

Tencent Push Notification Service

REST API V3

Product Documentation



Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

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

Trademark Notice

 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

REST API V3

- REST API Overview

- Tag API

- Push API

- Account API

REST API V3

REST API Overview

Last updated : 2019-06-26 10:01:32

Differences Between V3 and V2

- V3 is fully based on HTTPS and no longer provides HTTP access.
- V3 supports POST access and no longer provides GET access.
- V3 uses HTTP Basic Authentication for access authorization. API requests can be done using common HTTP tools such as curl and Postman.
- In V3, APIs are categorized according to business functions, including Push API, Tag API, Device API, and Tool API. The same URL is used for all APIs in one category to improve API usability.
- V3 has improvements in functionality. A tool API for arrival statistics query is added, which is currently under beta test.

For V2, see the documentation at <https://intl.cloud.tencent.com/document/product/1024/30744>.

Note: It is recommended to use the V3 APIs, and V2 APIs may be discontinued in the future.

API Overview

TPNS provides REST-compliant HTTP APIs for developers to remotely call the services provided by it.

The APIs are mainly divided into four categories:

API type	Description	Status
Push API	It includes various APIs used to push messages	Normal
Tag API	It includes various APIs used to add, delete, and query tags	Normal
Device API	It includes various APIs used to query and delete account	Implementing
Tool API	It includes various APIs used to locate problems and query other data	Implementing

Request Method

- HTTPS is supported
- POST is supported

Authentication Method

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

Authorization: Basic base64_auth_string

- The generation algorithm for base64_auth_string is: base64(APPID:SECRETKEY)

That is to add a colon after APPID followed by SECRETKEY to form a string and then perform base64 encoding on the string.

- APPID and SECRETKEY can be obtained in **My apps > App configuration** at the [TPNS website](#).

Protocol Description

Request URL: https://openapi.xg.qq.com/v3/<class_path>/<method>

Field name	Usage	Note
openapi.xg.qq.com	API domain name	None
v3	Version number	None
class_path	Category of the API provided	Different APIs have different path names
method	Feature API name	Different feature APIs have different names

API Limitations

- For [Push API](#), the APIs for full push and tag push have limitations on call frequency. The default value is 30 times per ACCESS_ID per hour.
- The size of the pushed message body cannot exceed 4 KB, and this limitation applies to the message field in Push API.

Reference

- [Glossary](#)
- [Standard HTTP Error Codes](#)

Tag API

Last updated : 2019-06-26 10:01:40

Tag API Overview

- Tag API is a general term for all tag APIs
- Tag API includes various APIs used to set, update, and delete tags as follows:
 - Add a single tag
 - Delete a single tag
 - Add multiple tags
 - Delete multiple tags
 - Delete all tags
 - Update a tag
 - Set a tag for multiple tokens
 - Delete a tag from multiple tokens
 - Batch set tags
 - Batch delete tags
- All push targets use the same URL to initiate the request (URL: <https://openapi.xg.qq.com/v3/device/tag>).
- All request parameters are uploaded via JSON encapsulation to the backend, which distinguishes among different push targets based on the request parameters.

Tag API Call Address

<https://openapi.xg.qq.com/v3/device/tag>

Tag API Request Parameters

Parameter name	Type	Required	Description
operator_type	int	Yes	Operation type 1) 1: add a single tag to a single token 2) 2: delete a single tag from a single token 3) 3: add multiple tags to a single token 4) 4: delete multiple tags from a single token 5) 5: delete all tags from a single token 6) 6: add one or more tags to a single token in an overriding manner 7) 7: add a single tag to multiple tokens 8) 8: delete a single tag from multiple tokens 9) 9: batch add tags (up to 20 pairs are allowed per call; in each tag-token pair, the tag is before the token) 10) 10: batch delete tags (up to 20 pairs are allowed per call; in each tag-token pair, the tag is before the token)
platform	string	Yes	Client platform type 1) android: Android 2) ios: Apple

Parameter name	Type	Required	Description
token_list	array	No	Device list 1) Required when operator_type =1,2,3,4,5,6,7,8 2) When operator_type =1,2,3,4,5,6, if this parameter contains multiple tokens, only the first one will be set 3) Format example: ["token1","token2"] 4) The list can contain up to 20 values 5) A token string cannot exceed 64 characters
tag_list	array	No	Tag list 1) Required when operator_type =1,2,3,4,6,7,8; ignored when =5 2) When operator_type =1,2,3,4,6,7,8, if this parameter contains multiple tags, only the first tag will be set if for a single tag 3) Format example: ["tag1","tag2"] 4) The list can contain up to 20 values 5) A tag string cannot exceed 50 characters
tag_token_list	array	No	Tag-device list 1) Required when operator_type =9,10 2) Format example: [{"tag1","token1"}, {"tag2","token2"}] 3) In each pair, the tag must be before the token 4) The list can contain up to 20 values 5) A tag string cannot exceed 50 characters 6) A token string cannot exceed 64 characters
seq	int64_t	No	When the API is called, TPNS will return this field in the response packet, which can be used for async request Usage scenario: In async service, the corresponding response packet returned by the server can be found through this field
op_type	string	No	API operator type: qq, rtx, email, other
op_id	string	No	API operator type: API operator id (qq\rtx\email)

Tag API Samples

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

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

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

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

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

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

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

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

- Delete all tags from a single token (token1)

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

- Add multiple tags (tag1 and tag2) to a single token (token1) in an overriding manner

```
{
  "operator_type": 6,
  "platform": "android",
  "tag_list": ["tag1","tag2"],
  "token_list": ["token1"]
}
```

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

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

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

```
{
  "operator_type": 8,
  "platform": "android",
  "tag_list": ["tag1"],
}
```



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

- Batch set tags for [tag1,token1],[tag2,token2]

```
{
  "operator_type": 9,
  "platform": "android",
  "tag_token_list": [{"tag1","token1"}, {"tag2","token2"}]
}
```

- Batch delete tags for [tag1,token1],[tag2,token2]

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

Tag API Response Parameters

Field name	Type	Required	Comments
seq	int64_t	Yes	The same as the request packet (if the request packet is invalid JSON, this field is 0)
ret_code	int32_t	Yes	Error code; for details, see the error codes table
err_msg	string	No	error message when an error occurs in the request
result	string	No	When the request is correct, if there is extra data to be returned, the result will be encapsulated in the json of this field. If there is no extra data, there may be no such field

Complete Example of Tag API Request

Tag Setting Request Message

```
POST /v3/device/tag HTTP/1.1
Host: openapi.xg.qq.com
Content-Type: application/json
Authorization: Basic YTViNWYwNzFmZjc3YTplYTUxMmViNzcxNGQ1Zml1YTZhOTM3Y2FmYTcwZTc3MQ==
Cache-Control: no-cache
Postman-Token: 4b82a159-afdd-4f5c-b459-de978d845d2f
{
  "operator_type": 1,
  "platform": "android",
  "tag_list": ["tag1"],
  "token_list": ["token1"]
}
```

Tag Setting Response Message

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

Error Codes

You may encounter various problems when using APIs. Below are the common error codes and their definitions:

Error code	Meaning
0	Normal
10000	Unknown exception
10001	Failed due to timeout. Please retry
10102	Parameter is missing. Please check and retry
10103	Parameter value is invalid. Please check and retry
10104	Authentication failed. Please check the secret key
11001	Internal error. Please retry later
11002	Internal error. Please retry later
11003	Internal error. Please retry later
11004	Internal error. Please retry later
11005	Internal error. Please retry later
11006	Internal error. Please retry later
11007	Internal error. Please retry later
10113	Internal error. Please retry later

Push API

Last updated : 2019-06-26 10:01:48

Push API Overview

- Push API is a general term for all push APIs
- Push API has a variety of push targets:
 - Full push
 - Tag push
 - Single-device push
 - Device list push
 - Single-account push
 - Account list push
- All push targets use the same URL to initiate the request (URL: `https://openapi.xg.qq.com/v3/push/app`).
- All request parameters are uploaded via JSON encapsulation to the backend, which distinguishes among different push targets based on the request parameters.

Call Address

`https://openapi.xg.qq.com/v3/push/app`

Required Push API Parameters

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

Parameter name	Type	Required	Description
audience_type	string	Yes	Push target 1) all: full push 2) tag: tag push 3) token: single-device push 4) token_list: device list push 5) account: single-account push 6) account_list: account list push
platform	string	Yes	Client platform type 1) android: Android 2) ios: Apple iOS 3) all: Android and iOS, which is supported only for full push and tag push (reserved and not available yet)
message	object	Yes	Message body; for more information, see Message Body

Parameter name	Type	Required	Description
message_type	string	Yes	Message type 1) notify: notification 2) message: passthrough/silent message

audience_type: Push Target

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

Push API provides a variety of push targets, such as all, single device, device list, single account, and account list.

Push target	Description	Usage
all	Full push	None
tag	Tag push	<p>tag_list</p> <p>1) Push to devices with tag1 and tag2 { "tags": ["tag1", "tag2"], "op": "AND" }</p> <p>2) Push to devices with tag1 or tag2 { "tags": ["tag1", "tag2"], "op": "OR" }</p> <p>3) Tag list cannot exceed 512 characters</p>
token	Single-device push	<p>token_list</p> <p>1) If the parameter contains multiple tokens, it will only push to the first token</p> <p>2) Format example: ["token1"]</p> <p>3) A token string cannot exceed 64 characters</p>
token_list	Device list push	<p>token_list</p> <p>1) Up to 1,000 tokens</p> <p>2) Format example: ["token1", "token2"]</p> <p>3) token string cannot exceed 64 characters</p> <p>push_id</p> <p>1) If 0 is entered for this parameter during the first push, the system will create the corresponding push task and return pushid: 123</p> <p>2) In subsequent pushes, if you enter 123 for push_id (with the same text copy), it indicates to use the text copy corresponding to 123 id for the push</p>
account	Single-account push	<p>account_list</p> <p>1) If the parameter contains multiple account, it will only push to the first account</p> <p>2) Format example: ["account1"]</p>
account_list	Account list push	<p>account_list</p> <p>1) Up to 1,000 accounts 2) Format example: ["account1", "account2"]</p> <p>push_id</p> <p>1) If 0 is entered for this parameter during the first push, the system will create the corresponding push task and return pushid: 123</p> <p>2) In subsequent pushes, if you enter 123 for push_id (with the same text copy), it indicates to use the text copy corresponding to 123 id for the push</p>

- Full push: Push to all devices

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

- Tag push: Push to devices with both "tag1" and "tag2"

```
{
  "audience_type": "tag",
  "tag_list": {
    "tags": [
      "tag1",
      "tag2"
    ],
    "op": "AND"
  }
}
```

- Single-device push: Push to the device with token "token1"

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

- Device list push: Push to devices with token "token1" and "token2"

```
{
  "audience_type": "token_list",
  "token_list": [
    "token1",
    "token2"
  ],
  "push_id": "0"
}
```

- Single-account push: Push to device with account "account1"

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

- Account list push: Push to devices with account "account1" and "account2"

```
{
  "audience_type": "account_list",
  "account_list": [
    "account1",
    "account2"
  ],
  "push_id": "0"
}
```

platform: Push Platform

Currently, TPNS supports push for Android and iOS platforms.

The keywords are "android", "ios". If you want to push for both platforms at the same time, the keyword is: "all".

- Push to both platforms:

```
{
  "platform": "all"
}
```

- Push to the Android platform:

```
{
  "platform": "android"
}
```

- Push to the iOS platform:

```
{
  "platform": "ios"
}
```

message_type: Message Body Type

For different platforms, the message types are slightly different; for details, see the table below:

Message type	Description	Supported platform	Feature description
notify	Notification bar message	Android, iOS	Message is displayed in the notification bar
message	Passthrough/silent message	Android (passthrough message) iOS (silent message)	Message is not displayed in the notification bar

message: Message Body

The message body, i.e., the message delivered to the client.

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

General Message on Android

The specific fields for the Android platform are as follows:

Field name	Type	Default value	Required	Parameter description
title	string	None	Yes	Message title
content	string	None	Yes	Message content

Field name	Type	Default value	Required	Parameter description
accept_time	array	None	No	This specifies during which time periods the message will be pushed to users; it is recommended to specify less than 10 ones; and this parameter cannot be null
xg_media_resources	string	None	No	Rich media element addresses; up to 5 ones recommended (only for SDK v4.2.0 and higher)
n_id	int	0	No	Unique identifier of the notification message object (only valid for the TPNS channel) 1) If greater than 0, it will override the previous message with the same id 2) If 0, it will display this notification and not affect other messages 3) If -1, it will clear all previous messages and only display this message
builder_id	int	0	No	Local notification style identifier
ring	int	1	No	Whether there a ringtone 1) 0: No 2) 1: Yes
ring_raw	string	None	No	This specifies the name of the ringtone file in the raw directory of the Android project; no extension is needed
vibrate	int	1	No	Whether to use vibration 1) 0: No 2) 1: Yes
lights	int	1	No	Whether to use breathing light 1) 0: No 2) 1: Yes
clearable	int	1	No	Whether the notification in the notification bar can be dismissed
icon_type	int	0	No	Whether the notification bar icon is an in-app icon or an uploaded icon 1) 0: In-app icon 2) 1: Uploaded icon
icon_res	string	None	No	In-app icon file name or URL address of the downloaded icon
style_id	int	1	No	This specifies whether to override the notification style with the specified number
small_icon	string	None	No	The icon that the message displays in the status bar. If not set, the app icon will be displayed
action	JSON	Yes	No	This sets the action after the notification bar is tapped; the default action is to open the app
custom_content	JSON	None	No	User-defined key-value pair

Below is an example of a complete message:

```
{
  "title": "xxx",
  "content": "xxxxxxxxxx",
  "xg_media_resources": "xxx1" , // Enter the rich media element address, such as https://www.xx.com/img/bd_logo1.png?qua=high
```

```
"accept_time": [
{
"start": {
"hour": "13",
"min": "00"
},
"end": {
"hour": "14",
"min": "00"
}
},
{
"start": {
"hour": "00",
"min": "00"
},
"end": {
"hour": "09",
"min": "00"
}
}
],
"android": {
"n_id": 0,
"builder_id": 0,
"ring": 1,
"ring_raw": "ring",
"vibrate": 1,
"lights": 1,
"clearable": 1,
"icon_type": 0,
"icon_res": "xg",
"style_id": 1,
"small_icon": "xg",
"action": {
"action_type": 1, // Action type; 1. Open activity or app; 2. Open browser; 3. Open Intent
"activity": "xxx",
"aty_attr": { // activity attribute, only for action_type=1
"if": 0, // Intent's Flag attribute
"pf": 0 // PendingIntent's Flag attribute
},
"browser": {
"url": "xxxx ", // Only http and https are supported
"confirm": 1 // Whether user confirmation is required
},
"intent": "xxx" // The SDK version must be 3.2.3 or higher. Configure the data tag in the client's intent and set the scheme attribute
},
"custom_content": {
"key1": "value1",
"key2": "value2"
}
}
}
```

General Message on iOS

The specific fields for the iOS platform are as follows:

Field name	Type	Default value	Required	Parameter description
aps	JSON	None	Yes	APNs-specific message body field, where important key-value pairs include: alert: It contains the title and message content (required) badge_type: The badge number displayed by the app (optional) category: The action identifier displayed when the message is pulled down (optional) For details, see Payload
custom	string/JSON	None	No	Custom delivery parameter
xg	string	None	No	key reserved by the system, which should not be used

Below is an example of a complete message:

```
{
  "title": "xxx",
  "content": "xxxxxxxxxx",
  "ios":{
    "aps": {
      "alert": {
        "subtitle": "my subtitle"
      },
      "badge_type": 5,
      "category": "INVITE_CATEGORY"
    },
    "custom1": "bar",
    "custom2": [
      "bang",
      "whiz"
    ],
    "xg": "oops"
  }
}
```

Passthrough Message on Android

Passthrough message is unique to the Android platform and not displayed in the notification bar of the mobile phone. It can be used to deliver messages with control information to users in an imperceptible manner.

The specific fields for the Android platform are as follows:

Field name	Type	Default value	Required	Parameter description
title	string	None	Yes	Message title
content	string	None	Yes	Message content
custom_content	JSON	None	No	Custom content
accept_time	array	None	No	This specifies during which time periods the message will be pushed to users; and it is recommended to specify less than 10 ones

Complete example:

```

{
  "title": "this is title",
  "content": "this is content",
  "android": {
    "custom_content": {
      "key1": "value1",
      "key2": "value2"
    }
  },
  "accept_time": [
    {
      "start": {
        "hour": "13",
        "min": "00"
      },
      "end": {
        "hour": "14",
        "min": "00"
      }
    },
    {
      "start": {
        "hour": "00",
        "min": "00"
      },
      "end": {
        "hour": "09",
        "min": "00"
      }
    }
  ]
}

```

Silent Message on iOS

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

The specific fields are as follows:

Field name	Type	Default value	Required	Parameter description
aps	JSON	None	Yes	APNs-specific field, where the most important key-value pair is as follows content-available: It identifies the message type (which must be 1) and cannot contain alert, sound, or badge_type fields For details, see Payload
custom	string/JSON	None	No	Custom delivery parameter
xg	string	None	No	key reserved by the system, which should not be used

Complete example:

```

{
  "ios":{

```

```

"aps": {
  "content-available": 1
},
"custom": {
  "key1": "value1",
  "key2": "value2"
},
"xg": "oops"
}
}

```

Optional Push API Parameters

Optional Push API parameters are the optional advanced parameters except `audience_type` , `platform` , `message_type` , and `message` .

Parameter name	Type	Required	Default value	Description
<code>expire_time</code>	int	No	259200 (72 hours)	Offline message retention duration (in seconds), up to 72 hours 1) If <code>expire_time=0</code> , it indicates real-time message 2) If <code>expire_time</code> is greater than 0 and less than 800 seconds, the system will reset it to 800 seconds 3) If <code>expire_time >= 800</code> seconds, the message will be retained according to the actual set duration, up to 72 hours 4) The value set cannot exceed 2147483647; otherwise, the push will fail
<code>send_time</code>	string	No	Current system time	This specifies the push time 1) The format is yyyy-MM-DD HH:MM:SS 2) If it is smaller than the current server time, the message will be pushed immediately 3) This field is supported only for full push and tag push
<code>multi_pkg</code>	bool	No	false	Multi-package name push 1) For an app that has multiple channel packages (such as for MyApp and Wandoujia), if you want the mobile phone to receive the push message no matter what channel is used for app installation, then you need to set this value to true
<code>loop_times</code>	int	No	0	Task repetitions 1) This is supported only for full push and tag push 2) Recommended value range: [1, 15]
<code>loop_interval</code>	int	No	0	Interval for looping message delivery 1) It must be used together with <code>loop_times</code> 2) In days, value range: [1, 14] 3) <code>loop_times</code> and <code>loop_interval</code> together indicate the loop rule for the message delivery task
<code>environment</code>	string	Yes (only for iOS)	product	This specifies the push environment (only available for pushes on iOS) 1) product: Production push environment 2) dev: Development push environment
<code>badge_type</code>	int	No	-1	This specifies the badge number in the <code>aps</code> field (only for iOS) 1) -1: Badge number does not change 2) -2: Badge number is automatically increased by 1 3) <code>>=0</code> : Set "custom" badge number

Parameter name	Type	Required	Default value	Description
stat_tag	string	No	None	Statistics tag for aggregated statistics Usage scenario (example): There is now one event id: active_picture_123, and you need to deliver a message via the single push API (or list push API) to 10,000 devices, and set this field to active_picture_123. After the push is completed, you can use the v3 statistics querying API to query the actual deliveries, arrivals, displays and taps for the 10,000 devices according to the active_picture_123 tag
seq	int64_t	No	0	When the API is called, TPNS will return this field in the response packet, which can be used for async request Usage scenario: In async service, the corresponding response packet returned by the server can be found through this field
tag_list	object	Only required by tag push	None	1) Push to devices with tag1 and tag2: {"tags":["tag1","tag2"],"op":"AND"} 2) Push to devices with tag1 or tag2: {"tags":["tag1","tag2"],"op":"OR"}
account_list	array	Only required by single-account push and account list push	None	For single-account push 1) audience_type must be account 2) Parameter format: ["account1"] For account list push 1) Parameter format: ["account1","account2"] 2) Up to 1,000 accounts
account_push_type	int	Optional for single-account push	0	1) 0: Push message to the latest device of the single account 2) 1: Push message to all device associated with the single account
account_type	int	Optional for single-account push	0	1) Account type (see the account description below) 2) It must be the same as the account type set when the account is bound
token_list	array	Required by single-device push and device list push	None	For single-device push 1) audience_type must be token 2) Parameter format: ["token1"] For device list push 1) Parameter format: ["token1","token2"] 2) Up to 1,000 tokens

Parameter name	Type	Required	Default value	Description
push_id	string	Required by account list push and device list push	None	For account list push and device list push, if 0 is entered for this parameter during the first push, the system will create the corresponding push task and return the corresponding pushid: 123. In subsequent pushes, if you enter 123 for push_id (with the same text copy), it indicates to use the text copy corresponding to 123 id for the push. (Note: The validity period of the text copy is determined by the expire_time field above)

Push API Response Parameters

Field name	Type	Required	Comments
seq	int64_t	Yes	The same as the request packet (if the request packet is invalid JSON, this field is 0)
push_id	string	Yes	Push id
ret_code	int32_t	Yes	Error code; for details, see the error codes table
environment	string	Yes	This specifies the push environment (only for iOS) product: Production environment dev: Development environment
err_msg	string	No	error message when an error occurs in the request
result	string	No	When the request is correct, if there is extra data to be returned, the result will be encapsulated in the json of this field. If there is no extra data, there may be no such field

Complete Example of Push API Request

Android Tag Push Request Message

```
POST /v3/push/app HTTP/1.1
Host: openapi.xg.qq.com
Content-Type: application/json
Authorization: Basic YTViNWYwNzFmZjc3YTplYTUxMmViNzcxNGQ1Zml1YTZhOTM3Y2FmYTcwZTc3MQ==
Cache-Control: no-cache
Postman-Token: 4b82a159-afdd-4f5c-b459-de978d845d2f
{
  "platform": "android",
  "audience_type": "tag",
  "tag_list": {
    "tags": [
      "tag1",
      "tag2"
    ],
    "op": "AND"
  },
  "message_type": "notify",
  "message": {
    "title": "this is title",
    "content": "this is content",
    "custom_content": {
      "key1": "value1",
      "key2": "value2"
    }
  }
}
```

```
"accept_time": [  
  {  
    "start": {  
      "hour": "13",  
      "min": "00"  
    },  
    "end": {  
      "hour": "14",  
      "min": "00"  
    }  
  }  
]
```

Tag Push Response Message

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

iOS Single-device Push Request Message

```
POST /v3/push/app HTTP/1.1  
Host: openapi.xg.qq.com  
Content-Type: application/json  
Authorization: Basic ODhjNzE1Mzc1MDQ0ZDowNGM4NmNhZmI0ZTMxZDU4M2UzYjg0M2VhMDc4YTU5ZQ==  
Cache-Control: no-cache  
Postman-Token: 71670b5b-3149-4883-a427-d75cf9f42188
```

```
{  
  
  "platform": "ios",  
  "audience_type": "token",  
  "environment": "dev",  
  "token_list": [ "55c2ddbba664e9abcacea7daab0887939893b18e1a2cf4475c5382f5bcb2ab25b"],  
  "message_type": "notify",  
  "message": {  
    "title": "xxx",  
    "content": "https://xg.qq.com/docs/ios_access/ios_faq.html",  
    "ios": {  
      "aps": {  
        "alert": {  
          "subtitle": "my subtitle"  
        },  
        "badge_type": -2,  
        "sound": "Tassel.wav",  
        "category": "INVITE_CATEGORY"  
      }  
    },  
    "custom1": "bar",  
    "custom2": [  
  ]  
}
```

```
"bang",
"whiz"
],
"xcg": "oops"
}
}
}
```

Single-device Push Response Message

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

Account Type

The account type is bound by calling the SDK API by the client. The types are as shown in the table below:

Account type	Meaning
0	Unknown
1	Mobile number
2	Email
1000	WeChat openid
1001	QQ openid
1002	Sina Weibo
1003	Alipay
1004	Taobao
1005	Douban
1006	Facebook
1007	Twitter
1008	Google
1009	Baidu
1010	JD
1011	LinkedIn
1999	Other
2000	Visitor login

Account type	Meaning
2001 or above	User-defined

Error Codes

There are many REST APIs in TPNS. You may encounter various problems when using them. Below are the common error codes and their definitions, which correspond to the `ret_code` field in the [General Basic Return Values](#).

Error code	Meaning
10100	System busy. Please retry later
10101	System busy. Please retry later
10102	Parameter is missing. Please check and retry
10103	Parameter value is invalid. Please check and retry
10104	Authentication failed. Please check the secret key
10105	Invalid certificate
10106	The current push type does not support multi-platform push
10107	Message body is in invalid JSON format
10108	Internal error. Please retry later
10109	Internal error. Please retry later
10110	Device not registered
10111	Internal error. Please retry later
10112	Internal error. Please retry later
10113	Internal error. Please retry later
10114	Internal error. Please retry later
10115	Account cannot be empty
10116	The account does not exist
10117	Push content too large
10201	Failed to create push task. Please retry
10202	Failed to convert the push message content to APNs
10203	Failed to create push task. Please retry
10204	Push failed. Please retry later
10205	Push task expired. Please check
10206	Failed to get a copy of the message. Please retry later
10207	Failed to get a copy of the message. Please retry later

Error code	Meaning
10301	Account list push failed. Please retry
10302	Account list push partially failed. Please check whether the failed accounts have been bound to devices
10303	Account list push completely failed. Please check whether the accounts have been bound to devices
10304	Token List push partially failed. Please check whether the failed devices have been normally registered
10305	Token List push completely failed. Please check whether the devices have been normally registered
10401	Internal error. Please retry later
10402	Internal error. Please retry later
10403	Internal error. Please retry later
10404	Internal error. Please retry later
10405	Internal error. Please retry later
10406	Internal error. Please retry later
10407	Internal error. Please retry later
10501	Internal error. Please retry later
10502	Internal error. Please retry later
10503	Internal error. Please retry later
10504	Internal error. Please retry later
10505	Internal error. Please retry later
10506	Internal error. Please retry later
10507	Internal error. Please retry later
10601	Internal error. Please retry later
10602	Internal error. Please retry later
10603	Internal error. Please retry later
10604	Internal error. Please retry later
10605	Internal error. Please retry later
10606	App not registered. Please register and retry
10701	Internal error. Please retry later
10702	Internal error. Please retry later
10707	Internal error. Please retry later
10708	Internal error. Please retry later
10709	Internal error. Please retry later
10710	Internal error. Please retry later

Error code	Meaning
10711	Internal error. Please retry later
10712	Internal error. Please retry later
10713	Internal error. Please retry later
Other	Unknown error. Please retry later

Account API

Last updated : 2019-06-26 10:01:57

Glossary

Name	Field	Description
Account	account	It refers to the user's app login account. One login account can be bound to multiple device tokens. The character length of an account must be greater than 1 byte.
Device	token	It refers to the user's device ID. One device can be bound to multiple accounts. The character length of a token must be between 1 and 64 bytes.
Account type	account_type	It is used to identify the account type. *For specific type value, see the table.

- [Account Type Value Table](#)

Account API Overview

- Account API is a general term for all account APIs
- It is mainly divided into account operation APIs and account query APIs as follows:
 - Bind token to an account
 - Bind token to an account in an overriding manner
 - Unbind token from an account
 - Unbind token from all accounts
 - Unbind account from all tokens
 - Batch query the token relationships bound to accounts
 - Batch query the account relationships bound to tokens
 - Batch query the tokens bound to accounts
 - Batch query the accounts bound to tokens

Account API Request Description

Account Binding and Unbinding (Batch Operation)

API Description

POST <https://openapi.xg.qq.com/v3/device/account/batchoperate>

Async API. The API is only responsible for task delivery. Currently, it does not support real-time operations.

Parameter Description

Parameter name	Type	Required	Description
operator_type	int	Yes	Operation type 1: Bind token to an account 2: Bind token to an account in an overriding manner 3: Unbind token from multiple accounts 4: Unbind token from all accounts 5: Unbind account from all tokens

Parameter name	Type	Required	Description
platform	string	Yes	Client platform type 1) android: Android 2) ios: Apple
account_list	jsonArray	No	The collection of account IDs, which is valid when operator_type=5, and each element is required, including account and account_type fields. Example: [{"account": "926@126.com", "account_type": 2}, {"account": "1527000000", "account_type": 1}]
token_list	jsonArray	No	The collection of device IDs, which is valid when operator_type=4 and required
token_accounts	jsonArray	No	It is valid when operator_type=1, 2, or 3 and must be called each time. Up to 20 tokens are allowed, and each token_account consists of 1 token and 1 account_list. Below is a specific example: [{"token": "token1", "account_list": [{"account": "926@126.com", "account_type": 2}, {"account": "1527000000", "account_type": 1}]}, {"token": "token2", "account_list": [{"account": "926@163.com", "account_type": 2}, {"account": "1527000001", "account_type": 1}]}]
op_type	string	No	API operator type: qq, rtx, email, other
op_id	string	No	API operator type: API operator id (qq\rtx\email)

Response Parameter Description

Parameter name	Type	Description
ret_code	int	Return code
err_msg	String	Error message
result	JSONArray	Result of operation for each element, for example: ["ok", "token_not_exists"]

Account API Samples

- Bind token to an account

```
{
  "operator_type": 1,
  "platform": "android",
  "token_accounts":
  [{"token": "token1", "account_list": [{"account": "926@126.com", "account_type": 2}, {"account": "1527000000", "account_type": 1}]},
  {"token": "token2", "account_list": [{"account": "926@163.com", "account_type": 2}, {"account": "1527000001", "account_type": 1}]}]
```

- Bind token to an account in an overriding manner

```
{
  "operator_type": 2,
  "platform": "android",
  "token_accounts":
```

```
{
  "token": "token1",
  "account_list": [
    { "account": "926@126.com", "account_type": 2 },
    { "account": "1527000000", "account_type": 1 }
  ],
  "token": "token2",
  "account_list": [
    { "account": "926@163.com", "account_type": 2 },
    { "account": "1527000001", "account_type": 1 }
  ]
}
```

- Unbind token from an account

```
{
  "operator_type": 3,
  "platform": "android",
  "token_accounts": [
    { "token": "token1", "account_list": [
      { "account": "926@126.com", "account_type": 2 },
      { "account": "1527000000", "account_type": 1 }
    ] },
    { "token": "token2", "account_list": [
      { "account": "926@163.com", "account_type": 2 },
      { "account": "1527000001", "account_type": 1 }
    ] }
  ]
}
```

- Unbind token from all accounts

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

- Unbinding account from all tokens

```
{
  "operator_type": 5,
  "platform": "android",
  "account_list": [
    { "account": "926@126.com", "account_type": 2 },
    { "account": "1527000000", "account_type": 1 }
  ]
}
```

Account-device Binding Query (Batch Operation)

API Description

POST <https://openapi.xg.qq.com/v3/device/account/query>

The API feeds back in real time.

Parameter Description

Parameter name	Type	Required	Description
operator_type	int	Yes	Operation type 1: Batch query the corresponding token according to account 2: Query account according to token
platform	string	Yes	Client platform type 1) android: Android 2) ios: Apple

Parameter name	Type	Required	Description
account_list	jsonArray	No	The account list to be queried, which is valid when operator_type = 1 and required. Each element consists of a pair of account and account_type. Below is a specific example: [{"account":"account1","account_type":1}, { "account":"account2","account_type":2}]
token_list	jsonArray	No	The token list to be queried, which is valid when operator_type = 2 and required.
op_type	string	No	API operator type: qq, rtx, email, other
op_id	string	No	API operator type: API operator id (qq\rtx\email)
##### Response Parameter Description			
Parameter name	Type	Description	
-----	-----	-----	
ret_code	int	Return code	
err_msg	String	Error message	
account_tokens	JsonArray	Array of mappings from account to token, for example: : [{"account":"account1","account_type":1,"token_list":["token1","token2"]} { "account":"account2","account_type":2,"token_list":["token2","token3"]}]	
token_accounts	JsonArray	Array of mappings from token to account, for example: :[{"token":"token1","account_list": [{"account":"926@126.com","account_type":2}, {"account":"1527000000","account_type":1}]}, {"token":"token2","account_list": [{"account":"926@163.com","account_type":2}, {"account":"1527000001","account_type":1}]]	
#### Sample			

- Batch query the token relationships bound to accounts

```
{
  "operator_type":1,
  "platform":"android",
  "account_list":
```

```
[{"account":"account1","account_type":1},{"account":"account2","account_type":2}]
```

- Batch query the accounts bound to tokens

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

Return Field Sample

- Batch query the tokens bound to accounts

```
{  
  "ret_code":0,  
  "err_msg":"ok",  
  "account_tokens":  
  [{"account":"account1","account_type":1,"token_list":["token1","token2"]},  
  {"account":"account2","account_type":2,"token_list":["token2","token3"]}]  
}
```

- Batch query the accounts bound to tokens

```
{  
  "ret_code":0,  
  "err_msg":"ok",  
  "token_accounts":  
  [{"token":"token1","account_list":[{"account":"926@126.com","account_type":2},{account":"1527000000","account_type":1}]},  
  {"token":"token2","account_list":[{"account":"926@163.com","account_type":2},{account":"1527000001","account_type":1}]}  
}
```