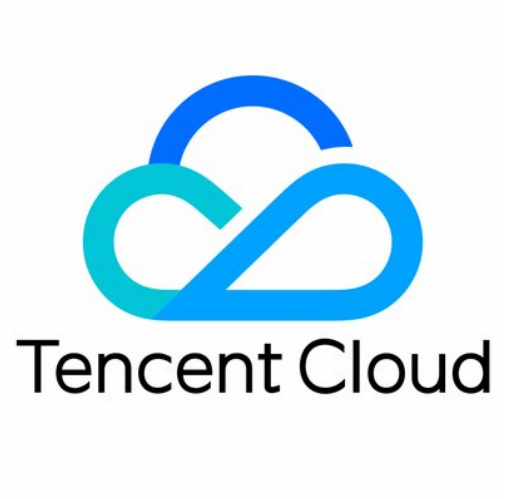
Tencent Push Notification Service

Overview

Product Documentation



©2013-2019 Tencent Cloud. All rights reserved.

Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

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

Trademark Notice

Tencent Cloud

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

Service Statement

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

Contents

©2013-2019 Tencent Cloud. All rights reserved.

Overview
Product Overview

Last updated：2019-11-12 16:57:17

Product Overview

Overview

TPNS is a professional mobile app push platform that can deliver tens of billions of notifications/messages per second. It can effectively increase user retention rate and engagement.

TPNS provides full-linkage mobile push capabilities. To push messages to users' mobile devices, you only need to connect Tencent Cloud's TPNS SDK, which takes less than 10 minutes. Click here to view tutorials.

TPNS also provides a convenient web-based console for operations, so developers can easily manage push messages, view push data, and test push notifications. You can send push messages and monitor data in real time through simple configurations.

TPNS also provides comprehensive device-specific and backend APIs , so you can customize push-related businesses and achieve advanced use cases such as personalized targeting.

Basic Features

TPNS offers a variety of powerful features to help you deliver messages to users precisely and rapidly.

Multiple push methods and types

TPNS offers a variety of push methods, to meet different business and industry needs. TPNS offers push types such as notification panel messages, in-app push notifications, as well as redirection to apps, H5 and Deeplink after clicking. TPNS also offers scheduled and repeated push notifications to meet your needs in different use cases.

Self-built channels+ vendor channels, rapid and stable delivery

TPNS supports the integration of primary vendor channels and the Google-supported FCM channel overseas. It can select the optimal channel based on the device brand, to implement system-level push on

©2013-2019 Tencent Cloud. All rights reserved.

vendor mobile phones and effectively improve delivery rate. You can push up to 300,000 messages per second via exclusive channels and deliver a message in milliseconds.

Session Keep-Alive

TPNS's unique dual service architecture for session keep-alive uses master-slave dual services for multiple apps instead of only using one service for one app. It enables stable message delivery, reduces power and traffic consumption, increases delivery as well as click-through conversion rate. This feature is free for users of top apps such as Arena of Valor, significantly improving message delivery.

Abundant tags, precise push

Developers can call TPNS SDK and backend APIs to bind one or multiple tags to the device. After binding, developers can target tags for push notifications to facilitate more precise operations.

Real-time push effect analysis

TPNS provides real-time analysis of push effects. Data for message delivery and click-through is collected in real time and displayed visually in the console, to facilitate real-time monitoring of push results.

©2013-2019 Tencent Cloud. All rights reserved.

Product Advantages

Last updated：2019-11-12 16:58:43

Product Advantages

Excellent and reliable performance metrics

TPNS provides processing power for 300,000 push messages per second, outperforming its competitors in the industry. It also supports billions of push messages per day for applications like Arena of Valor. The stability of the system and SDK has been tested on a large number of apps. TPNS is also selected as the only official third-party push service provider for Android system (Google).

Strong keep-alive capability ensures high delivery rate

With the industry's highest daily live device coverage of 320 million and Tencent app's mutual dual keep-alive mechanism, TPNS ensures stable persistent connections, offers a high keep-alive rate, and a delivery rate of 99.9% for online devices. It supports different vendor channels, and can select the optimal channel based on the device brand, to implement system-level push on vendor mobile phones and effectively improve the push delivery rate.

Massive number of tags and precise push notifications

Developers can call TPNS SDK and backend APIs to bind one or multiple tags to the device. After binding, developers can target tags for push notifications to facilitate more precise operations.

Tencent digital application ecosystem

Taking TPNS as the core, applying Tencent's digital capabilities to elastically and rapidly establish a closed-loop digital ecosystem for enterprises from the base layer to the top layer. Using Tencent's big data capabilities such as smart push, data analysis, and digital marketing, and combining them with TPNS to help customers break through push service bottlenecks and drive engagement.

Real-time push effect analysis

TPNS provides real-time analysis of push effects. Data for message delivery and click-through is collected in real time and displayed visually in the console, to facilitate real-time monitoring of push results.

©2013-2019 Tencent Cloud. All rights reserved.

TPNS Overview

Last updated：2019-08-08 17:22:25

1. Services Provided by TPNS

Push notification is a critical way to reach users in the development and operation of mobile applications. Push notification can build up interactions with users and increase product usage.

TPNS helps developers integrate mobile push capabilities, manage push messages, and evaluate push effects.

1.1. TPNS PUSH SDK

TPNS provides full-link mobile push capabilities. TPNS SDK access takes less than 10 minutes and user-end push notifications can be enabled. For tutorials, click here.

1.2. Web-based TPNS Console

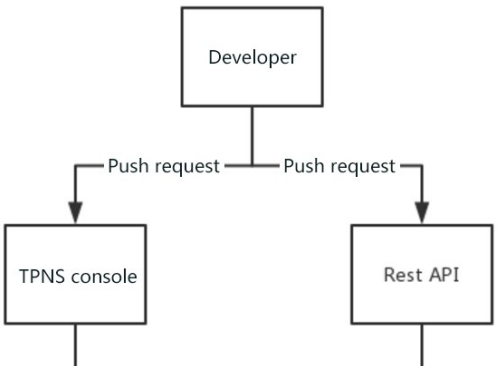
TPNS console is web-based and very convenient. You can manage push messages, view push data, and test push notifications. You can send messages and monitor data in real time with some simple configurations.

1.3. Advanced Custom APIs

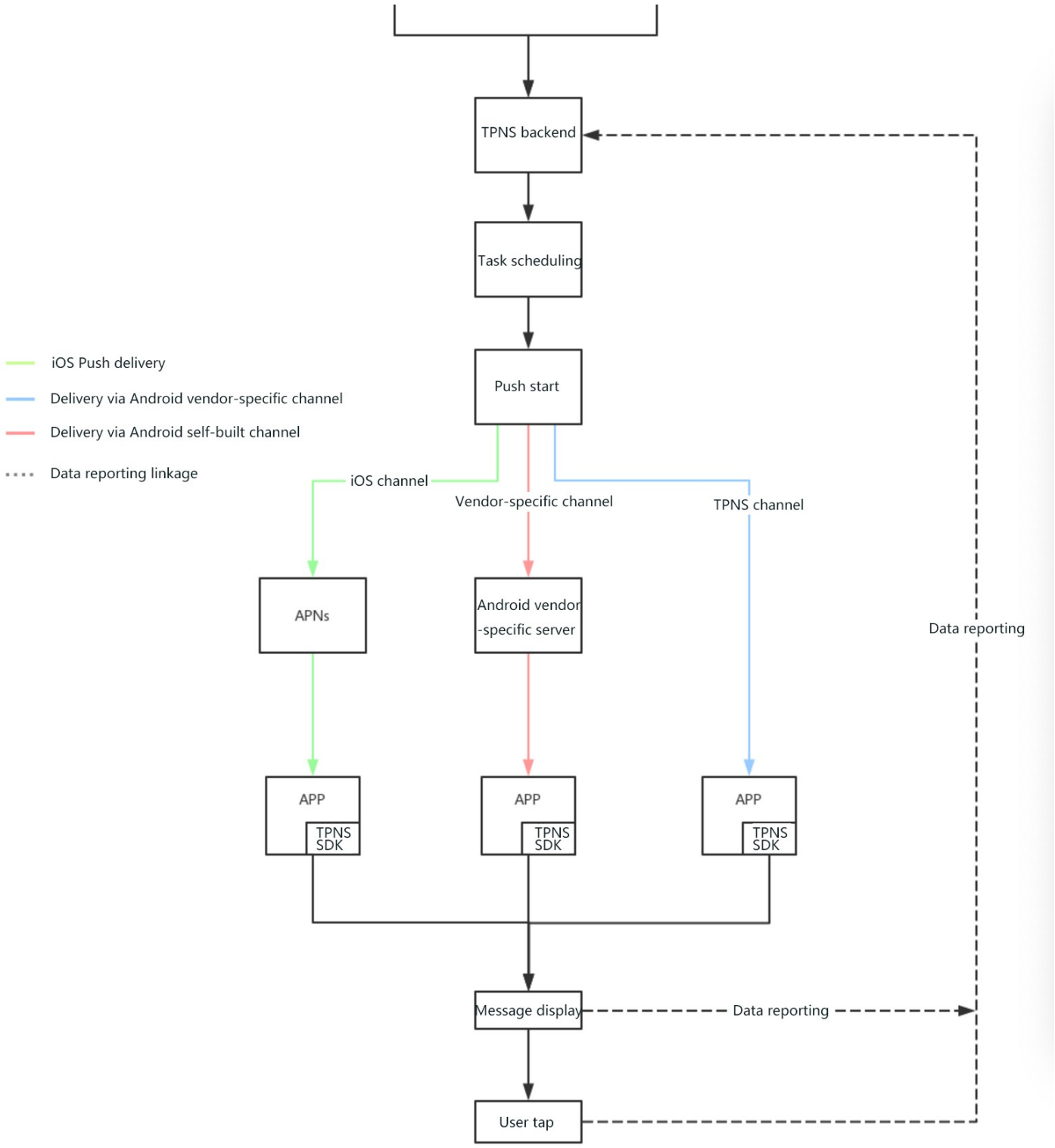
TPNS offers comprehensive device-specific and backend APIs to help you customize push-related business logic, so you can achieve advanced functionalities such as personalized targeting.

2. Overview of TPNS Push Process

This section describes the push process of TPNS consisting of two linkages: "message delivering" and "data reporting".



©2013-2019 Tencent Cloud. All rights reserved.



3. Basic Concepts

3.1. Overview

©2013-2019 Tencent Cloud. All rights reserved.

This section defines some basic concepts in TPNS. **This section helps you better understand TPNS and other documents.**

3.2. Message Type

Notification bar message

This refers to the message displayed by the operating system in the notification bar, and the user can tap the message to open the app.

In-app message

This refers to the message directly passed through to the Android device, which will not be actively displayed in the notification bar and will be handled by the app after received.

3.3. Push Channel

Android vendor-specific channel

This refers to the push channel at the operating system level provided by Android mobile phone manufacturers, which enables message receipt and display without launching the app. Major Chinese mobile phone manufacturers has already launched this service.

APNs

APNs is short for Apple Push Notification service, the push channel in iOS. All iOS push messages are sent to the device via APNs.

Android TPNS channel

This is TPNS' self-built push channel, which maintains the connection between the device and the TPNS server through a unique sharing channel mechanism to ensure delivery of the push message.

FCM channel

FCM is short for Firebase Cloud Messaging, a Google push service and channel with push capabilities at the operating system level outside mainland China.

TPNS multi-channel integration

TPNS can intelligently select the push channel based on different use cases. It selects the best channel according to device types to ensure effective message delivery.

3.4. TPNS Service Format

TPNS console

This is TPNS' visual push management interface, where you can send push messages, query push history, analyze data, develop and test pushes. It can be used after you log in on TPNS' official website.

REST API

This refers to the backend API provided by TPNS, which implements functions such as message push,

©2013-2019 Tencent Cloud. All rights reserved.

device mapping management, and data querying. It makes it easier for you to customize pushes and is compatible with your existing business logic. For details, see TPNS Backend REST API.

Mobile SDK

TPNS provides a push SDK for both iOS and Android. The SDK is mainly used to display push messages and collect statistics. In addition, the SDK comes with a set of APIs that can help you achieve the business logic of targeted push. For details, see Android SDK API or iOS SDK API.

3.5. Push Process

Push_Token

Push_Token is an identifier of a device generated by TPNS (hereinafter referred to as Token) used to push messages to the device, which is the smallest unit of push.

Device push registration

The push registration of a device indicates that a persistent connection has been successfully established between the device and the TPNS server. The device will communicate with the server.... (professional description is required here)

Persistent connection

A persistent connection is a continuous connection between the TPNS Android SDK and TPNS backend. Messages can be delivered and displayed as long as the connection is valid. If a persistent connection cannot be established, the push message will be retained in TPNS backend and delivered after the connection succeeds.

Account

This is a concept used in targeted push, which may be called alias in other push systems. Accounts can be configured and managed in the device SDK and TPNS backend. After an account is bound, it can be used as the push target to push messages.

Tag

This is another concept used in targeted push. You can call the TPNS SDK and backend API to bind one or more tags to the device. After that, you can push messages based on the tags, which makes lean operations easier.

©2013-2019 Tencent Cloud. All rights reserved.

Update Log

Last updated：2019-06-26 10:41:25

Notes for the Android SDK Update

Android SDK v4.3.2 (Beta)

Update Date

- March 7, 2019

Version Update Notes

1. Addressed the occasionally failed FCM unregistration issue
2. Updated the Meizu SDK to fix the NotificationService component vulnerability
3. Optimized the issue with push using the combination of Mi channel and FCM channel
4. Fixed known bugs

Upgrade Tips

- Recommended to upgrade

Download Address

Android SDK v4.3.1

Android SDK v4.3.1 (Beta)

Update Date

- January 24, 2019

Version Update Notes

1. Fixed the bug with persistent connection in the SDK which is now more stable
2. Adapted to the latest version of Mi, Meizu, and Huawei channels
3. Fixed the issue with exceptions in the dynamically loaded Huawei channel

Upgrade Tips

- Recommended to upgrade

Download Address

©2013-2019 Tencent Cloud. All rights reserved.

Android SDK v4.3.1

Android SDK v4.0.5

Update Date

- January 11, 2019

Version Update Notes

1. Enhanced the stability of the TPNS channel
2. Enhanced the compatibility with Android 7+
3. Fixed occasional crashes

Download Address

Android SDK v4.0.4

Android SDK v4.3.0 (Beta)

Update Date

- January 14, 2019

Version Update Notes

1. Fixed the bug with persistent connection in the SDK which is now more stable
2. Adapted to the latest version of Mi, Meizu, and Huawei channels

Download Address

Android SDK v4.3.0

Android SDK v4.0.4

Update Date

- December 26, 2018

Version Update Notes

1. Enhanced the stability of the TPNS channel
2. Enhanced the compatibility with Android 7+

Upgrade Tips

- Recommended to upgrade

©2013-2019 Tencent Cloud. All rights reserved.

Download Address

Android SDK v4.0.4

Android SDK v4.2.0 (Beta)

Update Date

- September 13, 2018

Version Update Notes

1. Added the support for rich media notification bar push to increase tap rate
2. Added the support for batch account API to make it easier to configure push targets
3. Fixed known issues

Upgrade Tips

- Recommended to upgrade

Download Address

Android SDK v4.2.0

Android SDK v4.0.3

Update Date

- September 13, 2018

Version Update Notes

1. Added the support for dynamically loaded vendor-specific channel to reduce the package size
2. Adapted to the latest Android P
3. Added the batch tag operation API
4. Optimized session keep-alive

Upgrade Tips

- Recommended to upgrade

Download Address

Android SDK v4.0.3

Android SDK v3.2.6

Update Date

©2013-2019 Tencent Cloud. All rights reserved.

- August 6, 2018

Version Update Notes

1. Upgraded the Mi channel and fixed security vulnerabilities
2. Fixed known crashes
3. Fixed the bug where notifications could not be dismissed

Download Address

Android SDK v3.2.6

Android SDK v3.2.5

Update Date

- July 20, 2018

Version Update Notes

1. Fixed the ANR issue
2. Fixed vendor-specific channel vulnerabilities
3. Fixed the issue with offline retention failure

Download Address

Android SDK v3.2.5

Android SDK v3.2.3

Update Date

- May 4, 2018

Version Update Notes

1. Optimized the network connectivity to increase the stability
2. Fixed the issue where tap on notifications could not redirect to the specified page on certain devices
3. Fixed the issue where notifications could not be popped up on certain models running Android 8.0
4. Fixed other common issues

Download Address

Android SDK v3.2.3

Android SDK v4.0.3 (Beta)

Update Date

©2013-2019 Tencent Cloud. All rights reserved.

- July 20, 2018

Version Update Notes

1. Added the support for dynamically loaded vendor-specific channel to reduce the package size
2. Adapted to the latest Android P
3. Added the batch tag operation API
4. Optimized session keep-alive

Download Address

Android SDK v4.0.3 (beta)

Android SDK v3.2.4 (Beta)

Update Date

- May 4, 2018

Version Update Notes

1. Fixed the occasional 10103 error
2. Unified the receiver for Mi, Meizu, and Huawei channels (so that you no longer need to develop receivers for each vendor-specific channel on your own)
3. Upgraded the FCM version and fixed crashes
4. Added the support for vendor-specific channel display and tap data reporting (currently, this has not been fully co-debugged with the backend and will continue to be verified next week)

Download Address

Android SDK v3.2.4 (beta)

iOS SDK Update Notes

iOS SDK v3.3.5

Update Date

- February 19, 2019

Version Update Notes

1. Fixed the conflict in loop calls when coexisting with Firebase SDK

iOS SDK v3.3.4

©2013-2019 Tencent Cloud. All rights reserved.

Update Date

- December 26, 2018

Release Update Notes

1. Added the support for rich media push to increase tap rate (v3.3.2 beta)
2. Added the support for iOS 12 to make push delivery more stable (v3.3.2 beta)
3. Added the support for batch account API to make push more flexible (v3.3.2 beta)
4. Fixed memory leaks to improve stability (v3.3.3 beta)

Upgrade Tips

Recommended to upgrade

Download Address

iOS SDK v3.3.4

iOS SDK v3.3.3

Update Date

- November 22, 2018

Version Update Notes

1. Added the support for rich media push to increase tap rate
2. Added the support for iOS 12 to make push delivery more stable
3. Added the support for batch account API to make push more flexible
4. Improved the stability of the SDK version

Upgrade Tips

Recommended to upgrade

Download Address

iOS SDK v3.3.3

iOS SDK v3.2.2

Update Date

- November 22, 2018

Version Update Notes

1. Improved the stability of the SDK version

©2013-2019 Tencent Cloud. All rights reserved.

Upgrade Tips

Recommended to upgrade

Download Address

iOS SDK v3.2.2

iOS SDK v3.3.1

Update Date

- September 11, 2018

Version Update Notes

1. Added the support for rich media notification bar push to increase tap rate
2. Added the support for iOS 12 to make push delivery more stable and increase arrival rate
3. Added the support for batch account API to make it easier to configure push targets

Upgrade Tips

Recommended to upgrade

Download Address

iOS SDK v3.3.1

iOS SDK v3.2.0

New features in the current SDK version:

Feature description	Prerequisite	API call method	Location in the console
Automatically increasing badge number by 1	SDK v3.1.0 and higher	Click here to view	Create a push > Notification bar message > General settings > Badge number

Update Date

- July 20, 2018

Version Update Notes

1. Added clearing and replacing APIs
2. Added the support for binding and unbinding multiple tags
3. Added the badge setting API to increase the badge number by 1

©2013-2019 Tencent Cloud. All rights reserved.

4. Added the statistics collection logic for single push

Download Address

iOS SDK v3.2.0

iOS SDK v3.1.1

Update Date

- May 3, 2018

Version Update Notes

1. Updated the account registration and unbinding protocol (fixed the issue where account registration and unbinding were unavailable in old versions)
2. Removed the support for simulator
3. Fixed the issue with the binding API
4. Fixed the issue with the unregistering API

Download Address

iOS SDK v3.1.1

iOS SDK v3.1.0

Update Date

- April 11, 2018

Version Update Notes

1. Added the message receipt statistics collection SDK (iOS 10+)
2. Automated device token reporting

Download Address

iOS SDK v3.1.0

©2013-2019 Tencent Cloud. All rights reserved.

Glossary

Last updated：2019-06-26 10:42:34

List of Applications

- Devices connected yesterday: The number of devices that were successfully connected to the TPNS server yesterday.
- Devices uninstalled yesterday: The number of devices on which the app was uninstalled yesterday. TPNS can count the actions of uninstalling the app and reports every single uninstallation, so the number of devices here is not deduplicated.
- Valid devices: The number of devices currently registered with TPNS.

App Configuration

- App package name: This is the package name of the app used for AndroidManifest.xml configuration, such as com.tencent.news. Push operations can be performed only after the app package name is entered, which cannot be modified once entered for security reasons.
- Admin: All admin accounts are granted the same permissions. An admin account can delete all other admin accounts but not itself. To replace your current admin account, we recommend adding a new account and using it to delete the old account.
- Test device: This is used to test push conditions on the specific device to make sure everything works as expected before pushing real messages. The ID of the test device is DeviceToken, which can be obtained through logcat. For details, see the developer manual.
- ACCESS KEY: This is the client authentication key, which verifies together with the Access ID to determine that the call is valid. It needs to be configured into the client SDK and cannot be modified.
- SECRET KEY: This is used to verify together with the APPKEY to determine that the API call is valid. If it is disclosed, you need to replace with a new one and reconfigure the client SDK immediately.

©2013-2019 Tencent Cloud. All rights reserved.

- ACCESS ID: This is the unique identifier of an app, which cannot be modified and needs to be configured into the client SDK. It has to be provided when a backend API is called.

Creating a Push

- Content Push: The message command is the code that is run after received by the app. The specific code form is defined by the app developer. Using the message command, you can remotely control various behaviors of the app, such as downloading the app and changing the splash screen, modifying item prices in a mobile game, or silently updating in-app text or image content.
- Immediate/scheduled push: Immediate push is more suitable for testing a push. Scheduled push is mostly used for regular push.
- Custom parameter: You can also set custom key-value pairs to implement custom requirements based on different key-value pairs.
- Offline push retention: Users always receive the push when they go back online within a certain period of time and before the customized expiration time. Users will not receive the expired push. For events that have no limited-time offers involved, it is highly recommended to retain the push for 72 hours.
- Time control: Set the time period during which a push can be received. This allows you to not disturb your users at night, or to specify when the user can receive your pushes.
- Action for notification tap: This is to set the response action after the user taps the notification, which can be launching the app directly, jumping to a specified function page of the app, or opening a URL using the browser.
- Multi-package name push: multi-package name prompt mode, all apps on the device that use this access_id to register the push service will receive the message. This feature is used to differentiate apps with different channels and package names.

Push List

- Push list: It displays the data of full/batch push (broadcast), but not the data of peer-to-peer (unicast) push. To view unicast push, go to the "Push data" page.

©2013-2019 Tencent Cloud. All rights reserved.

- Push time: The time this push starts.
- Android effective pushes: This refers to the number of pushes sent to currently online devices (i.e., the devices connected to the server). If offline retention is set for the message, the number of effective pushes will increase numerically over time as users go online, indicating that subsequent online users also receive this push.
- iOS effective pushes: This refers to the number of pushes successfully sent to Apple's server. TPNS is responsible for successfully pushing the message to Apple's server, but due to Apple's server restrictions, the number of arrivals cannot be counted.
- Arrivals: The number of users to whom this push is successfully sent. Real-time data collection has a delay of around 5 minutes.
- Undo scheduled task: This is to undo the scheduled push that has not yet occurred; after that, the push will not be executed.
- Delete offline message: This is to delete the offline message which is retained in the backend and will be sent. After the offline message is successfully deleted, the message will not be sent. However, messages that have already been successfully sent cannot be deleted.

Push Data

- Android effective pushes: This refers to the number of pushes sent to currently online devices (i.e., the devices connected to the server). If offline retention is set for the message, the number of effective pushes will increase numerically over time as users go online, indicating that subsequent online users also receive this push.
- iOS effective pushes: This refers to the number of pushes successfully sent to Apple's server. TPNS is responsible for successfully pushing the message to Apple's server, but due to Apple's server restrictions, the number of arrivals cannot be counted.
- Arrivals: The number of users to whom this push is successfully sent. Real-time data collection has a delay of around 5 minutes.
- Arrival rate: Arrivals/pushes × 100%

©2013-2019 Tencent Cloud. All rights reserved.

- Taps: The number of taps on the message after successfully pushed. The device needs to call a specific function to report the taps.
- Tap rate: Taps/arrivals × 100%

Basic Data

- Daily new devices: The number of devices newly registered today. The client SDK V2.3.6 or higher needs to be integrated.
- Daily uninstalled devices: The number of devices on which the app is uninstalled today. The client SDK V2.3.6 or higher needs to be integrated.
- Daily connected devices: The number of devices connected to the TPNS server today.

My Tags

- Tag: Labeling a user or a group of users. Custom tags can be set by calling the client or server and then used in the frontend.
- Advanced data tag: Directly labeling specific users by sorting data. Advanced tags can only be used and cannot be edited or deleted.

©2013-2019 Tencent Cloud. All rights reserved.

FAQs

Last updated：2019-06-26 09:53:38

Data Push Issues

The data push is paused

- Full push limitations (V2, V3):
- You can create up to 30 pushes per hour for full push, and excessive ones will fail.
- The full push of the same content can be performed only once per hour, and excessive ones will fail.
- Tag push limitations (V3):
- The same app can create up to 30 tag pushes per hour, and excessive ones will fail.
- The push of the same content with the same tag can be performed only once per hour, and excessive ones will fail.

Effect statistics

- Next day: The push data can be viewed the next day after pushed
- Real-time: The push data can be viewed immediately after pushed. Currently, up to 14 times of real-time statistics collection are supported per week.

Actually delivered pushes

- During the offline retention period of the message, there will be successful connections to the TPNS server and normally delivered pushes. For example, if the message is retained offline for 3 days, the actually delivered data will stabilize on the 4th day. Data will increase as more devices turn on and connect to the TPNS server.

Historical details

- Historical details only show full pushes, tag pushes, and number package pushes at the official website. Currently, batch accounts and batch devices do not show push details.

©2013-2019 Tencent Cloud. All rights reserved.

Data overview

- It shows the day's data, which is the day's total number of pushes from various push actions. There are four types of pushes, including notification bar message, in-app message, unicast, and broadcast, i.e., batch push and full push.

FAQs

Q: I received a prompt saying that the package of the number is invalid when I try to upload it to console?

A: If you compress a folder using the zip command on macOS, the hidden files will be compressed to o, making the console unable to recognize the zipped package. It is recommended to compress the f older on Windows or retry after deleting the hidden files on macOS.

Q: What if error 48, 10302, or 10303 is returned for account push?

A: Check whether the token is bound to the account. For more information, see [Account-device Bind ing Query](https://intl.cloud.tencent.com/document/product/1024/30742).

Q: Why can't the pushes be received on a Nubia phone?

A: At present, TPNS does not support Nubia phones released after 2015, because the new version of Nubia operating system has a super power-saving feature which will kill background processes very q uickly. Therefore, the TPNS service cannot be started normally, making it impossible to register succes sfully.

Q: What if the pushes can be received in the development environment on iOS but not in the production environment after packaging?

A:
1. Check whether the file archived-expanded-entitlements.xcent is present in the package.
If not, an error may occur:

No valid 'aps-environment' entitlement string found for application 'com.xxx.xxx': (null).

Solution: Go to TARGET -> Build Setting -> Code Signing Identity -> Code Signing Entitlements
Check whether there is the aps-environment field; if not, add it.
<plist version="1.0">
<dict>
<key>aps-environment</key>
<string>production</string>

©2013-2019 Tencent Cloud. All rights reserved.

```
</dict>
</plist>
```

2. If the device prompts for error "Error Domain=NSCocoaErrorDomain Code=3000" "No valid 'aps-environment' entitlement string found" or "UserInfo=0x16545fc0 {NSLocalizedDescription= No valid 'aps-environment' entitlement string found}":
Check whether the bundle ID configured in the Xcode project matches the configured Provision Profile file and whether the Provision Profile file corresponding to the app has been configured with the message push capability.

Q: Why is the iOS token invalid?

A:
(1) The system was logged out or the app was uninstalled
(2) The user installed the app on a new device
(3) The user restored the device from a **backup**
(4) The **user** reinstalled the OS
(5) Other **system**-defined **events** (1. **After calling** the API unregisterNotification, **register** the notification again **and clear** device **data and settings**)

Q: How to generate a pem certificate for TPNS?

A:
(1) Download the TPNS testing tool on the iOS app configuration page.
(2) Upload a p12 certificate **and** push a message successfully to APNs using a TPNS token.
(3) The tool will generate the pem certificate required **by** TPNS **in** the directory **of** the p12 certificate.

Q: Why can't the pushes be received in the production environment?

A: The production environment must meet the following testing conditions: The app is the ad-hoc/App Store build (**with** the **release** certificate "Production"), **and** the **release** certificate **is** uploaded **and** successfully verified.

Q: Is TPNS currently adapted to Android P?

A: V4.X is already compatible **with** Android P. HTTPS **is** used **by** default. **If** you want **to use HTTP**, you need **to** configure it **by** yourself ([Click here **to view** the configuration method](https://intl.cloud.tencent.com/**document**/product/1024/30723)).

Q: What if "[TPush] channelId is not initialized" appears in the log for v4.X?

A: This **is** an internal log **of** TPNS **and** does **not** affect the registration.

©2013-2019 Tencent Cloud. All rights reserved.

Q: What if "Server response error code:404, error:{"ret":-1, "msg":"invalid appkey"}" appears in the log for v4.X?

A: The MTA SDK is embedded in the TPNS SDK. This log is an MTA log and does not affect the registration in TPNS.

Q: What if the log prints out "otherpushToken = null" when I register with a vendor-specific channel?

A: 1. TPNS v4.X integrates the vendor-specific channels in a dynamic loading manner. When the app is launched for the first time, the vendor dex configuration package corresponding to the phone model will be downloaded. After the download is completed, the app process will be killed and the registration will be done when the app is restarted.
2. If the dex configuration package cannot be downloaded at all, you can use the non-dynamic loading method to integrate it. In this case, you need to use the TPNS v4.X jar without the vendor-specific channels and then integrate the jars of each vendor-specific channel. For more information about how to integrate, see the relevant document.

Q: After the Mi channel is integrated, there is no tap callback. How to achieve redirection to the specified page when the notification bar message is tapped?

A: The integrated vendor-specific channel must use the [intent](https://intl.cloud.tencent.com/document/product/1024/30720) method to redirect.

Q: Why can only one push message be displayed on a device that has integrated the Mi channel?

A: According to the documentation at Mi's official website, the notification bar only displays one push message by default. If you want multiple messages to be displayed in the notification bar, you need to set a unique notify_id for different messages (a new notification bar message with the same notify_id will override the previous one).

Q: Is there a limit to the number of messages displayed in the device notification bar? Why is a message not displayed after it arrives at the device?

A: There is no limit to the number of notification bar messages that a phone can receive and display. The reasons for not displaying may include:
(1) The notification bar on a Mi phone displays the latest message by default. If you want multiple messages to be displayed, you need to set a unique n_id for different messages.
(2) The message broadcast is blocked by a phone manager app.
(3) A Meizu phone has a message box, and uncommon messages will go directly to the message box. The messages can be viewed here.

Q: Why is there no historical record after a push is sent?

©2013-2019 Tencent Cloud. All rights reserved.

A: The details of **push** history displayed in the TPNS console include device **push**, full **push**, tag **push**, **and** number **package** push. Account list **push and** device group **push** through APIs will **not** be displa yed in the console.

Q: How to set a custom ringtone?

A:
1. In the console: Go to **Create** a push > Notification bar message > **Advanced settings** > Reminder **mode** > Customize (**for** Android, **select** the ringtone **file in** the **raw directory**, which does **not** need t he extension, such **as** xg_ring. **For** iOS, **select** the ringtone **file in** the bundle **directory**, which requires the extension, such **as** xg_ring.wav).
2. **In** Rest API V3: **For** Android, **set** ring=1 **in** the push message **body**, **and set** ring_raw **to** the **name o f** the specified ringtone **file without** the extension **in** the **raw directory of** the Android project. **For** i OS, **set** the sound **in** the push message **body to** the **name of** the specified ringtone **file with** the exte nsion **in** the bundle **directory of** the project.
Note: **If** the **client** has integrated a vendor-specific channel, due **to** the limitations **of** Huawei **and** Me izu, custom ringtone **file is not** allowed **on** their phones, **and** the **system** sound will be used **by** defa ult. Currently, custom ringtone can be used **on** Mi phones.

Q: How to customize the icon in the status bar? Why is the icon gray?

A: ROMs running native Android 5.0 or above will process icon of an app **with** the **value of** target sdk larger **than or** equal **to** 21 **by** adding a layer **of** color **to** make it gray. Therefore, **if** you want color ico ns, **set** the **value of** target sdk below 21. However, **if** you need **to** have the **value of** target sdk **no** sma ller **than** 21, **rename** a png formatted image **with** a **clear** background **to** notification_icon.png **and** pl ace it **in** drawable; in this way, the icon will be in gray and shaped.

Q: Can an app still receive push messages after it is closed or its process is ended?

A: TPNS mainly uses the TPNS service **to** push and receive **messages**. When the process **is** killed, the TPNS service will also **be** killed.
Only after the service **is** pulled back or the app **is** restarted can the **messages be** received and pushe d. If another app accessing TPNS **is** launched **on** the phone, **messages** can **be** received and pushed u sing the service of that app.
However, the shared service channel **is** also limited by the phone's ROM, and 100% success rate cann ot be guaranteed.

Q: How many offline messages can be retained for a single device? And how long can they be retained?

A: TPNS **only** retains two offline messages: If the device receives multiple **messages** when it **is** offline, TPNS will **only** retain the **last** two ones, and the **previous** ones will **be** lost. When the device goes bac k online, **only** the **last** two **messages** can **be** received.

©2013-2019 Tencent Cloud. All rights reserved.

Q: What if Android v4.4.4 prompts a compiling error where XGPushProvider and MidProvider can be found?

A: If the number of loading methods in the project exceeds 65 K, please create sub packages for the project.

Q: What are the restrictions on tags?

A: A maximum of 100 tags can be set for a single device, and one single app can have up to 10,000 different tags.

Q: After the initial registration is successful, no unregistration is performed. Do I need to register again afterward?

A: No, as long as no unregistration is performed, you do not need to register again.

Q: Why can't callback information be received for device registration?

A: In the registration operation, there are only three types of errors in the backend:

(1) No response;

(2) A data packet in a wrong format is returned;

(3) An error code is returned. The device should be able to detect all these three types of errors and give a callback accordingly.

Q: What is the difference between token and account?

A: Token is the identifier of the app for receiving the push message (device); while account is the identifier of a user.

Q: If an account is logged in to on device A and then on device B, what happens when a message is sent to this account?

A: Device B can receive the push, but device A cannot, as only the last device bound to the account can receive pushes.

Q: What is the difference between tag and account?

A: A tag is used to identify a token or a user's attributes, such as Guangdong, male, and gamers. An account is the user's account. Please do not use tags as aliases.

©2013-2019 Tencent Cloud. All rights reserved.

Q: What should I do if I am not able to redirect to another page when I open a page for an activity?

A: On some phones, when users click the icon of the app on the Notification bar, users may not be redirected to that app due to permission limits.
Solution: In AndroidManifest.xml, add android:exported="true" to the activity to be opened.

Q: What does the "number of covered devices" in the app list mean?

A: It refers to the number of devices on which the app is in registered status. It is also the maximum number of devices that the app can cover when pushing a message. If a device calls the unregister API for unregistration, the number of covered devices will decrease.

Q: Why are there many platform-specific .so files in the libs directory, such as armabi and x86?

A: TPNS has developed the .so libraries for all Android platforms.
Solution: You can delete the unwanted platform-specific directories. For example, if your game only has armabi, you can delete other directories.

Q: Why is the server busy when I push data to the website?

A: Please check whether the token and the selected push environment are correct, then check whether the certificate has been submitted correctly. If the error persists, you can create a new certificate without a password, submit it, and retry.

Q: Can a non-scheduled push (i.e., immediate push) be canceled during the pushing process?

A: No. Only tasks that return the push_id can be canceled.

Q: When I check the push list after performing a push task, it shows that the push has been completed, but its status is displayed as "pushing". What should I do?

A: The webpage's status is delayed. Please refresh and retry.

Q: What is the order in which multiple push messages are received after the user goes back online?

A: They are in the ascending order by message ID. The client also receives messages according to this rule; there, the order in which the messages are received is the same as that in which they are sent.

Q: I have Android users and iOS users. Do I have to write two different APIs in the PHP backend for pushes to users on different platforms?

A: You need to call two push APIs to encapsulate them as one.

©2013-2019 Tencent Cloud. All rights reserved.

Q: If the time selected for a scheduled push is in the past, will the push be sent?

A: If the **time** selected is in the past, the **system** will **send** the **push** immediately.

Q: Why is there only sound but no text when a push notification arrives?

A: This issue is related **with** the system. You need **to analyze** the cause **using** the logcat **of** the device.

Q: Can TPNS push messages to regions outside mainland China?

A: **As** long **as** the TPNS server's domain name (openapi.xg.qq.com) is pingable, push messages can be received. The global server of TPNS is deployed in Hong Kong, but due to high network latency in overseas regions, the push effect of TPNS is slightly inferior to that in mainland China.

Q: Is the APPID data of TPNS and the Tencent Open Platform interconnected?

A: After you register your app on the open platform and **use** TPNS, the app information will be automatically synchronized **from** the **open** platform **to** the TPNS platform, so that you don't need to register the app again when using TPNS alone.
However, apps registered with TPNS won't be synchronized **to** the **open** platform.

Q: Can TPNS be used if there is no SD card?

A: It can **be** used, but the location where the log **is** stored **is** different.

Q: Can the registration method be created in the thread? Can it be created in APPLICATON onCreate?

A: The registration method can **be** called anywhere, but the applicationContext needs **to be** passed.

Q: How to delete the Toast prompt for successful registration?

A: The CustomPushReceiver in the demo comes **with** a Toast **prompt** processing method, which **is to delete** the **content of** the Toast inside CustomPushReceiver.

Issues with Payments

Q: What if my order cannot be placed as the system prompts that there is already a VIP package corresponding with the access_id?

A: Please log in **to** the TPNS console, **go to** your app, and select "App configuration" > "App information" **to** check whether there **is a** VIP package that has not been renewed. If yes, please renew it direct

©2013-2019 Tencent Cloud. All rights reserved.

ly.

Q: What if my order cannot be placed as the system prompts that the access_id is invalid?

A: Please log in to the TPNS console, go to your app, and select "App configuration" > "App information" to check whether the access_id is correct.

Q: What should I do if the payment cannot go through?

A: Please contact your account manager or customer service.

Q: What should I do if the payment went through but I am not able to upgrade the service?

A: Please contact your account manager or customer service.

©2013-2019 Tencent Cloud. All rights reserved.

Technical Support

Last updated：2019-06-26 09:53:46

Business consultancy: data@tencent.com

©2013-2019 Tencent Cloud. All rights reserved.