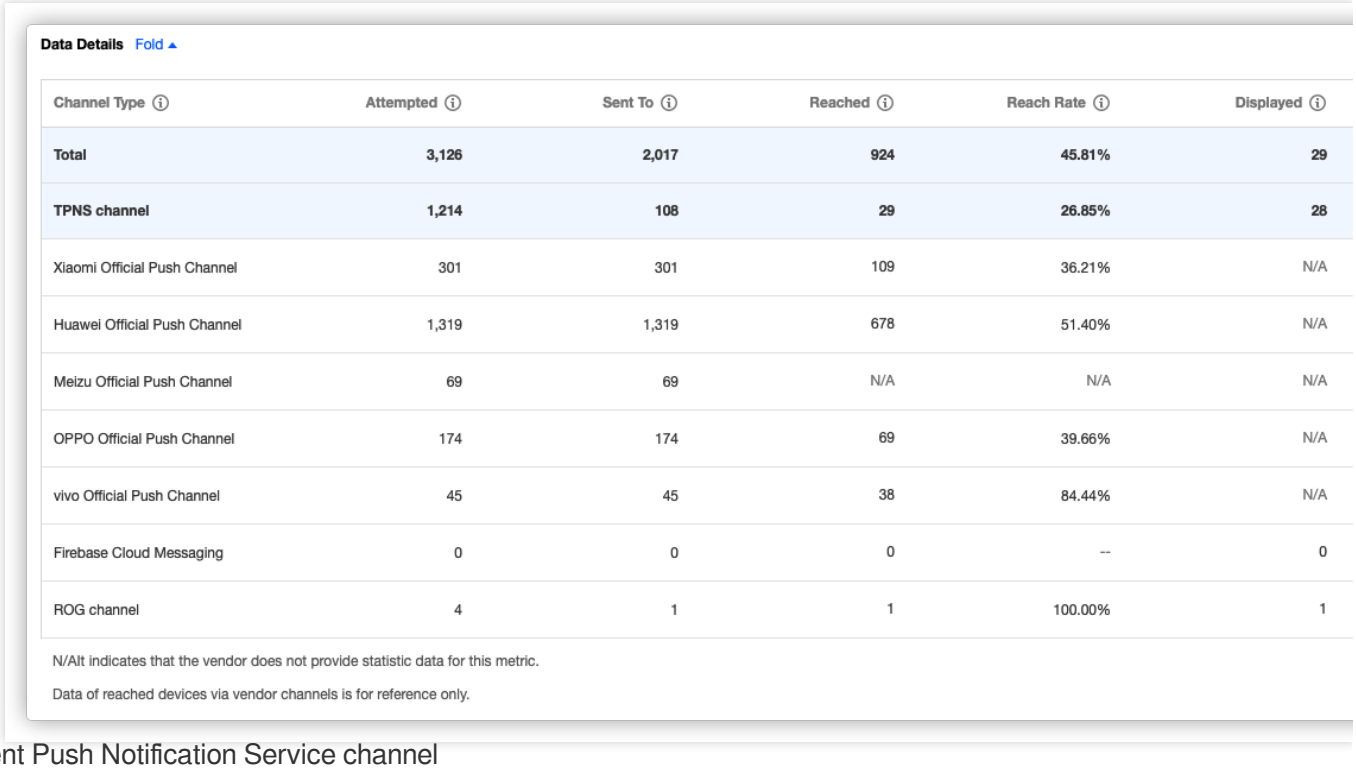
Directions

1. Log in to the Tencent Push Notification Service console, and click **Message Management** > **Task List** in the left sidebar.
2. On the task list page, click **View Details**. (Currently, only push records within the last month are saved.)



3. On the **Detailed Data** page, the data from each push process is displayed in funnel format. Android and iOS data funnels are different.

Android terminals:



Data Details Fold

Channel Type	Attempted	Sent To	Reached	Reach Rate	Displayed
Total	**3,126**	**2,017**	**924**	**45.81%**	**29**
TPNS channel	**1,214**	**108**	**29**	**26.85%**	**28**
Xiaomi Official Push Channel	301	301	109	36.21%	N/A
Huawei Official Push Channel	1,319	1,319	678	51.40%	N/A
Meizu Official Push Channel	69	69	N/A	N/A	N/A
OPPO Official Push Channel	174	174	69	39.66%	N/A
vivo Official Push Channel	45	45	38	84.44%	N/A
Firebase Cloud Messaging	0	0	0	--	0
ROG channel	4	1	1	100.00%	1

N/AIt indicates that the vendor does not provide statistic data for this metric.

Data of reached devices via vendor channels is for reference only.

Tencent Push Notification Service channel

Attempted: Of available devices in the push target devices, the number of valid devices that have connected to the server within 90 days and enabled the application's notification bar messages.

Sent To: Number of available devices to which the message was successfully delivered through the Tencent Push Notification Service channel out of the attempted devices.

Reached: Number of devices that received the message.

Clicked: Number of devices on which the message was clicked.

Cleared: Number of devices on which the notification was dismissed

Reach Rate: Reached/Sent to.

Click Rate: Clicked/Reached.

Vendor channel

Attempted: Total number of devices through the vendor channel with the notification bar enabled in the push target devices.

Sent To: Pushed to vendor server and the server returned success.

Reached: Number of devices that received the message.

Clicked: Number of devices on which the message was clicked.

Reach Rate: Reached/Sent to.

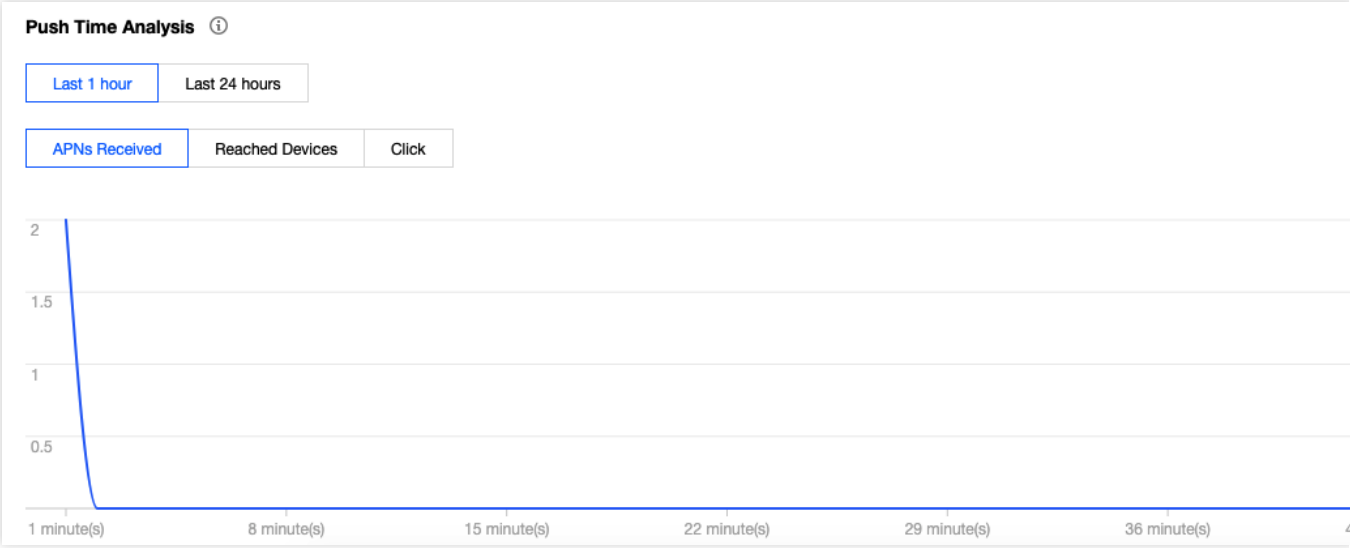
Click Rate: Clicked/Reached.

Note:

1. The number of "cleared" messages cannot be returned for vendor channels.

2. The arrival data is for reference only. For the Huawei and Meizu channels, you need to configure the arrival callback by yourself. For more information, see Acquisition of Vendor Channel Arrival Receipt.

Push time analysis



According to different time dimensions after the push task is started:

By minute: Number of unique devices for each metric per minute within one hour.

By hour: Total number of devices for each metric per hour within 24 hours or 72 hours.

By different metrics:

Sent To: Number of devices to which the message is successfully delivered in the specified period.

Reached Devices: Number of devices that received the push message within a specified period.

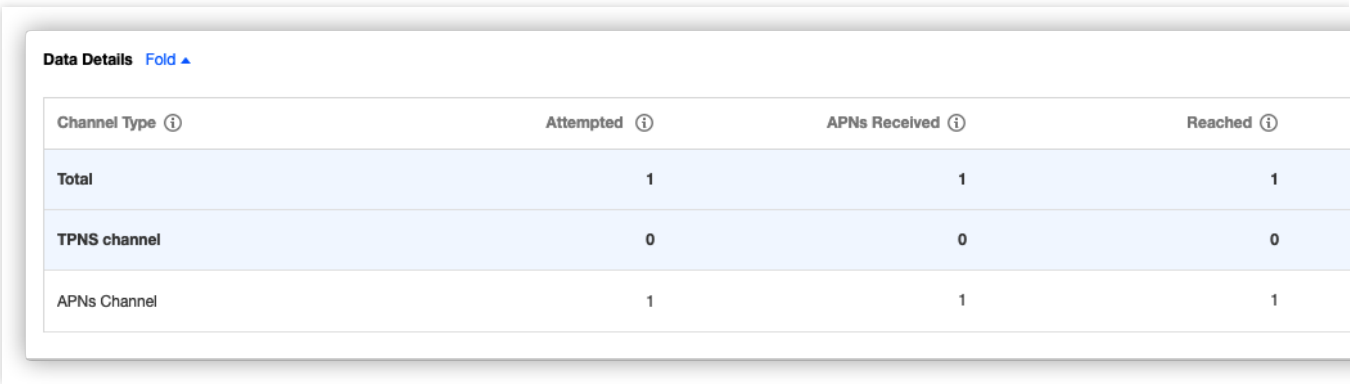
Clicked: Number of devices on which the push message was clicked within a specified period.

Cleared: Number of devices on which the notification was dismissed within a specified period.

Note:

1. The "Cleared" metric is only supported for the Tencent Push Notification Service channel.
2. Push time analysis is not supported for single-device/account pushes or device/account list pushes.
3. The statistics of timeliness metrics are slightly ahead of the statistics in the push funnel, and it is normal if you see small differences in the pushing process.

iOS:



Tencent Push Notification Service channel:

Attempted: Available devices in the push target devices (excluding devices with the notification bar disabled).

Sent To: Actual number of messages successfully delivered by the device through the Tencent Push Notification Service channel.

Reached: Number of unique devices that received push messages.

Clicked: Number of devices on which the message was clicked.

Click Rate: Clicked/Reached devices.

Note:

If no click data is reported after the message is clicked in the notification bar, check whether the application is integrated with XGMTACloud.framework.

-The Tencent Push Notification Service channel only takes effect in SDK for iOS 1.2.8.0 and above.

通道类型	计划发送	实际发送数	抵达数
总计（去重）	**1**	**1**	**1**
APNs通道	1	1	1

APNs channel:

Attempted: Available devices in the push target devices (excluding devices with the notification bar disabled).

Sent To: Actual number of messages successfully delivered by the device through the APNs channel.

Reached: Unique devices that received push messages (make sure you have completed the integration procedure. iOS 10 and below versions do not support this statistic item).

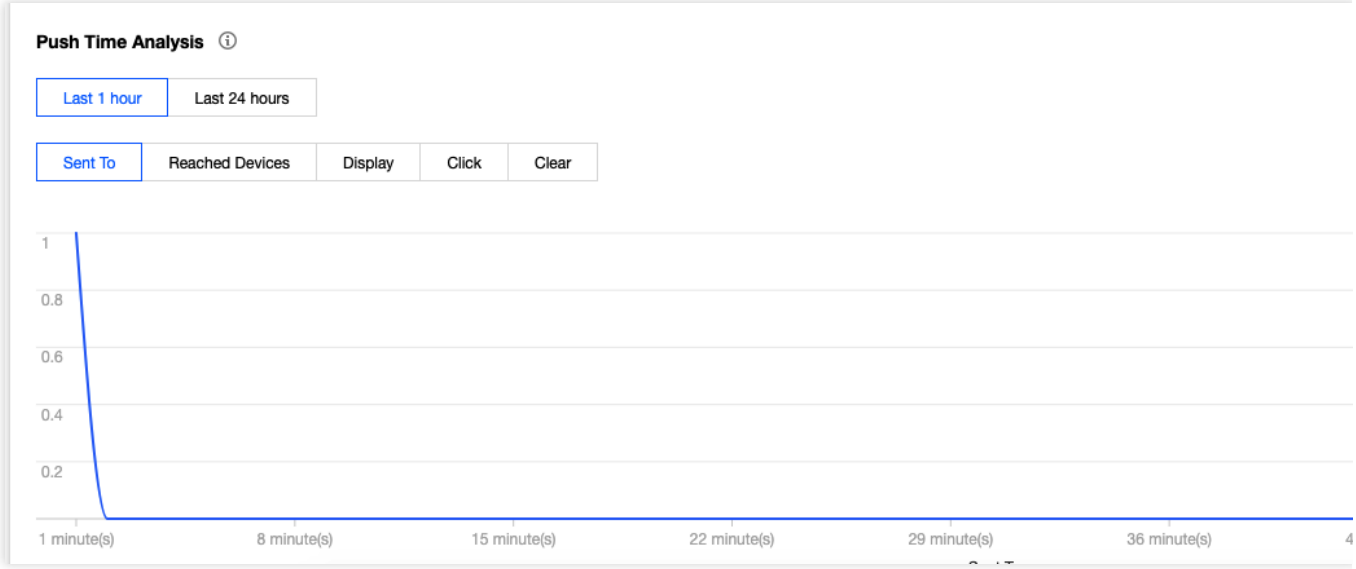
Clicked: Number of devices on which the message was clicked.

Click Rate: Clicked/Reached devices.

Note:

If no click data is reported after the message is clicked in the notification bar, check whether the application is integrated with XGMTACloud.framework.

Push time analysis



According to different time dimensions after the push task is started:

By minute: Number of unique devices for each metric per minute within one hour.

By hour: Total number of devices for each metric per hour within 24 hours or 72 hours.

By different metrics:

Sent To: Number of push messages sent to the APNs or Tencent Push Notification Service channel within a specified period.

Reached Devices: Number of devices that received the push message within a specified period.

Clicked: Number of devices on which the push message was clicked within a specified period.

Note:

1. Push time analysis is not supported for single-device/account pushes or device/account list pushes.
2. The statistics of timeliness metrics are slightly ahead of the statistics in the push funnel, and it is normal if you see small differences in the pushing process.

Guidelines for Push Testing Methods

Last updated：2024-01-16 17:31:40

Actions

Tencent Push Notification Service provides multiple push methods. You can test message pushes in different scenarios as instructed in this document. If you don't have enough testing devices, you can use the Cloud Phone service provided by WeTest.

Testing Basic Features

Push to all devices (broadcasting)

Tested Feature	Push to all devices
Test Objective	To verify that a message can be pushed to all devices.
Test Environment	Network environment: Wi-Fi or 4G Device: one or more mainstream Android or iOS devices, as applicable
Prerequisites	The Tencent Push Notification Service SDK has been integrated, and device registration has been successfully completed.
Procedure	Call the relevant API or select all devices as the push target in the console to push a message to all devices.
Expected Result	All the devices receive the message.

Push to a single device

Tested Feature	Push to a single device
Test Objective	To verify that a message can be pushed to a single device based on the device token.
Test Environment	Network environment: Wi-Fi or 4G Device: one or more mainstream Android or iOS devices, as applicable

Prerequisites	The Tencent Push Notification Service SDK has been integrated, and device registration has been successfully completed.
Procedure	1. Obtain the token information of the device to be tested. 2. 2. Call the relevant API or select a token as the push target in the console to push a message.
Expected Result	The phone receives the message.

Push to a single account

Tested Feature	Push to a single account
Test Objective	To verify that a message can be pushed to a single account.
Test Environment	Network environment: Wi-Fi or 4G Device: one or more mainstream Android or iOS devices, as applicable
Prerequisites	-
Procedure	1. Call the SDK API on the mobile app to bind the account. 2. After the specified account has been bound to the device token, push a message to this account through the relevant API or in the console.
Expected Result	The device bound to the account receives the message.
Remarks	Binding an account on Android Binding an account on iOS

Push to a list of accounts

Tested Feature	Push to a list of accounts
Test Objective	To verify that a message can be pushed to a list of accounts.
Test Environment	Network environment: Wi-Fi or 4G Device: one or more mainstream Android or iOS devices, as applicable
Prerequisites	-

Procedure	Call the relevant API or use the console to push messages to the bound accounts.
Expected Result	The devices bound with the accounts receive the message.
Remarks	Binding an account on Android Binding an account on iOS

Push to devices with specified tags

You can use the console or relevant API to push messages to devices with specified tags, and the API allows you to set "AND" and "OR" relationships for multiple tags.

Tested Feature	Push to devices with specified tags
Test Objective	To verify that you can set tags for different user groups and then push messages to the user groups by tag names.
Test Environment	Network environment: Wi-Fi or 4G Device: one or more mainstream Android or iOS devices, as applicable
Prerequisites	The Tencent Push Notification Service SDK has been integrated, and you have successfully set custom tags.
Procedure	1. When creating a push task in the console, select one or more custom or system preset tags, set the "AND" or "OR" relationship, and start the push. 2. Call the relevant API to select one or more custom tags, set the "AND" or "OR" relationship, and start the push.
Expected Result	The user groups with the specified tags received the messages.
Remarks	API for setting custom tags (Android) API for setting custom tags (iOS)

Advanced Feature Tests

Push through multiple vendor channels

Tested Feature	Push through multiple vendor channels
Test Objective	To verify that a message can be received after the application process is ended on a device.

Test Environment	Mi, Huawei, Meizu, OPPO, and vivo models are required. For FCM, Google Play must be installed.
Prerequisites	1. You have signed up with these vendors' push platforms and created the required application. 2. You have configured vendor channels on the Configuration Management page. 3. You have integrated the SDK according to the integration methods for different vendor channels provided on the official website. 4. You have enabled the vendor channels in the SDK.
Procedure	1. Install the app integrated with the vendor channel on a phone. 2. Sign up with the vendor channel and get the vendor token. 3. Push a message to the phone through the relevant API or in the console.
Expected Result	The message can be successfully pushed to a single device or all devices after the Tencent Push Notification Service application runs in the background and all application processes are ended.
Remarks	For Huawei Push, you need to use a signed package. For more information, see here.

Scheduled push

Tested Feature	Scheduled push
Test Objective	To verify that a message can be successfully pushed at a specified time point.
Test Environment	Network environment: Wi-Fi or 4G Device: one or more mainstream Android or iOS devices, as applicable
Prerequisites	The Tencent Push Notification Service SDK has been integrated, and device registration has been successfully completed.
Procedure	1. Set the push time through the relevant API or in the console. 2. Select all devices or a tag as the push target.
Expected Result	The tested phones receive the messages at the specified time.
Remarks	Scheduled push is supported only by push to all devices and push to devices with specified tags.

Loop push

Tested Feature	Loop push

Test Objective	To verify that a message can be received when the customized looping conditions (loop push date and loop type) are met.
Test Environment	Network environment: Wi-Fi or 4G Device: one or more mainstream Android or iOS devices, as applicable
Prerequisites	The Tencent Push Notification Service SDK has been integrated, and device registration has been successfully completed.
Procedure	1. Set the loop push time and loop type through the relevant API or in the console. 2. Select all devices or a tag as the push target.
Expected Result	When the loop conditions are met, the tested phones receive the message.
Remarks	Loop push is supported only by push to all devices and push to devices with specified tags.

In-app message push

Messages directly passed through to Android devices are not displayed on the notification bar. They will be handled by the Tencent Push Notification Service application after being received.

Tested Feature	In-app message push
Test Objective	To verify in-app message push.
Test Environment	Network environment: Wi-Fi or 4G Device: one or more mainstream Android or iOS devices, as applicable
Prerequisites	The Tencent Push Notification Service SDK has been integrated, and device registration has been successfully completed.
Procedure	Push an in-app message in the console or through the relevant API.
Expected Result	The application can receive the in-app message.

Rich media push

You can push multimedia (rich media) messages containing information such as images, audio, and videos to clients.

Tested Feature	Rich media push
Test Objective	To verify that a rich media message such as an image can be successfully pushed.

Test Environment	Network environment: Wi-Fi or 4G Device: one or more mainstream Android or iOS devices, as applicable
Prerequisites	**Android:** messages with rich media will be delivered only through the Tencent Push Notification Service channel. Images in native Android systems are available in big images and thumbnails, whose display effects vary by model and custom system. 1. Android supports static images, audio, and videos as rich media content. 2. The image resolution should be 430*2303, and the rich media link can only use HTTPS. 3. For audio/video rich media, you need to create an .xml file after integrating the SDK. For more information, please see Instructions for Audio/Video Rich Media. **iOS:** the system supports rich media content such as image, audio, and video. For images, iOS displays the big images when the user uses Force Touch for interaction and thumbnails in other scenarios (regular image formats and GIF are supported). 1. Only images in JPEG, PNG, and GIF formats are supported. 2. The size of an image must be kept below 10 MB. 3. Only audio files in AIFF, WAV, MP3, and MP4 formats are supported. 4. The size of an audio file must be kept below 5 MB 5. Only videos in MPEG, MPEG2 Video, MPEG4, and AVI formats are supported 6. The rich media link can only use HTTPS.
Procedure	1. Create a push task in the console or through the relevant API. 2. Enable rich media and enter the rich media file address.
Expected Result	The application can receive the rich media message.

Offline message retaining

Tested Feature	Offline message retaining
Test Objective	To verify that messages can be retained offline.
Test Environment	Network environment: Wi-Fi or 4G Terminal: Android devices from mainstream vendors
Prerequisites	The Tencent Push Notification Service SDK has been integrated, and device registration has been successfully completed.
Procedure	1. Make the Tencent Push Notification Service application run in the background and end all application processes. 2. Push multiple messages.
Expected	When the application runs in the background, the tested devices cannot receive the

Result	messages. When the application is started again, the tested devices receive the messages, and the messages are displayed in the push order.
Remarks	This feature can be tested only through the Tencent Push Notification Service channel. Offline messages can be retained for up to 72 hours, and the latest three messages can be retained. If you need to retain more messages, please contact the customer service.

Message reminder (custom ringtone)

Tested Feature	Message reminder (custom ringtone)
Test Objective	To verify that you can set a custom ringtone.
Test Environment	Network environment: Wi-Fi or 4G Device: one or more mainstream Android or iOS devices, as applicable
Prerequisites	The Tencent Push Notification Service SDK has been integrated, and device registration has been successfully completed.
Procedure	1. In the console, create a push task and select a custom ringtone in **Advanced Settings**. 2. Push a message.
Expected Result	The message reminder is the custom ringtone.
Remarks	Android supports customizing a ringtone, vibration, and notification LED (note: only the Tencent Push Notification Service channel supports this feature). iOS supports customizing a ringtone.

iOS badge setting

Tested Feature	iOS badge setting
Test Objective	To verify that you can set an iOS badge.
Test Environment	Network environment: Wi-Fi or 4G Terminal: iOS device
Prerequisites	You have completed configurations as instructed in the development documentation at the official website for the "automatically increasing the badge number by 1" feature.
Procedure	1. Create a push task in the console and set **Badge Number** to **Unchanged**, **Set to**, or **Auto**

	increased by 1. 2. Push a message. 3. Receive the message and check the badge number.
Expected Result	The badge display meets the settings.
Remarks	Implementation method of the iOS badge API

Push redirection

Tested Feature	Redirect push to a specified page
Test Objective	To verify that redirection works properly after a message is clicked on the notification bar.
Test Environment	Network environment: Wi-Fi or 4G Terminal: Android devices from mainstream vendors
Prerequisites	You have completed custom redirection configuration on the client as instructed in here.
Procedure	1. Create a push task in the console. Expand the **Advanced Settings** area. 2. Set **Click to Open** to **Application**, **Custom Intent(Recommended)**, **URL**, or **In-App Activity**. 3. Push a message. 4. Click the message on the notification bar and check whether the system redirects to the specified page as expected.
Expected Result	The system redirects to the page as expected after the message is clicked in the notification bar.
Remarks	Vendor channels only support clicking to open an app or custom item (intent), while the Tencent Push Notification Service channel supports all click actions. For more information on iOS push redirection, please see here.

Product Restrictions

Last updated：2024-01-16 17:31:40

Trial-Edition Application Limit

For applications of the trial edition, up to 1,000 devices can be used for testing. If the limit is exceeded, the trial may be terminated. To avoid loss, do not use applications of the trial edition for commercial purposes.

Notification Title and Content Length Limits

Android push channel limits

Channel	Notification Title Length Limit	Notification Content Length Limit
Tencent Push Notification Service	20 characters (Excess parts will be displayed as an ellipsis.)	Unlimited
Mi	50 characters	128 characters
Huawei	40 characters	80 characters
Meizu	32 characters	100 characters
OPPO	32 characters	200 characters
vivo	40 characters	128 characters

Note:
Pushes through vendor channels will fail if corresponding length limits are exceeded, and they will be retried through the Tencent Push Notification Service channel.
Pushes through vendor channels will fail if the notification title or content is empty, and they will be retried through the Tencent Push Notification Service channel.
For Android, the size of the pushed message body cannot exceed 4 KB, and this limit applies to the `message` field in the Push API.

iOS push channel limits

Channel	Notification Title Length Limit	Notification Content Length Limit

APNs	40 characters (Excess parts will be displayed as an ellipsis.)	Up to 110 characters will be displayed in the notification center, and excess parts will be displayed as an ellipsis. Up to 110 characters will be displayed when the phone screen is locked, and excess parts will be displayed as an ellipsis. Up to 62 characters will be displayed in the top pop-up window, and excess parts will be displayed as an ellipsis.
Tencent Push Notification Service	Same as that of APNs	Same as that of APNs

Note:
For iOS, the size of the pushed message body cannot exceed 4 KB, and this limit applies to the `message` field in the Push API.

API Use Limits

For batch pushes through APIs, you can specify up to 1,000 device tokens or accounts at a time.
For pushes to devices with specified tags, the tag list cannot exceed 512 characters.
For pushes to all devices, the same message can only be sent once per hour, and there is no limit on the frequency and number of sending times for other push targets.
The call frequency limit for statistics APIs is 200 times per hour.
One application can have up to 10,000 custom tags. One device token can be bound to a maximum of 100 custom tags (if you want to increase this limit, please submit a ticket). One custom tag can be bound to an unlimited number of device tokens.

Push Volume and Push Rate Limits

For the Tencent Push Notification Service channel, the push volume is unlimited. For Android vendor channels, see their push volume limits in Vendor Channel Limit Description.
For the Tencent Push Notification Service channel, the push rate is unlimited by default. If necessary, you can call the relevant API to customize a push rate. For relevant parameter descriptions, see Optional Parameters.
For Android vendor channels, see their push rate limits in Vendor Channel QPS Limit Description.