

Tencent Push Notification Service

REST API V2

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



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 V2

REST API V2 User Guide

REST API V2

REST API V2 User Guide

Last updated : 2019-06-26 10:03:52

TPNS provides REST-compliant HTTP APIs for developers to remotely call the services provided by it. The APIs are mainly divided into four categories:

- Push API includes various APIs used to push messages
- Tag API includes various APIs used to add, delete, and query tags
- Account API includes various APIs used to query and delete account
- Tools API includes various APIs used to locate problems and query other data

V2 supports both HTTP and HTTPS protocols

Request Method

GET is supported;

POST is supported, but the "Content-type" field in the HTTP Header must be set to "application/x-www-form-urlencoded"

Protocol Description

Request URL:

`http://openapi.xg.qq.com/v2/class_path/method?params`

Field name	Usage	Note
openapi.xg.qq.com	API domain name	None
v2	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

Field name	Usage	Note
params	Parameters passed when the API is called	<ol style="list-style-type: none"> 1. This includes two parts: general basic parameters and API-specific parameters; 2. All parameters must be UTF8-encoded; 3. All the Vs in the K-V pairs in params must be URL-encoded

General Basic Parameters

General basic parameters refer to the (basic) parameters that must be included in the params field in the request URL structure of each API.

The details are as follows:

Parameter name	Type	Required	Description
access_id	uint	Yes	Unique app identifier, which can be found in the console at xg.qq.com
timestamp	uint	Yes	<ol style="list-style-type: none"> 1. UNIX timestamp, used to confirm the validity period of the request 2. If its difference from the server time (Beijing time) is greater than valid_time, the request will be rejected
valid_time	uint	No	<ol style="list-style-type: none"> 1. It determines the validity period of the request together with timestamp 2. It is in seconds 3. The maximum value is 600 4. If it is not passed in, smaller than 0, or greater than 600, it will be set to 600
sign	string	yes	API authentication; for the generation rule, see [Authentication Method](#Authentication Method)

Authentication Method

Calculation formula: Sign=MD5(http_methodURIK1=V1...Kn=Vnsecret_key); (Note: The parameters must be placed in this order as shown)

Parameter name	Description
http_method	Request method; GET or POST
URI	Request URL information, including IP or domain name and the path part of the URI Example: Domain name: openapi.xg.qq.com/v2/push/single_device IP: 10.198.18.239/v2/push/single_device (Note that port and request string are not included)
K1=V1...Kn=Vn	1. Format all request parameters into K=V 2. Sort the formatted parameters in ascending lexicographic order by K and splice them together (Note: 1. Do not include the sign parameter; 2. V should not be URL-encode)
secret_key	API key, which can be found in TPNS > App configuration > App information

For example:

Single-push API call:

POST request `http://openapi.xg.qq.com/v2/push/single_device`

Parameter list:

`access_id=123, timestamp=1386691200, Param1=Value1, Param2=Value2, secret_key=abcde`

Generate according to the formula above:

`sign=MD5(POSTopenapi.xg.qq.com/v2/push/single_deviceaccess_id=123Param1=Value1Param2=Value2timestamp=1386691200abcde)`

Result:

`sign=6b90c7f4a137c7d0b756d48f748c93b2`

General Basic Return Values

General basic return values are the fields in JSON format that will be included in all request responses.

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

```
"result":{"":""}
}
```

The details are as follows:

Parameter name	Type	Required	Description
ret_code	int	Yes	Return code
err_msg	string	No	Result description
result	JSON	No	If the request is correct and there is extra data, the result will be encapsulated in this field

API Limitations

1. Except the [full push](#) API, there is no limitation on the call frequency.
2. The size of the pushed message body cannot exceed 4 KB, and this limitation applies to the message field in Push API.
3. For [tag group push](#), the number of tags can be up to **50**.

Push API

Push API Basic Parameters

Basic parameters of the push API refer to the general parameters of all APIs that push messages. Remember that the API call parameters must include the [general basic parameters](#).

The specific general parameters are as follows:

Parameter name	Type	Required	Default value	Description
message	string	Yes	None	Message body; for more information, see Message Body Format
message_type	uint	Yes	None	For iOS, it must be 0 and does not distinguish between notification bar message and silent message 1 indicates Android notification bar message 2 indicates Android passthrough message

Parameter name	Type	Required	Default value	Description
expire_time	int	No	259200 (72 hours)	Offline message retention duration (in seconds), up to 72 hours 1. If expire_time=0, the default value (72 hours) will be used 2. If expire_time is greater than 0 and less than 800, the system will reset it to 800 seconds 3. If expire_time >= 800 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
send_time	string	No	Current time	1. This specifies the push time in the format of 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 group push
multi_pkg	uint	No	0	Multi-package name push 0 indicates distributing the message according to the package name provided during registration; 1 indicates ignoring the package name and distributing the message by access id (For Android only)
environment	uint	Yes (For iOS only)	1	This field describes the environment of the app 1 indicates the release environment, i.e. the app has been released in App Store 2 indicates the development environment, i.e., the app is still in the debugging environment (For iOS, there are two situations for message push: development environment and release environment)
loop_times	uint	No	None	Looping executions of message delivery Recommended value range: [1, 15]
loop_interval	unit	No	None	Interval for looping message delivery in days; value range: [1, 14]. loop_times and loop_interval together indicate the loop rule for the message delivery task (up to 14 days)

Full Push

This API is used to push the message to all devices. There is a limitation on the frequency of calls by the backend:

- The same message content can be pushed only once per hour;
- This API can be called up to 30 times per hour.

Request URL:

```
http://openapi.xg.qq.com/v2/push/all_device?params
```

Request parameters:

[General basic parameters](#) and [Push API basic parameters](#)

Response result:

[General basic return values](#); the result field will contain the task id delivered to the app:

```
{
  "push_id":10000
}
```

If it is a looping task, IDs of all tasks will be returned

Below is a specific example:

```
{
  "push_id":10000, // Parent task id
  "sub_push_ids":[234,235,236] // Sub-task id
}
```

Group Push

Group push refers to push based on tag, account, or device token groups.

Tag Group Push

It can push message to devices with specific tags. For example: gender, identity, etc.

Request URL:

```
http://openapi.xg.qq.com/v2/push/tags_device?params
```

Request parameters:

In addition to [General basic parameters](#) and [Push API basic parameters](#), there are the following specific parameters:

Parameter name	Type	Required	Default value	Description
tags_list	JSON	Yes	None	["tag1","tag2","tag3"] The number of tags cannot exceed 50; otherwise, the message will fail to be delivered If there are more than 50 tags, it is recommended to use the full push API to push the message
tags_op	string	Yes	None	Operator; value: AND or OR

Response result:

[General basic return values](#); the result field will contain the task id delivered to the app:

```
{
  "push_id":10000
}
```

If it is a looping task, IDs of all tasks will be returned

Below is a specific example:

```
{
  "push_id":10000, // Parent task id
  "sub_push_ids":[234,235,236] // Sub-task id
}
```

Account Group Push

Account group push refers to push to a group of accounts bound through the binding API of the client SDK. Both the iOS and Android SDKs provide corresponding APIs.

Request URL:

```
http://openapi.xg.qq.com/v2/push/account_list?params
```

Request parameters:

In addition to [General basic parameters](#) and [Push API basic parameters](#), there are the following specific parameters:

Parameter name	Type	Required	Default value	Description
account_list	array	Yes	None	JSON array format Each element is an account For a single push, there can be no more than 100 accounts Example: ["account1","account2","account3"]

Response result:

[General basic return values](#); the JSON of the result field sends the return code for each account.

Super-large Account Group Push

If the number of push target accounts is high (for example, >100), this method is recommended, which includes the following two steps:

Step 1. Create a push message:

Request URL:

```
http://openapi.xg.qq.com/v2/push/create_multipush?params
```

Request parameters:

[General basic parameters](#) and [Push API basic parameters](#)

Response result:

[General basic return values](#), where the JSON of the result field contains the message identifier, for example:

```
{
  "push_id":100000
}
```

Step 2. Use the super-large group push API to push the message.

Request URL:

```
http://openapi.xg.qq.com/v2/push/account_list_multiple?params
```

Request parameters:

In addition to the [general basic parameters](#), the following parameters are included:

Parameter name	Type	Required	Default value	Description
account_list	array	Yes	None	JSON array format Each element is an account For a single push, there can be no more than 1,000 accounts
push_id	uint	Yes	None	The message identifier returned by the API that created the push message

Response result:[General basic return values](#)**Device Group Push**

Device group push refers to push to a group of device tokens.

Use this API in 2 steps:

Step 1. Create a message:

Request URL:

```
http://openapi.xg.qq.com/v2/push/create_multipush?params
```

Request parameters:

[General basic parameters](#) and [Push API basic parameters](#)

Response result:

[General basic return values](#), where the JSON of the result field contains the message identifier, for example:

```
{  
  "push_id":100000  
}
```

Step 2. Call the push API

Request URL:

```
http://openapi.xg.qq.com/v2/push/device_list_multiple?params
```

Request parameters:

In addition to the [general basic parameters](#), the following specific parameters are included:

Parameter name	Type	Required	Default value	Description
device_list	array	Yes	None	JSON array format Each element is a token For a single push, there can be no more than 1,000 tokens Example: ["token1","token2","token3"]
push_id	uint	Yes	None	The message identifier returned by the API that created the push message

Response result:

[General basic return values](#)

Single Push

Single push refers to push to the specified account or device token.

Single-account Push

Single-account push refers to push to a specified account bound through the binding API of the client SDK. Both the iOS and Android SDKs provide corresponding APIs.

Request URL:

`http://openapi.xg.qq.com/v2/push/single_account?params`

Request parameters:

In addition to [General basic parameters](#) and [Push API basic parameters](#), there are the following specific parameters:

Parameter name	Type	Required	Default value	Description
account	string	Yes	None	None

Response result:

[General basic return values](#)

Single-device Push

Single-device push refers to push to a device with the specified device token. The client SDK has a corresponding API for obtaining the device token.

Request URL:

http://openapi.xg.qq.com/v2/push/single_device?params

Request parameters:

In addition to [General basic parameters](#) and [Push API basic parameters](#), there are the following specific parameters:

Parameter name	Type	Required	Default value	Description
device_token	string	Yes	None	Device token

Response result:

[General basic return values](#)

Message Body Format

The message body is 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, corresponding to the message parameter in the Push API.

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

Message type	Supported platform	Feature description
Notification bar message	Android, iOS	Message is displayed in the notification bar
Passthrough message	Android	Message is not displayed in the notification bar
Silent message	iOS	Message is not displayed in the notification bar

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
xg_media_resources	array	None	No	Rich media element addresses; up to 5 ones recommended (only for SDK v4.2.0 and higher)
builder_id	int	None	No	Local notification style identifier
n_id	int	0	No	Unique identifier of the notification message object 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
ring	int	1	No	Whether there a ringtone 0: No 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 0: No 1: Yes
lights	int	1	No	Whether to use breathing light 0: No 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 0: In-app icon 1: Uploaded icon
icon_res	string	None	No	In-app icon file name or URL address of the downloaded icon

Field name	Type	Default value	Required	Parameter description
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
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

Below is an example of a complete message:

```
{
  "title ":"xxx",
  "content ":"xxxxxxxxx",
  "xg_media_resources" : "xxx",// Rich media element address, for example: https://www.xx.com/img/bd_logo1.png?qua=high
  "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",
  "custom_content":{
    "key1":"value1",
    "key2":"value2"
  },
  "action":{
    "action_type":1,// Action type; 1. Open activity or app; 2. Open browser; 3. Open Intent
    "activity":"MyActivityClassName"
  }
}
```

```
"aty_attr":{ // activity attribute, only for action_type=1
"if":0, // Intent's Flag attribute
"pf":0 // PendingIntent's Flag attribute
},
"browser":{
"url":"http://xg.qq.com", // Only http and https are supported
"confirm":1 // Whether user confirmation is required
},
"intent":"xgscheme://com.xg.push/notify_detail"// The client Android SDK version must be 3.2.3 or higher. Configure the data tag in the client's intent and set the scheme attribute
},
"accept_time":[// Between 1 pm and 2 pm or between 0 am and 9 am, the message can be displayed; in other time periods, the message will not be displayed
{
"start":{
"hour":"13",
"min":"00"
},
"end":{
"hour":"14",
"min":"00"
}
},
{
"start":{
"hour":"00",
"min":"00"
},
"end":{
"hour":"09",
"min":"00"
}
}
]
}
```

General Message on iOS

The specific fields for the iOS platform are as follows:

Field name	Type	Default value	Required	Parameter description
------------	------	---------------	----------	-----------------------

Field name	Type	Default value	Required	Parameter description
aps	JSON	None	Yes	<p>APNs-specific message body field</p> <p>Important key-value pairs include:</p> <p>alert: It contains the title and message content (required)</p> <p>badge_type: The badge number displayed by the app (optional), which can be set to the following three values:</p> <ol style="list-style-type: none"> 1) -1: Badge number does not change 2) -2: Badge number is automatically increased by 1 3) ≥ 0: Set "custom" badge number <p>category: The action identifier displayed when the message is pulled down (optional)</p> <p>For details, see Payload</p>
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:

```
{
  "aps":{
    "alert":{
      "title":"this is a title",
      "body":"this is content"
    },
    "badge_type":1,
    "category":"CategoryID"
  },
  "custom":{
    "key":"value"
  }
}
```

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",
  "custom_content":{
    "key1":"value1",
    "key2":"value2"
  },
  "accept_time":[// Between 1 pm and 2 pm or between 0 am and 9 am, the message can be displayed;
  in other time periods, the message will not be displayed
  {
    "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 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:

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

Querying Message Status (Unavailable in V2 and Available in V3)

Currently, this API only supports querying the sending status of messages for full push and tag group push.

Request URL:

http://openapi.xg.qq.com/v2/push/get_msg_status?params

In addition to the [general basic parameters](#), the following specific parameters are included:

Parameter name	Type	Required	Default value	Description
push_ids	array	Yes	None	Unique message identifier, which can be found in the console JSON format Example: [{"push_id": "10000"}]
start	int	Yes	None	Start record id, used in paged query
length	int	Yes	None	Number of query records
type	int	Yes	None	Push message type: 1 - notification bar message 2 - passthrough message (in-app message)
push_type	int	No	None	Mode: 1 - web, 2- REST API, 3 - all
task_type	string	No	None	Push value range: 0 - all ranges (device, account, tag) 1 - device 2 - account 3 - tag
platform	int	No	None	Platform: 1 - Android, 2 - iOS
start_date	string	No	None	Format: yyyy-mm-dd
end_date	string	No	None	Format: yyyy-mm-dd
status	int	No	None	0 - (all states) 1 - to be pushed 2 - pushing 3 - pushed 4 - push failed 5 - invalid task 6 - other status
message	string	No	None	Fuzzy search based on message content
operation	int	Yes	None	Recommended value: 1

Response result:

[General basic return values](#), where the JSON of the result field is in the following form:

```
{
  "Total": "1",
  "list": {
    "0": {
      "Content": "test", // Push message body
      "OfflineSave": "86400", // Android offline retention duration
      "ScheduleSendTime": "2017-04-12 17:50:00",
      "SendTime": "2017-04-12 17:50:01", // Choose from tobe_sent_tome and creat_time\start_time according to the algorithm
      "TagsList": "", // List of tags when tagging
      "Title": "this is title", // Push title
      "Type": "3", // Task type: 3 - full push, 4 - tag push
      "cleanup": "0", // Clear
      "click": "0", // Tap
      "create_time": "2017-04-12 17:49:01", // Task creation time
      "push_active": "0", // Active Android devices in the last 30 days
      "push_id": "2511161036",
      "push_online": "0", // Actual deliveries
      "start_time": "2017-04-12 17:50:01", // Push task start time
      "status": "2", // 0 - to be pushed, 1 - pushing, 2 - pushed, 3 - push failed, 4 - push task expired, 6 - pushing, 7 - stopped, 9 - push failed (too frequent), 10 - push failed (repeated content), 11 - stopped, 12 - stopped
      "verify": "123", // Display
      "verify_svc": "0" // Arrivals at devices
      "cal_type": "0"
    }
  }
}
```

Canceling Push

Currently, V2 supports the cancellation of scheduled [full push](#) or [tag group push](#) based on message ID.

Request URL:

```
http://openapi.xg.qq.com/v2/push/cancel_timing_task?params
```

Request parameters:

In addition to the [general basic parameters](#), the following specific parameters are included:

Parameter name	Type	Required	Default value	Description
push_id	string	Yes	None	Unique identifier of the created full push message or tag group push message, which can be found in the TPNS console

Response result:

[General basic return values](#), where the JSON of the result field is in the following form:

```
{
  "status": 0 // 0 - success; others - failure
}
```

Tag API

Tag API is mainly used to query, set, and delete tags.

The specific API supported by V2 are as follows:

1. Batch add tags
2. Batch delete tags
3. Query all tags
4. Query the tags of a single device (based on device token)
5. Query the number of device tokens with a single tag

Batch add tags

Batch adding tags can set tags for multiple devices (device tokens), but only up to 10,000 tags can be set under one app. If this limit is exceeded, this API will fail.

Request URL:

```
http://openapi.xg.qq.com/v2/tags/batch_set
```

Request parameters:

In addition to the [general basic parameters](#), the following specific parameters are included:

Parameter name	Type	Required	Default value	Description
----------------	------	----------	---------------	-------------

Parameter name	Type	Required	Default value	Description
tag_token_list	array	Yes	None	<p>JSON string</p> <p>Each array element is an array of tag-token pair</p> <p>Up to 20 pairs are allowed per call</p> <p>In each tag-token pair, the tag is before the token</p> <p>Example:</p> <pre>[["tag1","token1"],["tag2","token2"]]</pre> <p>Note:</p> <ol style="list-style-type: none"> 1. A tag string cannot exceed 50 bytes in length or contain spaces 2. The token must be a device identifier that meets the message reception requirements

Response result:

[General basic return values](#)

Batch delete tags**Request URL:**

```
http://openapi.xg.qq.com/v2/tags/batch_del
```

Request parameters:

In addition to the [general basic parameters](#), the following specific parameters are included:

Parameter name	Type	Required	Default value	Description
tag_token_list	array	Yes	None	<p>JSON string</p> <p>Each array element is an array of tag-token pair</p> <p>Up to 20 pairs are allowed per call</p> <p>In each tag-token pair, the tag is before the token</p> <p>Example:</p> <pre>[["tag1","token1"],["tag2","token2"]]</pre> <p>Note:</p> <ol style="list-style-type: none"> 1. A tag string cannot exceed 50 bytes in length or contain spaces 2. The token must be a device identifier that meets the message reception requirements

Response result:

General basic return values

Query all tags

This API is used to query the total number of tags and their names that are set under the currently specified app.

Request URL:

```
http://openapi.xg.qq.com/v2/tags/query_app_tags?params
```

Request parameters:

In addition to the [general basic parameters](#), the following specific parameters can be included:

Parameter name	Type	Required	Default value	Description
start	uint	No	0	Start index value
limit	uint	No	100	Limit the quantity in a single query, which is recommended to be below 200

Response result:

[General basic return values](#), where the result field is in the following JSON format:

```
{
  "total": 2, // Specify the total number of tags for the app
  "tags":["tag1","tag2"] // The array of tags queried according to the limit parameter
}
```

Query the tags of a single specified device

This API is used to query all the tags set for the specified device based on the device token. Please ensure the validity of the device token.

Request URL:

```
http://openapi.xg.qq.com/v2/tags/query_token_tags?params
```

Request parameters:

In addition to the [general basic parameters](#), the following specific parameters are included:

Parameter name	Type	Required	Default value	Description
device_token	string	Yes	None	Token of the device receiving message

Response result:

In the [General basic return values](#), the result field is in the following JSON format:

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

Query the number of tokens for a single specified tag**Request URL:**

```
http://openapi.xg.qq.com/v2/tags/query_tag_token_num?params
```

Request parameters:

In addition to the [general basic parameters](#), the following specific parameters are included:

Parameter name	Type	Required	Default value	Description
tag	string	Yes	None	Tag string to be queried

Response result:

[General basic return values](#), where the result field is in the following JSON format:

```
{
  "device_num":100000
}
```

Account API (Unavailable in V2, Only for V3)

Account API is mainly used to query and delete the account associated with the device token.

The specific API supported by V2 are as follows:

1. Query the device token associated with a single account
2. Delete the device token associated with a single account
3. Delete all device tokens associated with an account

Query the device token associated with a single account (unavailable at present)**Request URL:**

http://openapi.xg.qq.com/v2/application/get_app_account_tokens?params

Request parameters:

In addition to the [general basic parameters](#), the following specific parameters are included:

Parameter name	Type	Required	Default value	Description
account	string	Yes	None	Account ID

Response result:

[General basic return values](#), where the result field is in the following JSON format:

```
{
  "tokens":["token1","token2"]
}
```

Delete a single device token associated with a single account

Request URL:

http://openapi.xg.qq.com/v2/application/del_app_account_tokens?params

Request parameters:

In addition to the [general basic parameters](#), the following specific parameters are included:

Parameter name	Type	Required	Default value	Description
account	string	Yes	None	Account associated with the device token
device_token	string	Yes	None	Token of the device receiving message

Response result:

[General basic return values](#), where the result field is in the following JSON format:

```
{
  "tokens":["token1"]
}
```

Note: The value corresponding to the token field indicates the device token currently associated with the current account.

Delete all device tokens associated with a single account

Request URL:

```
http://openapi.xg.qq.com/v2/application/del_app_account_all_tokens?params
```

In addition to the [general basic parameters](#), the following specific parameters are included:

Parameter name	Type	Required	Default value	Description
account	string	Yes	None	Account ID

Response result:

[General basic return values](#)

Tool API (Unavailable in V2 and Available in V3)

Query the number of tokens covered by an app

This API is used to query the number of all device tokens registered with the specified app.

Request URL:

```
http://openapi.xg.qq.com/v2/application/get_app_device_num?params
```

Request parameters:

[General basic parameters](#)

Response result:

[General basic return values](#), where the JSON of the result field is in the following form:

```
{
  "device_num": 34567
}
```

Query the registration status of the specified device token

This API is used to query the registration status of the specified device token with the TPNS server. The first prerequisite for a device to receive messages from TPNS is that its device token has been registered with the TPNS backend; otherwise, TPNS cannot deliver messages to it.

Note: This API can only query the token for Android.

Request URL:

http://openapi.xg.qq.com/v2/application/get_app_token_info?params

Request parameters:

In addition to the [general basic parameters](#), the following specific parameters are included:

Parameter name	Type	Required	Default value	Description
device_token	string	Yes	None	Token of the device receiving message

Response result:

[General basic return values](#), where the JSON of the result field is in the following form:

```
{
  "isReg":1,// (1 - registered, 0 - not registered)
  "connTimestamp":1426493097, // (timestamp of the latest active status)
  "msgsNum":2 // (number of offline messages for this app)
}
```

Return Code List

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](#).

Value	Meaning	Solution
0	Call succeeded	
-1	Parameter error	Please check the parameter configuration
-2	Request timestamp is not valid	Please check timestamp and <code>valid_time</code> parameters
-3 -5 5	Error in TPNS handling	Please retry later
2	Error binding the tag	Token or tag is empty when binding the tag

Value	Meaning	Solution
6	Device token registration failed	Please check whether the device registration is successful
7	General error. Account limit exceeded	Delete other unused accounts (call the account unbinding API)
48	Push account is not bound to a token	Please check whether the account and token have a binding relationship
73	Message body limit exceeded	Currently, the limit is 4 KB
75	Message body is not in JSON format	Please check the message body (message field) content
78	Looping task parameter error	Please check loop_time
83	Push content invalid	Please check whether the text contains harmful information
91	Too many tags associated with the device token	Clear unused tags
92	Too many tags associated with the app	Clear unused tags (the limit is 10,000)
100	APNs certificate error	The format of the push certificate used by TPNS should be PEM. In addition, pay attention to the difference between production certificate and development certificate
-101	Parameter error	Please check the general basic parameters
-102	Request timestamp is not valid	Please check timestamp and valid_time parameters
-103	sign is invalid	Please check the signature generation process. The sign generating method must be the same as that used when requesting

Value	Meaning	Solution
-104	Internal Error	Please retry later
-106	Certificate expired	Certificate expired. Please re-upload a certificate