

# Chat Push Feature Product Documentation





#### Copyright Notice

©2013-2024 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** 

#### STencent 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.

Chat



### Contents

**Push Feature Push Function Introduction** Plugin Integration Push (Recommend) Overview Manufacturer Configuration Android iOS uniapp Flutter **React-Native Quick Integration** Android uniapp Flutter **React-Native** Statistics **Troubleshooting Tool Client APIs** Android iOS Flutter All Staff/Tag Push Pushing to All/Tagged Users **Obtaining Application Attribute Names** Setting Application Attribute Names **Obtaining User Attributes** Setting User Attributes **Deleting User Attributes Obtaining User Tags** Adding User Tags **Deleting User Tags Deleting All User Tags Recalling Push Advanced Features Custom Definition Badge** 

©2013-2022 Tencent Cloud. All rights reserved.



Custom Definition Ringtone Customized Icon Custom Definition Click Redirect Push Message Categorization Release Notes FAQS

# Push Feature Push Function Introduction

Last updated : 2024-06-13 10:21:45

# Overview

Instant Messaging provides you with two integration methods: TIMPush (recommended) and Self-integrated Push. Both integration methods support Xiaomi, Huawei, Honor, OPPO, vivo, Meizu, APNs, OnePlus, realme, iQOO, FCM and other manufacturer channels, but they are different in access integration, push type, data statistics, link tracking, etc., see the table below for details:

Comparison Items		TIMPush	Self-integrated Push
Price		299 USD/month	Free
	Android & iOS	1	$\checkmark$
Supported	uni-app	1	×
Platforms	Flutter	1	×
	React Native	1	×
	Manufacturer SDK Integration	One-click integration	Need to integrate 7 Android vendors and 1 iOS vendor one by one, a total of 8 SDKs
Access Integration	SDK Local Deployment	One-click configuration	Need to configure 8 manufacturers one by one
Access megration	Push registration, token reporting	$\checkmark$	Need to develop reporting logic by yourself
	Access test tool	✓	1
	Access cycle	1 hour	1 week
Push Type	Ordinary Message Push	✓	✓
	All/Tag Users Push:On-site push	✓	-



	All/Tag Users Push:Off-site push	✓	-
	Ordinary Message Push Record Query	$\checkmark$	-
	All/Tag Users Push Record Query	$\checkmark$	-
Statistics	Ordinary Message Push Data Statistics (Actual delivery rate, reach rate, click rate, etc.)	✓	-
	All/Tag Users Push Data Statistics (Actual delivery rate, reach rate, click rate, etc.)	√	-
	Push Channel Query	$\checkmark$	-
	Push Device Status Query	$\checkmark$	-
Link Tracking	Push Full Link Status Query (including IM server > manufacturer server > terminal device > the entire link clicked by the user)	✓	-
Offline Push Reach		Active users within 30 days	Active users within 7 days

## Contact us

If you encounter problems during use, you can solve them by checking the FAQ, or you can enter the communication group for consultation: Telegram communication group: click to join. WhatsApp communication group: click to join.

# Plugin Integration Push (Recommend) Overview

Last updated : 2024-06-13 10:21:45

TIMPush provides you with stable, timely and diversified push services. Compared with self-integrated push, TIMPush only requires simple configuration and can integrate and access push services from multiple manufacturers with one click. TIMPush supports ordinary message push and all/tag users push, and provides complete push life cycle query, data statistics, and problem troubleshooting services.

If you have chatting, audio and video calls, signaling and other scenarios and need to be able to reach them in time even in offline scenarios, you can pay attention to the **ordinary message push function**.

If you have marketing advertisements, notifications, news information, etc. that need to be pushed to all users or designated groups, you can pay attention to the **all/tag users push function**.

Offline push manufacturers support Xiaomi, Huawei, Honor, OPPO, vivo, Meizu, APNs, including sub-brands of various manufacturers such as OnePlus, realme, iQOO, etc., and support Google FCM overseas.

The push management console provides you with **full-link troubleshooting tools**, **push records**, and **statistical data on various indicators**, making it easy for you to view various indicators such as push reach rate, click-through rate, and conversion rate.

# **Features**

### Quick integration in 3 minutes

It is no longer necessary to configure push information for each manufacturer separately. You only need to download and import the json configuration file in the IM console, and you can complete the push information configuration for all mobile phone manufacturers with one click.

Supports the integration of one or more push channel packages from corresponding manufacturers on demand to easily cope with compliance requirements.

There is no need to handle push registration, token reporting, front-end and back-end status reporting, etc. by yourself. The push plug-in is self-closed.

There is no need to add management and processing reporting logic by yourself. The push plug-in reports and summarizes by itself. It also supports link troubleshooting and indicator statistics.

The plug-in encapsulates interface jumps, icon customization and other methods, and can be used directly.



# Ordinary message push

In ordinary IM message sending and receiving scenarios, messages can reach the device in time even when the application is offline and support customizable jump interfaces.



# All/tag users push

The all/tag users push function is designed to help you push marketing, advertising, notifications, etc. to all users or tag users, ensuring that the right messages are sent to the right users in the right way at the right time.



All/tag users push supports two push methods: on-site and off-site push:

#### **On-site push**

Supports message roaming. Push messages will also enter the IM message system, which will trigger updates of corresponding sessions, messages, and unread modules. Users can receive push messages when they are online. When users are not online, they can automatically receive push messages after logging in next time. news.

#### Off-site push

Push messages arrive on the device as system notifications and are not saved in the IM system.



# Custom jump interface

After the user receives the push, click on the notification bar to see the customized jump interface.

	Custom jump interface	
17:48 ┩ ▲ <sub>援索</sub> 推荐 乐馆 听书 会员 故事 演出	9:41 <b>I ≎</b> Candy	9:41
Music 1 minute Come and listen to good music	ago Candy Contact us at a later meeting Frances	DOMINIK I feel like Tencent Cloud is
* 6%5 f 美音适合放点 BOM 免費所 一路向北 - 陈满满 毎日30首 材	Have read Okay	Explore and learn more about Tencent Cloud's all cr and promotions
听「绅士」的也在听 ▶ 兔费 ···· 成全 好病点行热调 林宥嘉-成全	Candu <sup>1 m</sup>	Lighthouse Offers Enjoy discounts up to 60% for new users and r Inute ago your business to cloud in a cost-effective and
夜曲 版版物 No.53 周杰伦-十一月的萧邦		A discount up to 60% on your first purchase     Buy more, save more     Choose from a wind unright of instances to more
<ul> <li>試預 進行通過 Jason小床-悬强</li> <li>斜洗节能执誓他按     或所中,完整挂数需并量 VP     </li> </ul>		different business needs.
	١	+ Free Tier
Jump to home page	Jump to the specified interf	ace

# Push custom styles

Supports custom styles of small icons, right icons, long text, large pictures, badge and ringtones.



# Full-link troubleshooting tools

Provide self-service troubleshooting tools to view the entire push link details, analyze push failure and click failure reasons, and improve conversions.

	Pu	sh troubleshooting	
	Push status		
		Task ID	
		User ID: (	
		Submit instant mes	saging IM server
Device Information		2024-05-07	14:56:06
Basic in. Device information			
Push time 2024-05-07 14:56:06		♥ ⊘	▼
Device model RMX3370		Submit online channel 2024-05-07 14:56:06	Submit offline channel 2024-05-07 14:56:06
Operating System Android		1	1
SDK version 7.9.5666		Ø	Ø
Push plug-in version 7.9.5666		Push message to device 2024-05-07 14:56:06	Submit vendor channel 2024-05-07 14:56:06
Manufacturer token rer CN euranouse, recourt	a	1	Ļ
Certificate ID		0	Ø
0000	sk t	User click status statistics for online push will be supported soon, so stay tuned	Push message to device 2024-05-07 14:56:08
Equipment status			↓ ↓
	Push f	ull link	③ User click
Notification bar status () Enable			UnKnown
token binding status bound	<ol> <li>The application in</li> </ol>	s switched to the background and the process is alive. Online and offline push will be s	ent at the same time

# Statistical data

Record and query all push data of users, organize and analyze various types of indicator data pushed by users every day, and generate recent delivery > reach > click funnel conversion charts, which can be viewed according to manufacturer channel classification.

Push indicator analysis



Push record

Page 16 of 446

Push Records 290 atbot      Current data cer	ter: Singapore ① Telegram group WhatsApp group	
All members/tag push Normal push		
2024-05-05 00:00:00 ~ 2024-05-05 23:58:59 T Push content Please enter the push cont Cuery Reset		
Task ID Push time Push rec	uest content	
10000548 2023-10-24 16:34:38 14e (5)	setti; "Tagicuri: nur, "AmsNuri: (), - Tagicuri: (), - Magdiooy; () "M on": 小TuniniasaD(**Hat, H), (**), "Deut": [] 包括書面和因为著戶 "这是通知的内容", "AndroidInto": (), "Aprointo": ("BadgeMode": 1.)),	terer: "Charge Science Samoeries Connact provintimer Vitomics" 4911, "Tex Data": "Content", "Charge Science: 2) J. Made Elimari: 3114400, "MagRan 53752948, 200005640a873, d604675_b06146" )
1000548	test"), "TagsOrt" null, "WithAnd"; (), "AttrSCh"; (), "MsgBody"; ( ( 'M path") 'UnuinestR"-'UnuinestR"-'Entri Ind',", "Desc", " ( Entri Ref. at 10:758/P	link(*'\'https://obud.tencent.com/act/pro/mnew/?hom=214811'\'ttex Data'' 'content" 'lindes'' 2111 'Mont Mallion'' 31104000 'MonBan
4e 10 2023-10-24 16:39:46	"这是进作的内容"、"AndroidInto"、("Smallcon", "private.png", "VivoCi .537509d8.2000005640aa873_507c6318,bc614e" )	", "HuiWeimage": "xxx.png", "HonorImportance": "NORAML", "Hon
2020-548 2023-10-26 11:49:08	test" ), "TagsOrt" nuli, "AttraAnd": (), "AttraOrt" () ), "MagBody": ( ( 'M ont": 1/\businessEV:'\ftest, link')", "Desc": " (国际通信 网络刀子名户	link':\'https://oloud.tencent.com/act/pro/innew?ftom=216811\'\'tex Data': 'contern', 'index: 2 ] ], 'MagLife'Imo': 3 1104000, 'MagBan'
198 0	"这是意作的内容"、"AndroidInto":()、"Aprolinto":( "BadgeMode":1 ) )。	537529d8_2000005e80aa873_431bd3bb_bc614e* )
100005e8 2023-10-26 11:49:23 4e 10	test [], "TagiCh": nuil, "Attrikent": (), "AttrixCh": (), "MagBody": (] 'M ont": ():VentsinessD(*):Ventsi, In(*)", "Desro": "(回时通信和地位): 76/ "这是遗传的内容", "Androidinto: (), "Apprinto": ("BadgaModot": 1),	link'', 'Trippu/blocki.stenoent.com/act/pac/imex/Thom/21481''\nex Data': 'content', 'linkev:' 21), 'MogLife'Time': 31104000, 'MogRan 63752048_20000056e0aa873_426614_00514*)
0000588 2000 10 00 00 00 00	test"), "TagsOr"; null, "AttraAnd"; (), "AttraCr" (), "MagBody"; ( ( "M	InV/1/Tetps://obud tencent.com/act/pro/imnew/Picors-214811/1/Tex
14e 10 2023-10-20 1337-33	dity::https://districtional.org/mix/j, Units:://Listightpin/distriction/ "認思思考例的外容", "AndroidHito": (), "Apresinto": ("BadgeMode": 1 )),	Lata: content, intext:2)], vegue inter:3109000, vegnan 33752968,200005460as73_396602bb_bc614e*)
2023-10-26 14:10:18 4e 10	test"), "TagsOr": null, "AttraAnd": (), "AttraOr": ()), "MagBody": ( ( "M ont": 1, "businessE/", "Mandia: ( ), "AttraOr": ( )), "MagBody": ( ( "M ): (注意思考试的思想: " Andramitica"), " Andramitica": ( ) 和andramitica": 1 ))	link*'\*https://oloud.tencent.com/sct/pro/mnew?ftom#21481\'\*tex Data': "content", "index: 2) ], "MgJd#1imo" 3104000, "MgBan 15275948, 2000066048873 89-004614 1:editad".
	test"], "TagsOrt" nul, "AttrsAnd": (), "AttrsCr"()), "MsgBody"; ( ( "M	link(*\'\'https://oloud.tencent.com/act/pro/mew?horm=21481\'\'\'tex
1000009 2023-10-26 14:13:25 3e (1)	ont't',t'/busines80'\'\'text,ink\'7, 'DesC': ' [即时通信 和 助方客户 '这是遗传的内容', 'Androidinto': (), 'Apnsinto': ('BadgeMode': 1 )),	Data*: "contern", "index: 21]), "MagLifeTime": 31104000, "MagRan 537529d8_2000005e80aa873_2fdtbadb_bo514e")
X0000566 2023-10-26 14:15:51	test" ], "TagaOr"; null, "AttraAnd"; (), "AttraCr"; () ], "MagBody"; (   'M on\":1,"businessD\'.\"test_In\(?)", "Desc"; * (回时通信 M 因)力答户	InAl "\/https://cloud.tencert.com/act/pro/imow/Ptom/21481/\/\*ex Data": foontert", "Index": 2 } }, "MagLifeTime": 31104000, "MagRan
	"这是进作的内容", "Androidhto"; (), "Aprendo"; ("BadgeMode"; 1)),	55752948_2000055480aa873_86482a50_bc614e* }
00005e8 2023-10-26 14:16:52 te f5	ment joingtoo (may contention to content	Data": Content", Trades": 21 ) J. "Model Inno": 31104000; "MsgRan 53782948,200005660a873,4achbdt,bc614e" )
X000546 2023-10-26 14:16:16	test" ], "TagsOr"; null, "AttrsAnd"; (), "AttrsOr"; (), "MsgBody"; [ ( 'M on\"-1,"businessOr\"\"test_ink"; ", "Dear"; * (BN)경제, M 8)가제주	Init'/'\'https://cioud.innoent.com/act/pro/innew?fform-21481/\\'tex Data'' content", 'Inidex': 213, 'MagLieTima': 31104000, 'MagRan
4e 10	"这是遗传的内容","AndroidInfo":(),"Apresimio":( "BadgetMode":( ) )。	537529d8_000005e80aa873_s28a2b75_bc814e* }
共 28 条		10 ≠ / pape → ( <

# Activate

# Step 1: Activate the TIMPush

Go to IM Console > Push, click Buy Now or Free Trial . (Each application can be tried once for free, valid for 7 days)

Chat	Access settings Current data center: Singapore (i) Telegram group WhatsApp group
로는 Application	Push Overview
음 Users	Function Preview       Function Preview <th< th=""></th<>
몲 Groups III Configuration ·	Purchase Now free trial
<ul> <li>Webhook</li> <li>Statistics</li> <li>Plugin</li> </ul>	Manufacturer configuration IM supports online and offline push notifications. Online push is delivered through the instant messaging IM channel, which is fast and reliable; for offline push, it is recommended that you use the system-level long connection, and the resource consumption is greatly reduced.
🔁 Push 🔹 ^	Android iOS
Access settings	Mi Huawei Meizu vivo OPPO Honor FCM NIO
Access settings <ul> <li>Access test</li> <li>Push Record</li> </ul>	Mi     Huawei     Meizu     vivo     OPPO     Honor     FCM     NIO       Add Certificate
Access settings <ul> <li>Access test</li> <li>Push Record</li> <li>Push Data</li> <li>Push Troubleshooting</li> </ul>	Mi Huawei Meizu vivo OPPO Honor FCM NIO Add Certificate

#### Notice :

After the trial or purchase of the TIMPush expires, **push services (including offline push of ordinary messages, all/tag users push, etc.) will be automatically stopped.** To avoid affecting the normal use of your business, please purchase/renew in advance.

### Step 2: Integrate and use TIMPush

Vendor configuration Android & iOS & Flutter & uniappo Quickly access Android & iOS & Flutter & uniappo

# Step 3: Send all/tag users push

For details, please see push to all/tags.

# Step 4: Push diverse implementations

For details, please see Advanced Features.

### Step 5: Push Management and Analysis



Statistics Troubleshooting tools

# **Contact us**

If you encounter problems during use, you can solve them by checking the FAQ, or you can enter the communication group for consultation: Telegram communication group: click to join. WhatsApp communication group: click to join.

# Manufacturer Configuration Android

Last updated : 2024-06-13 10:21:45

# Operation step

#### Step 1. Register your app with vendor push platforms

To utilize the offline push feature, you need to register your app on each vendor's push platform to obtain parameters such as AppID and AppKey. Currently, the mobile manufacturers supported in China include: Mi, Huawei, Honor, OPPO, Vivo, Meizu, and internationally Google FCM is supported.

#### Step 2. Create resources in the IM console

Log in to Tencent Cloud Chat Console, then in the **Push Management** > **Access Settings** feature section, add each vendor's push certificate, and configure the AppID, AppKey, AppSecret, and other parameters obtained in Step 1 to the added push certificate.

Explanation of the Subsequent Actions option:

Open Application: Clicking the notification bar launches the app, by default starting the app's Launcher interface; Open Web Page: Clicking the notification bar will redirect to the configured web link;

Open the specified interface within the app: clicking the notification bar will redirect the interface based on the configured self Definition, see Custom Redirect on Click.

Mi

Huawei

OPPO

vivo

Meizu

HONOR

Google FCM

Vendor Push Platform	Configuring in the IM console





Vendor Push Platform	Configuring in the IM cons



OPPO 1 ##25##Yi I in kit   DERJVK DERJVK × DERGE ×   O MB##24 Fig > DERDE ×   O M25 #K8 F	OPPOEM COOPOSALE 9	放平台 用户中心	Add OPPO ce
DERTINE       DERTINE       Package Name         OR MERE       DERTINE       AppKey         OR MERE       DERTINE       AppKey         OR MERE       LREME       ApplO         OR MERE       ApplO       AppSecret         OR MERE       ApplO       MasterSecret         OR MERE       AppSecret       75         DialeTe       MasterSecret       ChannellD         MasterSecret       State       Otherse	<b>の アクロ</b>   推送运营平台	III O 52 tuikit V	
Image: Second secon	应用列表	☆	Package Name *
• 数据统计 • tuikit   • 推发审核 • L/BBK   • 推发审核 • L/BBK   • ChannellD   • 加加    • 加加   • 加加 <td>☆ 创建推送 🛛 🗸 🗸 🗸</td> <td>首页 &gt; 应用配置</td> <td>AppKey <b>*</b></td>	☆ 创建推送 🛛 🗸 🗸 🗸	首页 > 应用配置	AppKey <b>*</b>
• ① 推送审核           • 广信服际           • 广信服际           • 八百百音理           • 八百百言章理           • 八百百言章理         • 八百百言章理           • 八百百言章理           • 八百百言章理           • 八百百言章理           • 八百百言章理           • 八百百言章理           • 八百百言章理           • 八百百言章理           • 八百百言章理           • 八百百言章理           • 八百百言章理           • 八百百言章理         • 八百百言章理         • 八百百言章理         • 八百言章         • 八百言         • 八百言章         • 八百言         • 八	√ 数据统计 ~	tuikit	AppID *
RETER     Appld       AppKey     cfrommer       AppSecret     75       Stable     AppSecret       AppSecret     75	▲ 推送审核 ✓		AppSecret *
ball     AppKey of     ball     ball     cf     channellD       Stelling     AppSecret 75     55     56     ChannellD	😤 配置管理 🔷 🔨		MasterSecret *
initializity     AppSecret 75************************************	应用配置	AppKey cf************************************	
通道記賞 NetterSearct E F T O Clock	新建通道	AppSecret 75************************************	ChannellD
	通道配置	MasterSerret	Response after

Vendor Push Platform		Configuring in the IM cons
	★ 推送运营平台	Add vivo certificate
88 云通信IM ~	一	Package Name • Enter F
☞ 推送工具 ~		AppKey *
olo 推送统计	应用名称: 云通信IM 应用类别: 移动应用	AppID • Enter A
☞ 应用管理 ^	推送权限:正式	Receipt ID Enter
应用信息	审核状态: 已通过 创建时间:	Category Enter of
测试设备	应用包名:	AppSecret * Enter /
⊘ 标签管理	AppID:	
Q。在线诊断	AppKey: AppSecret: 雙貫 ① 恢复 ①	Response after Oper Click

For receipt configuration, please refer to: Message reach statistics configuration - vivo

Vendor Push Platform	Configuring in the IM cons



	Add Meizu certifi
应用配置 标签用户 问题排查 黑名单 回执管理 常用设备 多包名 任务备注	Package Name *
TUIKit	AppID *
应用名称 TUIKit	AppKey *
应用把态 普通应用 应用包名 添加多包名 应用类型 通讯社交 ✓	receipt switch
瓜用图标 更换图片 尺寸为480*480,500KBM内	AppSecret • Response after Click
<ul> <li>App ID</li> <li>App Key</li> </ul>	
① App Secret 重置	

For receipt configuration, please refer to: Message reach statistics configuration - Meizu

Vendor Push Platform	Configuring in the IM cons
	Add Honor certificate
	Package Name • Enter F
	AppID • Enter A
	ClientID • Enter ti
	ClientSecret • Enter t
	ChannellD Enter a
	Badge Parameter Please
	"Note: It
	Response after Oper Click

٢			
÷	┼放能力 / 推送服务 / 查看推送 ★表書は☆*PPタ	服务	
°~	且有推达服力		
47			
	心用类型:	移动则⊻用	
$\otimes$	应用名称:	腾讯云通信IM	
	应用包名:		
	SHA256证书指纹1:		
	申请时间:		
	APP ID:		
	APP Secret:		
	Client ID:		
	Client Secret:		
	Android端SDK:	《点击下载荣耀PUSH Android端SDK》	
	Android端接入文档:	《点击下载荣耀PUSH Android端接入文档》	
	服务端接入文档:	《点击下载荣耀PUSH服务端接入文档》	

Vendor Push Plat	endor Push Platform		
A Firebase		Add FCM certificate	
Project Overview	Project settings	Adding Method OUplos	
Generative AJ	Manyou ancio account permission (5	Message type O Notifi	
Authentication           Test Lab           Set Lab           Ressaying           J_1         Remote Config           O         App Check	Probase Admin SDK         Probase Admin SDK           Legacy condentials         Your Drabase services account on the used to authenticate multiple Freebase Features, such as Drabase services. Drage and Auth, programmatically vis the unified Admin SDK, Learninger (C)           Image: Database services         Probase service account	Transparı available SDK enh:	
What's new App Hosting KKR O Lata Connect KKR	All sarrice accounts                 Zerrice accounts             Zerrice             Zerrice accounts             Zerrice             Zericerice             Zerriceri	Package Name * Enter P	
Product categories Build ~ Run ~	<pre>var admin = require("firebase-admin"); var servicebccount = require("path/to/service&amp;ccountKey.json");</pre>	Upload certificate	
Analytics ~	asocratical activity (serviceAccount) ));	How to G	
	Generate new private key	ChannellD Enter a	

#### Note:

Regarding **Click for Subsequent Actions** supports the Report Statistics feature:

1. If you choose to open an app or a web page, purchasing the plugin will by default support reporting statistics.

2. If you choose to open a specified interface within the application:

For new certificate status, please directly use the auto-fill default value to support click statistics reporting.

If there was a certificate configured previously, continue using the old certificate and modify it to the default value to support reporting statistics, or regenerate a new certificate.

#### About FCM Data Messaging

FCM provides two push methods: Notification Message and Data Messaging.

Notification Message, with a simple style not differentiated by device, can support offline push once successfully integrated;

Data Messaging, offering rich customization for specific devices, supports reach and click reporting, and requires testing on the device before going live after integration.

The console defaults to Notification Message, and switching between modes can be done in the IM Console:

Add FCM certificate		
Adding Method	O Upload certificate	
Message type	Notification message O Transparent transmission (data) message	
	Transparent transmission (data) messages can be used to report push reach d available after activating <b>Push plug-in</b> . It only supports pixel phones that integ SDK enhanced version v7.8 and above	
Package Name *	Enter Package Name How to Generate an FCM certificate	
Upload certificate	Select file	
	How to Generate an FCM certificate	
ChannellD	Enter a channel ID	
	Confirm	

#### Note:

©2013-2022 Tencent Cloud. All rights reserved.

FCM Data Messaging capability is only supported on Pixel phones with TIMPush 7.8 and above, other manufacturers' devices need to be self-tested for support;

# iOS

Last updated : 2024-06-13 10:21:45

Before integrating the TIMPush component, you need to apply for an APNs Push Certificate from Apple and then upload the Push Certificate to the IM console. After that, you can proceed with the quick access steps. There are currently two mainstream types of certificates for Apple Manufacturer Configuration: p12 certificates and p8 certificates. Each type of certificate has its advantages and drawbacks, and you can choose one according to your needs.

#### **Certificate Type:**

p12 Certificate: A p12 certificate is a binary file containing both a public key and a private key, used for certificatebased authentication. It bundles the public key certificate and the private key into one file, with extensions .p12 or .pfx. p8 Certificate: A p8 certificate is an Auth Key, used for token-based authentication. It is a text file containing a private key, with an extension of .p8.

#### Validity and Management:

p12 Certificate: A p12 certificate typically has a one-year validity period, after which it needs to be regenerated and deployed. Each application requires a separate P12 certificate to handle push notifications.

p8 Certificate: A p8 certificate does not have an expiration date, so you don't have to worry about the certificate expiring. Moreover, using a P8 certificate can simplify certificate management, as you can use a single p8 certificate to provide push notification services for multiple applications.

#### Security:

p12 Certificate: A p12 certificate uses certificate-based authentication and requires the private key to be stored on the server. This could increase security risks, as the private key could be accessed by unauthorized users.

p8 Certificate: A p8 certificate uses token-based authentication, which means your server periodically generates a JSON Web Token (JWT) to establish a connection with APNs. This method is more secure, as it doesn't require storing a private key on the server.

#### Dynamic Island:

p12 Certificate: Not supported.

p8 Certificate: Supports Dynamic Island push notifications.

# 1. Using a p12 certificate (traditional push certificate)

#### Step 1: Apply for an APNs certificate

Enable remote push for the app

1. log in to Apple Developer Center website, click Certificates, Identifiers & Profiles or the sidebar's Certificates, IDs & Profiles, enter the Certificates, IDS & Profiles page.



#### É Developer

## **Certificates, Identifiers & Profiles**

Certificates	Identifiers 😌	Q AI
Identifiers	NAME ~	IDENTIFIER
Devices		
Profiles		
Keys		
More		
	Copyright © 2020 Apple Inc. All ri	ghts reserved. Terms of Use Privacy Policy

3. You can follow the steps below to create a new AppID or add a Push Notification Service to your existing AppID.

#### Note:

Your App's Bundle ID cannot use the wildcard \* , otherwise, the remote push service cannot be used.

4. Check the App IDs box, click Continue to proceed to the next step.

#### **É** Developer

 $\bigcirc$ 

### **Certificates, Identifiers & Profiles**

#### < All Identifiers

#### **Register a new identifier**

Continue

App IDs Register an App ID to enable your app, app extensions, or App Clip to access available services and identify your app in a provisioning profile. You can enable app services when you create an App ID or modify these settings later.

#### ○ Services IDs

For each website that uses Sign in with Apple, register a services identifier (Services ID), configure your domain and return URL, and create an associated private key.

#### O Pass Type IDs

Register a pass type identifier (Pass Type ID) for each kind of pass you create (i.e. gift cards). Registering your Pass Type IDs lets you generate Apple-issued certificates which are used to digitally sign and send updates to your passes, and allow your passes to be recognized by Wallet.

#### O Website Push IDs

Register a Website Push Identifier (Website Push ID). Registering your Website Push IDs lets you generate Apple-issued certificates which are used to digitally sign and send push notifications from your website to macOS.

#### O iCloud Containers

Registering your iCloud Container lets you use the iCloud Storage APIs to enable your apps to store data and documents in iCloud, keeping your apps up to date automatically.

#### O App Groups

Registering your App Group allows access to group containers that are shared among multiple related apps, and allows certain additional interprocess communication between the apps.

O Merchant IDs

5. Select App, click Continue to proceed to the next step.

6. Configure the Bundle ID and other information, click **Continue** to proceed to the next step.

ertifica	tes. Identifiers & Profil	es	
<ul> <li>All Identifiers</li> </ul>			
Register	an App ID		Back
Platform iOS, macOS, tvC	S, watchOS	App ID Prefix	
Description		Bundle ID • Explicit	
IMSDK Demo		com.imsdk.pushdemo	
You cannot use s	special characters such as @, &, *, ', ", -, .	We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	
ENABLED	NAME		
	🔍 Access WiFi Information 🕕		
	App Attest		
	Apple Pay Payment Processing ①		
	Apple Pay Payment Processing ①		
	Apple Pay Payment Processing () Associated Domains () AutoFill Credential Provider ()		

7. Check the **Push Notifications** box to enable the remote push service.

< All Identifier	S	
Regist	er an App ID	Back Continu
	<b>√→</b> Multipath (i)	
	Network Extensions	
	N) NFC Tag Reading	
	VPN Personal VPN	
	Push Notifications (i)	
	Sign In with Apple      Configure	
	SiriKit 🗊	
	System Extension	
	OUSer Management	
	Wallet ①	
	Wireless Accessory Configuration	
	Mac Catalyst (Existing Apps Only) (i) Configure	

#### **Certificate Generation**

1. Select your AppID and choose **Configure**.

< All Ident	tifiers	
Edity	your App ID Configuration	Remove Save
	Network Extensions (i)	
	N) NFC Tag Reading	
	VPN Personal VPN	
	Push Notifications	Configure Certificates (0)
	Sign In with Apple	Configure
	SiriKit 👔	
	System Extension	
	User Management	
	Wallet	
	Wireless Accessory Configuration ①	
	Mac Catalyst (Existing Apps Only) (i)	Configure

2. In the **Apple Push Notification service SSL Certificates** window, there are two SSL Certificates for the development environment (Development) and the production environment (Production), as shown below:

É Developer	Apple Push Notification service SSL Certificates	Yardiac M Bao Ma - 304735703
<b>Certificat</b> To configure push notifications for this App ID, a Client SSL Certificate that allows your notification server to connect to the Apple Push Notification Service is required. Each App ID requires its own Client SSL Certificate. Manage and generate your certificates below.		
< All Identifiers Development SSL Certificate		
Edit your App Create an additional certificate to use for this App ID.		Remove
Platform	Create Certificate	
iOS, macOS, tvOS, watchO	Production SSL Certificate	
Description           TPNS SDK demo         Create an additional certificate to use for this App ID.		
You cannot use special cha Create Certificate		
Capabilities		
ENABLED NAME	Done	
	cess WiFi Information ①	
- 🐼 Ap	p Attest (i)	
D GĐ Ap	p Groups (i) Configure	
- 💼 Ap	Apple Pay Payment Processing (i)	
	sociated Domains (i)	

#### 3.

#### We

first select the **Create Certificate** for the Development environment, the system will prompt us that we need a Certificate Signing Request (CSR).

< All Cert	ificates Ite a New Certificate Back Continu
<b>Certific</b> Apple Pu	<b>ate Type</b> Ish Notification service SSL (Sandbox)
Platform	:
iOS	
Platform iOS	a Certificate Signing Request

4. On a Mac, open Keychain Access tool, in the menu select Keychain Access > Certificate Assistant >

Request a Certificate From a Certificate Authority ( Keychain Access - Certificate Assistant -

Request a Certificate From a Certificate Authority ).

Keychain Access F	le Edit	View Window Help	
About Keychain Access	5		
Settings	ж,		
Certificate Assistant	>	Open	
Ticket Viewer	<b>ご</b>	Create a Certificate	
Services	>	Create a Certificate Authority Create a Certificate For Someone Else as a Certificate Authority	
Hide Keychain Access	ж н	Request a Certificate From a Certificate Authority	
Hide Others	ъжн	Set the default Certificate Authority	
Show All		Evaluate a Certificate	
Quit Keychain Access	жQ	Keychain Access	

5. Enter your email address, Common Name (your name or company name), select **Save to disk**, click **continue**, the system will generate a \*.certSigningRequest file.

	Certificate Assistant
	Certificate Information
	Enter information for the certificate you are requesting. Click Continue to request a certificate from the CA.
Cen	User Email Address: youremail@example.com Common Name: IMSDK CA Email Address: Required Request is: O Emailed to the CA Saved to disk Let me specify key pair information
	Continue


6. Go back to the page on the Apple Developer website mentioned in Step 3, click Choose File to upload the

generated \*.certSigningRequest file.

All Certificates	
Create a New Certificate	Back Continue
<b>Certificate Type</b> Apple Push Notification service SSL (Sandbox)	
Platform:	
iOS	~

7. Click **Continue** to generate the push certificate.

evelope	
ertifi	cates, Identifiers & Profiles
< All Certifi	cates
Creat	e a New Certificate Back Continue
Certificat Apple Pusł	e Type Notification service SSL (Sandbox)
Platform:	
iOS	v
Platform: iOS Upload a To manuall Learn more	Certificate Signing Request / generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac.
Choose I	ile CertificateSigningRequest.certSigningRequest
	Copyright © 2020 Apple Inc. All rights reserved. Terms of Lise Privacy Policy

8. click **Download** to download the Development SSL Certificate to your local environment.

É Developer		
Certificates, Ide	entifiers & Profiles	
< All Certificates		
Download Your Certi	ficate	Revoke
Certificate Details		
Certificate Name com.tpnssdk.pushdemo	Certificate Type APNs Development iOS	Download your certificate to your Mac, then double click the .cer file to i Keychain Access. Make sure to save a backup copy of your private and p computer accura
Expiration Date	Created By	somewhere secure.

9. Repeat steps 1 - 8 above to download the Production SSL Certificate for the production environment to your local machine.

Note

The certificate for the production environment is actually a combined certificate of Development (Sandbox) + Production, and it can be used as a certificate for both the development and production environments.

	natificano O Drofilo -		
ertificates, Ide	entifiers & Profiles		
< All Certificates			
Create a New Cer	tificate	E	Back
Certificate Type Apple Push Notification service SSI	(Sandbox & Production)		
Platform:			
iOS	~		
Developer			
Developer ertificates, lo	dentifiers & Profiles		
Developer <b>ertificates, Ic</b> Certificates	dentifiers & Profiles		
Developer <b>ertificates, lo</b> Certificates wnload Your Cert	dentifiers & Profiles	R	evoka
Developer <b>ertificates, lo</b> Certificates wnload Your Cert	dentifiers & Profiles	R	evoke
Developer ertificates, Ic Certificates wmload Your Cert rtificate Details ficate Name tpnssdk.pushdemo	dentifiers & Profiles tificate	R Download your certificate to your Mac, then double click Keychain Access. Make sure to save a backup copy of yo	evoke

10. Double-click the downloaded SSL Certificate for the development and production environments. The system will import it into the keychain.

11. Open the Keychain App, go to log in to > My Certificates, right-click to export the newly created Apple Development IOS Push Services and Apple Push Services for the development and production environments as p12 files respectively.

•••	Keychain Access		Z i	Q Search		
Default Keychains	All Items Passwords Secure Notes My Cert	tificates Keys Certificates				
୍ର login ଟ Local Items System Keychains	Certificule Issued by: Apple Worldwide Develo Expires: Thursday, December 5, 20 This certificate is valid	per Relations Certification Authority 24 at 15:52:39 China Standard Time				
🔒 System						
System Roots	Name	and a second	Kind Date Modified	l enem l	Expires	Keyc
	Apple Persistent State Encryption		application password Yesterday, 15:	09	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	login
	Apple Push Services: com.	).xa	certificate		Oct 20, 2023 at 10:38:18	login
	Apple Push Services: com	ха	certificate		Dec 5, 2024 at 15:52:04	login
	Apple Push Services: com.t	J.xa.db	certificate		Mar 2, 2024 at 10:57:19	login
	Apple Push Services: com.	New Identity Preference		100°	Dec 5, 2024 at 15:52:39	login
	Apple Worldwide Developer Relations Cer				Feb 20, 2030 at 08:00:00	login
	Apple Worldwide Developer Relations Cert	Copy "Apple Push Services: com.	pushservice"		Mar 19, 2036 at 08:00:00	login
	Apple Worldwide Developer Relations Cer	Delete "Apple Push Services: com	"pushservice"		Feb 8, 2023 at 05:48:47	login
	Apple Worldwide Developer Relations Cer				Jan 24, 2025 at 08:00:00	login
	Apple Worldwide Developer Relations Cer	Export "Apple Push Services: com.t	.pushservice".		Dec 10, 2030 at 08:00:00	login
	AppleIDClientIdentifier			7:54		login
	Basic Attestation User Sub CA1	Get Info			Mar 22, 2032 at 08:00:00	login
	BeaconStore	Evaluate "Apple Push Services: com	.pushservice	e" <sup>:02</sup>		login
y Horis	Apple Worldwide Developer Relations Cerr  AppleIDClientIdentifier Basic Attestation User Sub CA1 BeaconStore	Export "Apple Push Services: com.t Get Info Evaluate "Apple Push Services: com	.pushservice"	····· 7:54	Dec 10, 2030 at 08:00:00  Mar 22, 2032 at 08:00:00 	lo lo lo

#### Note

Set a password when saving the .p12 file.

# Step 2: Upload the certificate to the console

- 1. Log in to the Chat Console.
- 2. Click Plugin Service-Push-Access Settings to enter the access settings page

Chat	Access settings Current data center: Singapore ()	Telegram group WhatsApp group			
표· Application Management 맘 Overview	The push plug-in trial is about to expire and push services (including offline push of ordinary m  Push Overview The trial Remaining 3 days All members/tap push interface call frequency10	essages, push to all members/labels, etc.) will be autom	atically stopped. To avoid affecting the normal u	se of your business, please <u>Purchase</u>	
으 Users 롦 Groups 먋 Configuration ~	Normal push     Activated	Function Preview	Function Preview O Push Records Activated	Function Preview O Push data Activated	<ul> <li>Push troubleshoot</li> <li>Activated</li> </ul>
Statistics  Plugin  Push ^	Purchase New the trial           1         Manufacturer configuration           IManufactore configuration         IManufactore configuration	messaging IM channel, which is fast and reliable; for offi	ine push, it is recommended that you use the sy	vstem-level push channel provided by each m	nanufacturer. The system The leve
Access settings   Access test  Push Record  Buck Data	more stable system-level long connection, and the resource consumption is greatly reduced.           Android         IOS           Add Certificate				
Push Troubleshooting     Al Chatbot					
<ul> <li>Desk</li> <li>Tools</li> <li>Monitor</li> </ul>	No connecte yer				
Dev Tools v	Local deployment  After purchasing the push plug-in, you can use the "quick configuration" and "download configuration"	iguration file <sup>*</sup> functions. With 3 lines of code and 2 minut	tes, you can complete push access from multip	e manufacturers at once.	

- 3. Click Add Certificate at the bottom of iOS in Vendor Configuration.
- 4. Select the certificate type, upload the iOS Certificate (.p12), set the certificate password, and click Confirm.

Access settings 2 <sup>r</sup> st -		ingapore (i) Telegram group WhatsApp group		
① The push plug-in trial is about to expire and push service				
Push Overview         free trial Remaining 3 days         All member	ers/tag push interface call	frequency100 times/day Edit		_
				Function Preview
Activated	Add iOS Certificate	3	×	Activated
	Push Type	O APNs Push VolP Push		
Purchase Now free trial	Cartificata Tupa			
	Certificate type	Please check that the uploaded certificate type is Apple Push Notification service SSL (Sandbox &		
		Production), and test it after the Archive releases the Release package. Note: It cannot be tested in Xcode.		
Manufacturer configuration	Configuration type	<b>o</b> p12  p8		
IM supports online and offline push notification. On	iOS certificate (.p12) *	Select file		-level push channel provided by
more stable system-level long connection and the re		How to concerte an APNe contificate?		
Android iOS				
Add Certificate	mutable-content 🛈			
	Certificate password *	Enter the certificate password		
		Que Berry		
		Comment		
No certificat				
After purchasing the push plug-in your sections the family	ick configuration" and "do			anufacturers at once
Aller purchasing the pash plag-in, you can use the "qu	ion confinguration and do			

#### Note:

We recommend naming the uploaded certificate in English (special characters such as brackets are not allowed). You need to set a password for the uploaded certificate. Without a password, push notifications cannot be received. For an app published on App Store, the environment of the certificate must be the production environment. Otherwise, push notifications cannot be received.

The uploaded .p12 certificate must be your own authentic and valid certificate.

5. After the pending certificate information is generated, record the certificate's ID.



# 2. Using a p8 certificate (supports Dynamic Island push notifications)

p8 Certificate: A p8 certificate does not have an expiration date, so you don't have to worry about the certificate expiring. Moreover, using a p8 certificate can simplify certificate management, as you can use a single p8 certificate to provide push notification services for multiple applications. In addition, p8 certificates support Dynamic Island push notifications.

# Step 1: Apply for an APNs certificate

To create a p8 certificate file, first log in to Apple Developer Center.

	Certificates, Identifiers & Profiles - Apple Developer
É Developer	1
Certificate	es, Identifiers & Profiles
Certificates	Keys
Identifiers	
Devices Profiles Keys More	Getting Started with Keys Creating a key allows you to configure, authenticate, and use one or more Apple services for that key. Unlike certificates, keys do not expire and can be modified to access more services after their creation. For more information on creating and using keys, refe to Developer Account Help. Create a key

Copyright © 2019 Apple Inc. All rights reserved. Terms of Use Privacy Policy

1. Enter Certificates, Identifiers & Profiles: In the top right corner of the page, click **Account**, then select **Certificates**, **Identifiers & Profiles** from the dropdown menu.

2. To create a new App ID: in the left-hand menu, click **Identifiers**, then click the + on the right to create a new App

ID. Fill in the relevant information and click **Continue**.

3. To create a new key: in the left-hand menu, click **Keys**, then click the **+** on the right to create a new key. Enter the name of the key, then check **Apple Push Notifications service (APNs)** and click **Continue**.



Confirm and generate the key: On the confirmation page, verify your key information, then click **Register**. Next, you'll see a page prompting you to download the key. Click **Download** and save the generated .p8 file to your computer.

#### Note:

The p8 certificate can only be downloaded once; please save it properly.

Please safeguard the downloaded p8 file, as you will not be able to download it again. You can use this p8 certificate to configure your iOS applications to receive push notifications.

# Step 2: Upload the p8 certificate to the IM console

1. Log in to the Chat Console.

2. Click Plugin Service-Push-Access Settings to enter the access settings page

Chat	Access settings	Current data center: Singapore ①	Telegram group WhatsApp group			
로는 Application	<ol> <li>The push plug-in trial is about to expire and</li> </ol>	f push services (including offline push of ordinary mes	sages, push to all members/labels, etc.) will be autor	natically stopped.To avoid affecting the normal u	se of your business, please <u>Purchase</u>	
문 Overview 은 Users	Push Overview free trial Remaining 3 days	All members/tag push interface call frequency100 t	mes/day Edit			
ය Groups 믎 Configuration ~	Normal push     Activated	Function Preview      All members/tag push     Activated	Function Preview      Quick integration     Activated	Function Preview O Push Records Activated	Function Preview O Push data Activated	<ul> <li>Push troubleshoot</li> <li>Activated</li> </ul>
<ul><li>Webhook</li><li>Statistics</li></ul>	Purchase Now free trial					
Plugin C3 Push Access settings Access settings Push Record Push Data Push Troubleshooting Al Chatbot Desk Tools	Manufacturer configuration     Manufacturer configuration     Marufacturer configuration	ations. Online push is delivered through the instant me and the resource consumption is greatly reduced.	essaging IM channel, which is fast and reliable; for of	fline push, it is recommended that you use the s	vstem-level push channel provided by each r	nanufacturer. The system The leve
⑦ Monitor ☐ Dev Tools	2 Local deployment					
	After purchasing the push plug-in, you can	use the "quick configuration" and "download configu	ration file" functions. With 3 lines of code and 2 minu	utes, you can complete push access from multip	le manufacturers at once.	
E	Quick configuration	Manua	Il configuration			

- 3. Click Add Certificate at the bottom of iOS in Vendor Configuration.
- 4. Select the .p8 certificate

Push Type	O APNs Push VoIP Push
Certificate Type	Production environment     Development Environment
	Please check that the uploaded certificate type is Apple Push Notification service SSL (Sandbox & Production), and test it after the Archive releases the Release package. Note: It cannot be tested in Xc
Configuration type	○ p12 ○ p8
iOS Certificate (.p8) *	Select file
	How to generate an APNs certificate?
mutable-content 🛈	
KeyID *	Enter
TeamID *	Enter
BundleID *	Enter

# Note:

**Key ID**: This is the unique identifier for your APNs Auth Key. When you create a new APNs Auth Key in the Apple Developer Center, a Key ID will be generated for you. You can find it in the "Certificates, Identifiers & Profiles" section under "Keys".

**Team ID**: This is the unique identifier for your developer account. You can find it on the account details page of the Apple Developer Center. Click "Membership" in the upper right corner, and you can find your Team ID in the "Membership Details" section.

**Bundle ID**: This is the unique identifier for your application, also known as the app ID. You can find it in the "Certificates, Identifiers & Profiles" section of the Apple Developer Center. Select "Identifiers", then find the corresponding Bundle ID in your list of applications.

# uniapp

Last updated : 2024-06-13 10:21:45

TencentCloud-TIMPush is a Tencent Cloud Chat Push plugin. Currently, the push supports channels from manufacturers such as Mi, Huawei, Honor, OPPO, Vivo, Meizu, APNs, OnePlus, realme, iQOO, and Apple. Android

iOS

# Register an application with the vendor push platform

To utilize the offline push feature, you need to register your app on each vendor's push platform to obtain parameters such as AppID and AppKey. Currently, the mobile manufacturers supported in China include: Mi, Huawei, Honor, OPPO, Vivo, Meizu, and internationally Google FCM is supported.

Mi Huawei OPPO vivo Meizu HONOR Google FCM

# Step 1: Register a Xiaomi Developer Account

Visit Xiaomi Open Platform, register a Xiaomi developer account. For details, refer to Enterprise Developer Account Registration Process.

#### **Step 2: Create an Application**

1. In the management console, click Message Push.

#### Note:

If you log in with a personal account, it will display "Sorry, you currently do not have push/review permissions".

分发服务				
<b>应用</b> 》 <sub>成布手机</sub>	<b>挤戏</b> 、平板应用与游戏	↓ ↓ 电视应用	<mark>RPK</mark> 小游戏	(
应用服务				
消息推送	<b>必</b> 账号服务	<b>テ</b> テ 云測服务	安全开放服务	
推广变现				
室销平台	(学) 广告联盟			

2. Create an application, complete the application information page, and click **Save**.

□□ 小米开放 ጓ	2 <b>台</b> ·推送运营平台	语言:	中文 🔹	文档
●● 全部应用	创建应用 🗸			应用
	应用列表			

# Note:

The application package name should be consistent with the plugin application package name.

创建应用			
	* 默认语言 🌒:	简体中文	$\vee$
	* 操作系统:	● Android ○ IOS(仅能使用推送服务和统计服务)	
	* 应用名称:	云通信 IM 离线推送测试	22/30
	* 应用包名:	comt.demolpush	

# Step 3: Enable Push

Go to the **Application List** page on the Push Operation Platform, click on the corresponding application name to **Enable Push**, and confirm to enable.

□□ 小米开放平台	·推送运营平台地区: 中国大陆 • 语言:	中文 🔻	
₿ 全部应用	创建应用 🖌		
	应用列表		
	应用名称	平台类型	启用状态
		•	已启用
		•	已启用
		•	已启用
	云通信 IM 离线推送测试	•	未启用

Step 4: View and Obtain Application Information

Enter the **Application Information** page on the Push Operation Platform, and review the **Application Information**.

<b>同</b> 小米开放平台	• • 推送运营平台	地区: 中国大陆	- 语言:	中文 🔻		
□□ 全部应用	创建应用 🚽					
	应用列表					
		应用名称		平台类型	启用状态	
		Urgetter.		•	已启用	
				•	已启用	
		. Jimix		•	已启用	
□□ 小米开放平台	・推送运营平台	TUIKit 👻				
✔ 推送工具	TUIKit					
推送统计	应用类型	Android				
■ 应用管理 ~	创建时间					
● 应用信息	主包名			设置多包名 了解参	9包名使用方法	
通知类别						
角色管理	AppID					
♀ 调查工具	АррКеу	查看				
三 法律文档	AppSecret	上传				

# Step 5: Configure Push Certificate

Log in to Tencent Cloud Chat Console, go to **Push** > **Access Settings** feature section to add vendor push certificates, and configure the parameters such as AppId, AppKey, AppSecret obtained from the vendor to the added push certificate.

Vendor Push Platform	Configuring in the IM console
	<b>Note:</b> <b>Specified in-app page</b> link cannod dispatching event listening for the configuration of in-app page navigation of

☆ 推送统计	TUIKit 应用类型 Android
<ul> <li>应用管理 ~</li> <li>应用信息</li> </ul>	0(1895)0
通知类别	主信名 设置多包名 了解多包名使用功法 AcpiD
♀ 调查工具	AppKey 288 AppSecret 288
三 法律文档	B-6.28 EP



#### Step 1: Register a Huawei Developer Account

Go to Huawei Developer Alliance, register for a Huawei Developer Account. For details, please refer to Register Account.

#### Step 2: Create an Application

1. In the Account Center, click My Projects to add a new project.

AppGallery Connect	全部服务 > 解决方案 >		
我的应用	1 我的项目	应用分析	<b>上</b> 用户与访问
我的收藏	14 TC to		
向小姐之 主的加密 的不能过失 加,能加强到	2012年4.		
常用服务			
😑 应用发布	🍠 推送服务	🔓 社区管理	反 应用下载直达

2. In the Project Settings section, click Push Service > Activate Now.



3. Click Project Settings > API Management to enable permissions for the push service.

AppGallery Connect	全部服务 ~	我的项目 ~		开发机构	◇ ● 开发林	几构 〜
项目设置 公利 个	常规 实时监测	API管理 J渲染、启动、卡顿等应	Server SDK 用性能问题,并根排	<b>项目套餐</b> 居报告信息改善应用性能。	<b>项目配额</b> 自动化性能跟踪:	<b>项目费用</b> 自动采集应用启录
。 应用联运						
<i>》</i> 游戏联运	增长					
₹₀ 付费下载						
🔄 应用内支付服务	<i>\$</i> 推	挂送服务				
二 华为钱包			/// 1E.NU - 16.NI - 7.	** <u>1 11 10 16 10 16 10 17 - 80</u>		

#### **Step 3. Add Application**

Click **Project Settings > General** to add an application.

# Note:

The application package name should be consistent with the plugin application package name.

AppGallery Connect	全部服务 ~	我的项目 ~				
项目设置	常规	API管理	Server SDK	数据处理位置	项目套餐	项目配额
HarmonyOS应用    ~	添加应用					
盈利 <sup>•</sup> ^	项目中还没?	<b>———</b> 有应用,请先添加应用	1			
32、应用联运						
<i>&amp;</i> 游戏联运	开发考					
Ĭ 付费下载	ЛХН	Developer ID: (?)				
🔄 应用内支付服务		验证公钥: ⑦				Q
● AGD Pro应用变… ~	而日					
🦉 华为支付服务 ( 🔤	201	项目名称:	Ø			
增长~		项目ID:				
须 推送服务		数据が理습業: ①	山田(明34) (第	₩ ₩		
<sup>┣</sup> ́в A/B测试		如油红生位目: ①		1×±		
· S· 动态标签管理		客户端ID:	Client ID (?)			
💮 远程配置		ΔPI宓钼 (恁抿):				
☴ 应用内消息						
<u>前</u> 预测	删除项目					
云开发(Serverless) ~						
构建 ~ ~						
质量●    ~						
华为分析 ~	formention					

# Step 4: Obtain Application Information

Click **Project Settings > General** to obtain application information.

# Note:

The General page contains the Client ID and Client Secret for both projects and applications, which have inconsistent parameters. Please scroll down to the bottom of the page to obtain the application's Client ID and Client Secret. You must add the SHA256 Certificate Fingerprint, which should match the SHA256 Certificate Fingerprint of your own packaging certificate.

Download the agconnect-services.json file and place it in the project under: **nativeResources/android/assets**/ path. If you modify the project, application information, or development service settings, you need to re-download the agconnect-services.json file.

AppGallery Connect	全部服务 ~	我的项目 ~					te
项目设置	常规	API管理	Server SDK	数据处理位置	项目套餐	项目配额	项目费用
HarmonyOS应用 ~	开发者						
云开发 (Serverless) ~		Developer ID:					
质量		验证公钥: 🥐				. (	
増长       へ         が       推送服务         (回)       A/B測试         (公)       动态标签管理         (公)       远程配置         (二)       应用内消息         (二)       App Linking         (回)       预测	项目	项目名称: 项目ID: 数据处理位置: ⑦ 客户端ID: API密钥 (凭据):	test-push 久 中国(默认) 管 Client ID ⑦ Client Secret ⑦	理		Ū.	Q
构建 <sup>●</sup> ~ 盈利 <sup>●</sup> ~ 祭 应用联运	应用	SDK配置:	下载最新的配置文件 <u> 业</u> agconn	-(如果您修改了项目、) ect-services.json	立用信息或者更改了 不包含語	某个开发服务设置,『 容钥 ⑦	可能需要更新该文件
<ul> <li>(行费下载)</li> <li>(更) 应用内支付服务</li> <li>(正) 华为钱包</li> <li>(子) AGD Pro应用变 ~</li> </ul>	Dov the	wnload th project u <sup>4A256证书指纹:</sup> ②	e agconn inder: nat • SH	ect–servic iveResour	ces.json ces/and ate fingerpr	file and p roid/asso int must be	blace it i ets/ patl configured
<ul> <li>※ 华为支付服务 ( www</li> <li>开发管理 ~</li> <li>华为分析 ~</li> </ul>	OAuth 2	2.0客户端ID(凭据): 回调地址: ⑦	Client ID Client Secret	• Ap	pID pSecret		Q
<b>全新自定义菜单栏</b> 您可在菜单栏中自定义添加/隐 藏服务,以更便提高效地访问	删除项	1	删除应用				

# Step 5: Add Push Certificate

Log in to Tencent Cloud Chat Console, click **Push** > **Access Settings** to add vendor push certificates, and configure the obtained vendor's AppId, AppKey, AppSecret, and other parameters to the added push certificate.

Vendor Push Platform	Configuring in the IM console
	Note: Client ID corresponds to App <b>Specified in-app page</b> link dispatching event listening fo configuration of in-app page I



项目设置	常规 API管理	Server SDK 项目整顿 项目配额 项目费用	Add Huawei c	ertificate
盈利 ^	AF-102103(2038)。		Package Name •	Enter Pa
<ul> <li>(Y) 新な鉄道</li> <li>(F) 付费下载</li> <li>(F) 応用由本が紹名</li> </ul>	SDK配置:	下数量新的記室文件(20年8日ゆ27月日。反用信息或者要定了其个开发服务设置。可能需要更新这文件)	AppID *	Enter Ap
<ul> <li>単为钱包</li> </ul>	55.		Category	Enter ca
l长 ^ ∅ 推送服务	BA: APP ID:			
A/B测试	SHA256证书指纹: ⑦	2 © 2 ©	AppSecret *	Enter A
运程配置 应用内消息		2 0 2 0	ChannellD	Enter a
App Linking     历 短信服务	OAuth 2.0客户端ID(凭据):	Cient ID	Badge Parameter	Please
✓ Indexing		Client Secret		*Note: It is
() 1000 () 100			Response after	ㅇ Open
			Click	
			select O	nen anr

#### Note:

Notification Bar Push: The app needs to be published on the OPPO Software Store;

Notification Bar Push Test Permission: Only 1,000 messages can be pushed per day, limited to testing purposes. After the app is published, you need to reapply for "Notification Bar Push" permission to obtain a normal message

push quantity;

The platform will return review results within 1 business day. Developers can view the review results on the application page. For other inquiries, consult the Open Platform Customer Service.

#### Step 1: Register an OPPO Developer Account

Enter OPPO Open Platform, register an OPPO Developer Account. For details, see OPPO Enterprise Developer Account Registration.

#### **Step 2: Create an Application**

Enter the OPPO Open Platform, click **Product** > **Application Distribution**> **OPPO Software Store** > **Release Application** to access the Management Center and create an application.

<b>ОРРО</b> 开放平台	产品へ 活动 社区	学堂 文档中心 商城 ~		
		开发		
		移动服务	输入产品名称,快速查找产品	Q 远程真机
		分发		
		应用分发	概览	
		内容分发	OPPO 软件商店	
		服务分发	OPPO 游戏中心	
		推广	OPPO 小游戏	
		变现	OPPO 游戏服务	
			OPPO 应用合作	
			创建应用	
			开发者认证流程 🍐	
			团队账号	
			接入软件商店	
			接入合作游戏	

# Step 3: Activate the PUSH Service

1. Enter the OPPO Open Platform, click **Product** > **Mobile Services** > **Push Service** to access the Push Homepage, then click **Apply for** access to activate the Push Service.

OPPO 开放平台	产品へ 活动 社区	学堂 文档中心 商城 ~			
		开发 移动服务	输入产品名称,快速查找产品	Q. 远程真机	移动应用加固服务数据服务
		分发			
		应用分发	应用服务		安全服务
		内容分发	账号服务		恶意URL检测能力
		服务分发	推送服务		终端环境安全检测能力
		推广	游戏SDK		日志防泄漏能力
		夺现	广告服务		移动应用加固服务 🖕
		~~~	钱包服务		隐私安全检测服务 🌢
					可信数字身份密钥服务 New
			潘塔纳尔 🍐		
			潘塔纳尔系统		系统能力
			服务流转		Hyper Boost
			服务入口模板		网络能力
			多模态交互		窗口能力

2. Click to enter the **Management Center** > **Application List** > **Apply for Push Services** interface, to apply for push permissions for applications not yet enabled.

# Note:

Enabled Services: Applications that have applied for PUSH permissions and have been approved. Disabled Services: Applications that can apply for PUSH permissions.

орро яфт	Z台	管理中心首	f页 产品 ∨								 	
: 应用列表		I	0 8 0	junkmenTestDem	o v							
☑ 创建推送	~	ŵ	应用列表 🗙									
┢ 数据统计	~	首页 > 6	拉用列表									
◎ 推送审核	~	请输	)入应用名称或包名或a	ippld搜索	卖							
② 配置管理	~		应用名称		应用包名		Appld (2)		推送类型 ⑦	累计用户 ②	可推送总数(条/日) ②	消息审核
			junkmenTestDemo						测试推送	0	1000	(已开,
			opushDemo						测试推送	0	1000	(已开,
			tuikit						测试推送	475	1000	(已开)
☆ 生态服务	_		主称版分 / 四	1月11页95								
应用服务			移动应用		~	推	送服务					
内容服务			快应用		~	系统级創	蚊达, 无拦截, 亿级	建送能力				
智慧服务			小游戏		~	已开启服	绣					
白 API服务		,	大屏应用		~							
A 开发者中心		/	智慧服务		~							
			开发服务		^			<b>U</b> TT				
			游戏服务SD	K		Opush 普通	<b>Demo</b> 应用	OpushDem 普通应用	0			
		3	帐号服务									
			消息推送			未开启服	踴					
			ARUnit									
			Hyper Boos	st								
			CameraUni	it								

3. Click **Apply for activation**. In the Disabled Services, click on the application that needs to apply for PUSH permissions, enter the PUSH Service and click **Apply for activation**.

	^	4	<b>主态服务</b> > 应用服务									
应用服务			移动应用	~		AppKey:	3c*********	********	****		查看	
内容服务			快应用	~	;	注意: 以上秘钥需要	要先申请开通	"消息推送"	权限才可有效使用。	请点击如下	"申请开通"	按钮申请消息推
智慧服务			小游戏	~								
🖾 API服务			大屏应用	~	Ę	服务开启详情						
<b>久</b> 开发者中心			智慧服务	~								
			开发服务	^		能力项			简介		状	초
			游戏服务SDK			通知栏推送			通过ColorOS系统级通 息	通道,推送通知林	送信 已沢	开通
			帐号服务			通知栏推送测试机	又限		推送测试权限,只能用 需要重新申请"通知档	]于测试,正式机 {推送"权限(}	Q限 資源 未予	开通
			消息推送						上架后方可申请)。			L

Step 4: Add Push Certificate

Log in to Tencent Cloud Chat Console, go to Push > Access Settings feature section to add vendor push

certificates, and configure the parameters such as AppId, AppKey, AppSecret obtained from the vendor to the added push certificate.



#### Note:

If the application is not listed in the App Marketplace, push permissions are restricted. Official messages cannot be sent through the Vivo official website's Web Interface or API backend. However, test messages can be sent to set test devices through the API backend for testing.

#### Step 1: Register a Vivo Developer Account

Enter Vivo Open Platform, register a Vivo Developer Account. For details, see Vivo Enterprise Developer Account Registration.

# Step 2. Create an application

Enter Vivo Open Platform, click Distribution > Application Distribution > App Store > Upload Application to create your application.

vivo 开放平台	首页	开发 ~	分发 ^	推广变现	文档中心	联系我们					
						<u>应用分发</u> 内容分发 服务分发	概览 应用商店 游戏中心 快应用 小游戏 近警 推送服务 数据服务 A(B测试	<b>测</b> 云 自 如 道 道 道 道 。 二 章 。 章 。 二 章 。 二 》 章 。 章 。 章 。 章 。 章 。 章 。 章 。 章 。 章 。 章	试 :真机 :私自检 动化测试 :护 !以账号 !双转移 ;用认领	发布 上传游戏 上传统成用 上传小游戏 上传运用	1
							活动推广 评论管理				

#### **Step 3:Enable Push Notification**

Enter the Management Center and click **Push Service > Push Notification Application** to apply for enabling push for the newly created application.



#### Step 4: Obtain Application Information

Go to the Push Operation Platform and click **Application Management > Application Information** to obtain application information.

推送申请			
应用名称 🕈	应用类别 ▼	推送权限 🔻	审核状态 🔻
	移动应用	受限	
	移动应用	001	
		337. DFI	中*** +*

# Step 5: Add Push Certificate

Log in to Tencent Cloud Chat Console, click **Push** > **Access Settings** to add vendor push certificates, and configure the obtained vendor's AppId, AppKey, AppSecret, and other parameters to the added push certificate.

Vendor Push Platform	1	Configu	iring in the IM	l console
		Note: Specifi dispatc configu	<b>ed in-app p</b> hing event lis ration of in-a	<b>age</b> link stening fo op page i
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	1 推送运营平台			
음 云通信IM ∨	应用信息		Add vivo certifi	cate
☞ 推送工具 🗸			Package Name •	Enter Packa
。加 推送统计	应用名称: 云通信IM			
☞ 应用管理 ^	推送权限: 正式		AppKey *	Enter AppKe
应用信息	审核状态: 已通过		AppID •	Enter AppID
测试设备	创建时间: 应用包名:		Receipt ID	Enter
⊘ 标签管理	AppiD:		Category	Enter catego
<b>%</b> 在线诊断	AppKey: AnnSecret: 現實 0 恢复 0		AppSecret •	Enter AppSe
			Response after Click	Open appli
			select Op	en applicatio

# For receipt configuration, refer to: Message Delivery Statistics Configuration > VIVO

#### Step 1: Register a Meizu Developer Account

Register a Meizu developer account, see Developer Registration for details.

# Step 2: Create an Application

1. Click Console >Flyme Push.



2. After entering the application information, create the application.

# Note:

The application package name should be consistent with the plugin application package name.

Flyme 推送平台	首页				
┃应用列表				按应用搜 >  全	·部应用
应用名称	应用包名	应用形态	AppID	在线用户数	当
		普通应用	ITUL	0	



In the application list, click **Open Application**. Enter the configuration management page to obtain application information.

Flyme 推送平台	首页			
应用列表				按应用搜
应用名称	应用包名	应用形态	AppID	在线用户数
	<b>,</b> ne	普通应用		0

### Step 4: Add Push Certificate

Log in to Tencent Cloud Chat Console, click **Push** > **Access Settings** feature bar to add various manufacturer's push certificates, and configure the obtained parameters such as AppId, AppKey, AppSecret to the added push certificate.

Vendor Push Platform	Configuring in the IM console
	Note: Specified in-app page link of dispatching event listening for configuration of in-app page n
Flyme 推送平台 首页 创建推送 数据统计 配置管理	
应用配置 标签用户 问题排查 黑名单 回执管理 常用设备 多包名 任务备注	
TUIKit	Add Meizu certificate
应用名称 TUIkit	Deploye Name a
四用形态 曹通应用	Package Name • Enter Package
□□川田山石 液加1/2 毛名 液加1/2 毛名	AppID • Enter AppID
应用图标 更换图片 尺寸为480~480,500KB以为	AppKey • Enter AppKey
	receipt switch   After turning on 1  configure the rec
(i) App ID	AppSecret • Enter AppSecr
App Key     App Secret	Response after Open applica Click

#### For receipt configuration, refer to: Message Delivery Statistics Configuration > Meizu



# Step 1. Register an HONOR Developer Account

Register an HONOR Developer Account, for details see Developer Registration.

#### Step 2: Enter the Management Center Page.

简体中3	<b>HONOR</b> 探索 生态服务 开放能力 文档中心
	团 队 帐 号 团队帐号功能已上线,团队协作,安全高效, 多种角色权限按需配置,赋能开发者业务蓬勃 发展,快来体验吧! 马上体验 了解详情
安全共享 ・ 无缝协作	

#### Step 3: Enter the Push Service List

Click **Push Service** to access the Push Service list page.

HONOR			
⑥ 生态服务	开放能力		
。名。开放能力	0		8
测试服务	<b>帐号服务</b> 基于OAuth2.0标准协议,荣耀帐号授权登录,接 入快捷简单	<b>安全授权</b> 接入荣耀MDM能力,携手服务行业用户	<b>推送服务</b> 让应用将最新信息即时通知用户,提升F 活跃度
	4		

#### Step 4: Create an Application

1. Click **Apply for Push Services** to enter the application submission page.

HONOR					
Ô	开放能力 / 推送服务				
දේ	▶ 推送服务				
<u>()</u>	推送服务列表				
	应用名称	应用类型	申请时间		
	极光测试demo	移动应用	2022–05–18 12:02:58		
	推送Dev	移动应用	2022-05-12 17:29:39		
	推送Demo	移动应用	2022-03-29 11:09:27		
	•				

2. Select the Application Type "Mobile Application", fill in the Application Package Name and Certificate Fingerprint, agree to the Push Service Agreement and Data Processing Appendix, then click **Submit**.

#### Note:

You must add the packaged SHA256 Certificate Fingerprint, which should match the SHA256 Certificate Fingerprint of your own packaging certificate.

HONOR						
٢		开放能力 / 推送服务 / 申请推送服务				
ିବ	矣 ■ 申请推送服务					
<u></u>		* 应用类型:	● 移动应用 ◎ 服务器应用			
		* 应用名称:	请输入或者选择应用名称(限64字符) ~			
		* 应用包名:	应用包名应为4-64字符	0/64		
		* SHA256证书指纹1:	请输入指纹证书			
	•	SHA256证书指纹2:	请输入指纹证书			
		SHA256证书指纹3:	请输入指纹证书			
		SHA256证书指纹4:	请输入指纹证书			
		SHA256证书指纹5:	请输入指纹证书			
			我已经阅读并同意《荣耀推送服务使用协议》			
			我已经阅读并同意《荣耀开发者服务数据处理附录》			
			取消 提交			

# Step 5: Obtain Application Information

In the **Push Service** list, click **View** to obtain application information.

нон	HONOR				
ූ	开放能力 / 推送服务 【 <b>推送服务</b>				
<u></u>	推送服务列表				
	应用名称	应用类型	申请时间		
	极光测试demo	移动应用	2022-05-18 12:02:58		
	推送Dev	移动应用	2022-05-12 17:29:39		
	推送Demo	移动应用	2022-03-29 11:09:27		
	,				

# Step 6: Add Push Certificate

lor Pus	h Platform	Configuring in the IM cons
нс	NOR	Note: Specified in-app page li dispatching event listening configuration of in-app page
٢		
ංදී	开放能力 / 推送服务 / 查看推送服务	
47		Add Honor certificate
_	应用 <u>类型</u> :移动应用	Package Name • Enter F
$\otimes$	应用名称:腾讯云通信IM	AppID • Enter A
		ClientID • Enter t
		ClientSecret • Enter ti
	APP Secret:	ChannelID Enter a
	Client ID:	Badge Parameter Please
	Client Secret:	"Note: It
	Android端SDK: 《点击下载荣耀PUSH Android端SDK》	Response after Oper
	Android 講接入文档: 《点击下载荣耀PUSH Android 端接入文档》	Unity
	服务靖接入文档:《点击下载荣耀PUSH服务靖接入文档》	select Open app

Google FCM is currently under development, please stay tuned~~

Before integrating the TIMPush component, you need to first apply for an APNs push certificate from Apple, then upload the push certificate to the IM Console. Afterwards, you can follow the Quick Integration steps to integrate. There are currently two mainstream types of certificates for Apple, the p12 certificate and the p8 certificate. Each type of certificate has its advantages and disadvantages, and you can choose one of them according to your needs.

	Certificate Type	Validity Period and Management	Security	Dynamic Island
P12 Certificate	A P12 certificate is a binary file that contains both a public and a private key, used for certificate-based authentication. It	P12 certificates usually have a validity period of one year, after which they need to be regenerated and deployed. Each application requires a	Certificate: The P12 certificate uses certificate- based authentication, requiring the private key to be stored on the server. This may increase the security risk, as the	Not supported

Chat

	bundles the public key certificate and the private key into one file, with an extension of .p12 or .pfx.	separate P12 certificate to handle push notifications.	private key could be accessed by unauthorized users.	
P8 Certificate	A P8 certificate is an Authorization Key, used for token-based authentication. It is a text file containing a private key, with an extension of .p8.	P8 certificates do not have an expiration date, so you do not need to worry about certificate expiration. Moreover, using a P8 certificate can simplify certificate management, as you can use one P8 certificate to provide push notification services for multiple applications.	P8 certificates use token- based authentication, meaning your server periodically generates a JSON Web Token (JWT) to establish a connection with APNs. This method is more secure, as it does not require storing a private key on the server.	Supports Dynamic Island Push

# 1. Using a p12 certificate (traditional push certificate)

# Step 1: Apply for an APNs certificate

# Enable remote push for the app

1. log in to Apple Developer Center website, click Certificates, Identifiers & Profiles or the sidebar's Certificates, IDs & Profiles, enter the Certificates, IDS & Profiles page.



# É Developer

# **Certificates, Identifiers & Profiles**

Certificates	Identifiers 😌	
Identifiers	NAME ~	IDENTIFIER
Devices		
Profiles		
Keys		
More		
	the second s	
	Copyright © 2020 Apple Inc. All right	ts reserved. Terms of Use Privacy Policy

3. You can follow the steps below to create a new AppID or add a Push Notification Service to your existing AppID.

#### Note:

Your App's Bundle ID cannot use the wildcard \*, otherwise, the remote push service cannot be used. 4. Check the **App IDs** box, click **Continue** to proceed to the next step.

#### **É** Developer

# **Certificates, Identifiers & Profiles**

#### < All Identifiers

# **Register a new identifier**

#### App IDs

Register an App ID to enable your app, app extensions, or App Clip to access available services and identify your app in a provisioning profile. You can enable app services when you create an App ID or modify these settings later.

#### O Services IDs

For each website that uses Sign in with Apple, register a services identifier (Services ID), configure your domain and return URL, and create an associated private key.

#### O Pass Type IDs

Register a pass type identifier (Pass Type ID) for each kind of pass you create (i.e. gift cards). Registering your Pass Type IDs lets you generate Apple-issued certificates which are used to digitally sign and send updates to your passes, and allow your passes to be recognized by Wallet.

#### O Website Push IDs

Register a Website Push Identifier (Website Push ID). Registering your Website Push IDs lets you generate Apple-issued certificates which are used to digitally sign and send push notifications from your website to macOS.

#### O iCloud Containers

Registering your iCloud Container lets you use the iCloud Storage APIs to enable your apps to store data and documents in iCloud, keeping your apps up to date automatically.

#### O App Groups

Registering your App Group allows access to group containers that are shared among multiple related apps, and allows certain additional interprocess communication between the apps.

O Merchant IDs

5. Select **App**, click **Continue** to proceed to the next step.

6. Configure the Bundle ID and other information, click **Continue** to proceed to the next step.

ertifica	tes. Identifiers & Profil	es	
All Identifiers			
Register	an App ID		Back Con
Platform iOS, macOS, tvC	S, watchOS	App ID Prefix	
Description		Bundle ID • Explicit	
IMSDK Demo		com.imsdk.pushdemo	
You cannot use :	special characters such as @, &, *, ', ", -, .	We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	
ENABLED	NAME		
	🔍 Access WiFi Information 🕕		
	App Attest		
	Here and the second sec		
	단원 App Groups ①		
	Image: Comparison of the comparison		
	Image: Comparis		

7. Check the **Push Notifications** box to enable the remote push service.

< All Identifiers				
Regist	er an App ID	Back Continu		
	Network Extensions			
	N) NFC Tag Reading			
	VPN Personal VPN 👔			
	Push Notifications (j)			
	Sign In with Apple () Configure			
	SiriKit			
	System Extension			
	OUSer Management			
	Wallet ①			
	Wireless Accessory Configuration			
	Mac Catalyst (Existing Apps Only) (i) Configure			

# **Certificate Generation**

1. Select your AppID and choose **Configure**.

< All Ident	tifiers	
Edity	your App ID Configuration	Remove Save
	Network Extensions (i)	
	N) NFC Tag Reading	
	VPN Personal VPN	
	Push Notifications	Configure Certificates (0)
	Sign In with Apple	Configure
	SiriKit 🕦	
	System Extension	
	User Management	
	Wallet	
	Wireless Accessory Configuration ①	
	Mac Catalyst (Existing Apps Only) (i)	Configure

2. In the **Apple Push Notification service SSL Certificates** window, there are two SSL Certificates for the development environment (Development) and the production environment (Production), as shown below:
| É Developer                | Apple Push Notification service SSL Certificates                                                                                                                                                                                                                                      | Yardiac M<br>Bao Ma - 304735703 |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| Certificat                 | To configure push notifications for this App ID, a Client SSL Certificate that allows your notification server to<br>connect to the Apple Push Notification Service is required. Each App ID requires its own Client SSL Certificate.<br>Manage and generate your certificates below. |                                 |
| < All Identifiers          | Development SSL Certificate                                                                                                                                                                                                                                                           |                                 |
| Edit your App              | Create an additional certificate to use for this App ID.                                                                                                                                                                                                                              | Remove                          |
| Platform                   | Create Certificate                                                                                                                                                                                                                                                                    |                                 |
| iOS, macOS, tvOS, watchO   | Production SSL Certificate                                                                                                                                                                                                                                                            |                                 |
| TPNS SDK demo              | Create an additional certificate to use for this App ID.                                                                                                                                                                                                                              |                                 |
| You cannot use special cha | Create Certificate                                                                                                                                                                                                                                                                    |                                 |
| Capabilities               |                                                                                                                                                                                                                                                                                       |                                 |
| ENABLED NAME               | Done                                                                                                                                                                                                                                                                                  |                                 |
|                            | cess WiFi Information ①                                                                                                                                                                                                                                                               |                                 |
| □ (∑) Ap                   | p Attest (i)                                                                                                                                                                                                                                                                          |                                 |
| D GĐ Ap                    | p Groups (i) Configure                                                                                                                                                                                                                                                                |                                 |
| - 💼 Ap                     | ple Pay Payment Processing (1) Configure                                                                                                                                                                                                                                              |                                 |
|                            | sociated Domains (i)                                                                                                                                                                                                                                                                  |                                 |

# 3.

# We

first select the **Create Certificate** for the Development environment, the system will prompt us that we need a Certificate Signing Request (CSR).

< All Cert	te a New Certificate Back Continu
Certific Apple Pu	a <b>te Type</b> sh Notification service SSL (Sandbox)
Platform	
iOS	
iOS	✓

4. On a Mac, open Keychain Access tool, in the menu select Keychain Access > Certificate Assistant >

Request a Certificate From a Certificate Authority ( Keychain Access - Certificate Assistant -

Request a Certificate From a Certificate Authority ).

Keychain Access F	le Edit	View Window Help		
About Keychain Access	5			
Settings	ж,			
Certificate Assistant	>	Open		
Ticket Viewer	<b>ご</b>	Create a Certificate		
Services	>	Create a Certificate Authority Create a Certificate For Someone Else as a Certificate Authority		
Hide Keychain Access	ж н	Request a Certificate From a Certificate Authority		
Hide Others	ъжн	Set the default Certificate Authority		
Show All		Evaluate a Certificate		
Quit Keychain Access	жQ	Keychain Access		

5. Enter your email address, Common Name (your name or company name), select **Save to disk**, click **continue**, the system will generate a \*.certSigningRequest file.

Certificate Assistant
Certificate Information
Enter information for the certificate you are requesting. Click Continue to request a certificate from the CA.
User Email Address: youremail@example.com Common Name: IMSDK CA Email Address: Required Request is: Emailed to the CA Saved to disk Let me specify key pair information
Continue



6. Go back to the page on the Apple Developer website mentioned in Step 3, click Choose File to upload the

generated \*.certSigningRequest file.

< All Certificates	
Create a New Certificate	Back Continue
<b>Certificate Type</b> Apple Push Notification service SSL (Sandbox)	
Platform:	
iOS	v
<b>Upload a Certificate Signing Request</b> To manually generate a Certificate, you need a Cert Learn more >	ificate Signing Request (CSR) file from your Mac.

7. Click **Continue** to generate the push certificate.

	•	FIUIIES
All Certificates		
Create a Ne	w Certificate	Back Continue
Certificate Type Apple Push Notificatior	n service SSL (Sandbox)	
Platform:		
iOS		~

8. click **Download** to download the Development SSL Certificate to your local environment.

< All Certificates		
Download Your Certi	ficate	
Certificate Details		
Certificate Name com.tpnssdk.pushdemo	Certificate Type APNs Development iOS	Download your certificate to your Mac, then double of Keychain Access. Make sure to save a backup copy of
Expiration Date	Created By	somewnere secure.

Note

The certificate for the production environment is actually a combined certificate of Development (Sandbox) + Production, and it can be used as a certificate for both the development and production environments.

eveloper	
ertificates, Identifie	ers & Profiles
< All Certificates	
Create a New Certificate	
<b>Certificate Type</b> Apple Push Notification service SSL (Sandbox & P	Production)
Platform:	
iOS	~
Choose File	CertificateSigningRequest.certSigningRequest
Choose File	CertificateSigningRequest.certSigningRequest
Choose File	CertificateSigningRequest.certSigningRequest
Choose File Developer <b>ertificates, Identif</b>	CertificateSigningRequest.certSigningRequest
Choose File Developer <b>ertificates, Identif</b>	CertificateSigningRequest.certSigningRequest
Choose File Developer <b>ertificates, Identif</b> Certificates	CertificateSigningRequest.certSigningRequest

# **Certificate Details**

Certificate Name com.tpnssdk.pushdemo Certificate Type **Apple Push Services** 

Download your certificate to your Mac, then do Keychain Access. Make sure to save a backup c somewhere secure.

**Expiration Date** 2021/10/20

**Created By** 

10. Double-click the downloaded SSL Certificate for the development and production environments. The system will import it into the keychain.



Development IOS Push Services and Apple Push Services for the development and production environments as p12 files respectively.



# Note

Set a password when saving the .p12 file.

# Step 2: Upload the certificate to the console

- 1. Log in to the Chat Console.
- 2. Click Plugin Service-Push-Access Settings to enter the access settings page

Chat	Access settings	Current data center: Singapore ()	Telegram group WhatsApp group			
E‡ Application	① The push plug-in trial is about to expire an	d push services (including offline push of ordinary mess	ages, push to all members/labels, etc.) will be autor	natically stopped.To avoid affecting the normal u	se of your business, please Purchase	
음 Users	Push Overview free trial Remaining 3 days	All members/tag push interface call frequency100 ti	mes/day Edit			
ය could Groups	Normal push     Activated	Eunction Preview O     All members/tag push     Activated	Function Preview O           Quick integration           O Activated	Function Preview O Push Records Activated	Function Preview O Push data Activated	© P 6
<ul> <li>Webhook</li> <li>Statistics</li> </ul>	Purchase Now free trial					
Plugin C Push Access settings Access test Push Record Push Record Push Data Push Troubleshooting C Al Chatbot Desk Tools	1 Manufacturer configuration If supports online and offline push notific more stable system-level long connection Android iOS Add Contificate	ations. Online push is delivered through the instant me , and the resource consumption is greatly reduced.	ssaging IM channel, which is fast and reliable; for of	line push, it is recommended that you use the sy	ystem-level push channel provided by each	manufacturer
⑦ Monitor	2 Local deployment					
<u> </u>	After purchasing the push plug-in, you ca	n use the "quick configuration" and "download configu	ration file" functions. With 3 lines of code and 2 minu	ites, you can complete push access from multipl	le manufacturers at once.	
Ξ	Quick configuration	Manua	I configuration			

- 3. Click Add Certificate at the bottom of iOS in Vendor Configuration.
- 4. Select the certificate type, upload the iOS Certificate (.p12), set the certificate password, and click Confirm.

Push Overview free trial Remaining 3 days	All members/tag push interface call t	requency100 times/day Edit		
Normal push     Activated	All Add iOS Certificate		×	Push da
	Push Type	O APNs Push O VolP Push		
Purchase Now free trial	Certificate Type	O Production environment Development Environment		
		Please check that the uploaded certificate type is Apple Push Notification service SSL (Sandbox & Production), and test it after the Archive releases the Release package. Note: It cannot be tested in Xcode.		
1 Manufacturer configuration	Configuration type	<b>O</b> p12		
IM supports online and offline push notific more stable system-level long connection	ation iOS certificate (.p12) •	Select file		-level push cha
Android iOS		How to generate an APNs certificate?		
Add Certificate	mutable-content			
	Certificate password *	Enter the certificate password		
	E	Confirm		

## Note:

We recommend naming the uploaded certificate in English (special characters such as brackets are not allowed).

You need to set a password for the uploaded certificate. Without a password, push notifications cannot be received. For an app published on App Store, the environment of the certificate must be the production environment. Otherwise, push notifications cannot be received.

The uploaded .p12 certificate must be your own authentic and valid certificate.

5. After the pending certificate information is generated, record the certificate's ID.



# 2. Using a p8 certificate (supports Dynamic Island push notifications)

p8 Certificate: A p8 certificate does not have an expiration date, so you don't have to worry about the certificate expiring. Moreover, using a p8 certificate can simplify certificate management, as you can use a single p8 certificate to provide push notification services for multiple applications. In addition, p8 certificates support Dynamic Island push notifications.

# Step 1: Apply for an APNs certificate

To create a p8 certificate file, first log in to Apple Developer Center.

	🗎 developer.apple.com 🖒
	Certificates, Identifiers & Profiles - Apple Developer
🗯 Developer	
Certificate	es, Identifiers & Profiles
Certificates	Keys
Devices Profiles	Getting Started with Keys
Keys More	Creating a key allows you to configure, authenticate, and use one or more Apple services for that key. Unlike certificates, keys do not expire and can be modified to access more services after their creation. For more information on creating and using keys, refer to Developer Account Help.
	Create a key

1. Enter Certificates, Identifiers & Profiles: In the top right corner of the page, click **Account**, then select **Certificates**, **Identifiers & Profiles** from the dropdown menu.

2. To create a new App ID: in the left-hand menu, click Identifiers, then click the + on the right to create a new App

ID. Fill in the relevant information and click Continue.

3. To create a new key: in the left-hand menu, click **Keys**, then click the **+** on the right to create a new key. Enter the name of the key, then check **Apple Push Notifications service (APNs)** and click **Continue**.

< All Key	'S
Download Your Key	
4	After downloading your key, it can <mark>no</mark> t be re-downloaded as the server copy is removed. If you are not prepared to download your key at this time, click Done and download it at a later time. Be sure to save a backup of your key in a secure place.
Name:	ber
Key ID:	

Confirm and generate the key: On the confirmation page, verify your key information, then click **Register**. Next, you'll see a page prompting you to download the key. Click **Download** and save the generated .p8 file to your computer.

## Note:

The p8 certificate can only be downloaded once; please save it properly.

Please safeguard the downloaded p8 file, as you will not be able to download it again. You can use this p8 certificate to configure your iOS applications to receive push notifications.

# Step 2: Upload the p8 certificate to the IM console

- 1. Log in to the Chat Console.
- 2. Click Plugin Service-Push-Access Settings to enter the access settings page

Chat	Access settings Current data center: Singapore () Telegram group WhatsApp group
∃∄ Application	
Management	Ine push plug-in that is about to expire and push services (including omline push of ordinary messages, push to all memoers/labels, etc.) will be automatically stopped. to avoid affecting the norm
H Overview	Pueb Quantieur fas triel Pempining 2 daus All members/tag pueb interface call fragmence/100 times/day. Edit
은 Users	
퉒 Groups	▶ Normal push     ▲ All members/tag push       ♥ Quick integration
Configuration ·	Activated     Activated     Activated     Activated     Activated
🕸 Webhook	Purchase Now free trial
C Statistics	
Plugin	A Manufacturer configuration
🕞 Push 🔷	IM supports online and offline push notifications. Online push is delivered through the instant messaging IM channel, which is fast and reliable; for offline push, it is recommended that you use t
Access settings	more stable system-level long connection, and the resource consumption is greatly reduced.
Access test	Android iOS
Push Record	Add Certificate
Push Data	
<ul> <li>Push Troubleshooting</li> </ul>	
O Al Chatbot	
Desk	No certificate yet
Tools	
⑦ Monitor	
🛅 Dev Tools 🔹 🗸	2 Local deployment After purchasing the push plun-in, you can use the "quick configuration" and "download configuration file" functions. With 3 lines of code and 2 minutes, you can complete push access from m
	Quick configuration         Manual configuration

- 3. Click Add Certificate at the bottom of iOS in Vendor Configuration.
- 4. Select the .p8 certificate

Push Type	O APNs Push VolP Push
Certificate Type	<ul> <li>Production environment</li> <li>Development Environment</li> <li>Please check that the uploaded certificate type is Apple Push Notification s</li> <li>Production), and test it after the Archive releases the Release package. Not</li> </ul>
Configuration type	○ p12 ○ p8
iOS Certificate (.p8) *	Select file
	How to generate an APNs certificate?
mutable-content (	
mutable-content 🛈 KeyID *	Enter
mutable-content (j) KeyID * TeamID *	Enter Enter

# Note:

**Key ID**: This is the unique identifier for your APNs Auth Key. When you create a new APNs Auth Key in the Apple Developer Center, a Key ID will be generated for you. You can find it in the "Certificates, Identifiers & Profiles" section under "Keys".

**Team ID**: This is the unique identifier for your developer account. You can find it on the account details page of the Apple Developer Center. Click "Membership" in the upper right corner, and you can find your Team ID in the "Membership Details" section.

**Bundle ID**: This is the unique identifier for your application, also known as the app ID. You can find it in the "Certificates, Identifiers & Profiles" section of the Apple Developer Center. Select "Identifiers", then find the corresponding Bundle ID in your list of applications.



# Flutter

Last updated : 2024-06-13 11:00:50

The current messaging push plugin, when utilized within Flutter, exclusively supports dispatching notifications to Android devices (including channels from various manufacturers) and iOS devices.

iOS

# Android

Before integrating the TIMPush component, you need to apply for an APNs Push Certificate from Apple and then upload the Push Certificate to the IM console. After that, you can proceed with the quick access steps.

There are currently two mainstream types of certificates for Apple Manufacturer Configuration: p12 certificates and p8 certificates. Each type of certificate has its advantages and drawbacks, and you can choose one according to your needs.

# **Certificate Type:**

p12 Certificate: A p12 certificate is a binary file containing both a public key and a private key, used for certificatebased authentication. It bundles the public key certificate and the private key into one file, with extensions .p12 or .pfx. p8 Certificate: A p8 certificate is an Auth Key, used for token-based authentication. It is a text file containing a private key, with an extension of .p8.

# Validity and Management:

p12 Certificate: A p12 certificate typically has a one-year validity period, after which it needs to be regenerated and deployed. Each application requires a separate P12 certificate to handle push notifications.

p8 Certificate: A p8 certificate does not have an expiration date, so you don't have to worry about the certificate expiring. Moreover, using a P8 certificate can simplify certificate management, as you can use a single p8 certificate to provide push notification services for multiple applications.

# Security:

p12 Certificate: A p12 certificate uses certificate-based authentication and requires the private key to be stored on the server. This could increase security risks, as the private key could be accessed by unauthorized users.

p8 Certificate: A p8 certificate uses token-based authentication, which means your server periodically generates a JSON Web Token (JWT) to establish a connection with APNs. This method is more secure, as it doesn't require storing a private key on the server.

# **Dynamic Island:**

p12 Certificate: Not supported.

p8 Certificate: Supports Dynamic Island push notifications.

# 1. Using a p12 certificate (traditional push certificate)

# Step 1: Apply for an APNs certificate

# Enable remote push for the app

1. log in to Apple Developer Center website, click Certificates, Identifiers & Profiles or the sidebar's Certificates, IDs & Profiles, enter the Certificates, IDS & Profiles page.



# É Developer

# **Certificates, Identifiers & Profiles**

Certificates	Identifiers 😌	
Identifiers	NAME ~	IDENTIFIER
Devices		
Profiles		
Keys		
More		
	the second s	
	Copyright © 2020 Apple Inc. All right	ts reserved. Terms of Use Privacy Policy

3. You can follow the steps below to create a new AppID or add a Push Notification Service to your existing AppID.

# Note:

Your App's Bundle ID cannot use the wildcard \*, otherwise, the remote push service cannot be used. 4. Check the **App IDs** box, click **Continue** to proceed to the next step.

# Ceveloper

# **Certificates, Identifiers & Profiles**

## < All Identifiers

# **Register a new identifier**

### $\bigcirc$ App IDs

Register an App ID to enable your app, app extensions, or App Clip to access available services and identify your app in a provisioning profile. You can enable app services when you create an App ID or modify these settings later.

#### $\bigcirc$ Services IDs

For each website that uses Sign in with Apple, register a services identifier (Services ID), configure your domain and return URL, and create an associated private key.

#### $\bigcirc$ Pass Type IDs

Register a pass type identifier (Pass Type ID) for each kind of pass you create (i.e. gift cards). Registering your Pass Type IDs lets you generate Apple-issued certificates which are used to digitally sign and send updates to your passes, and allow your passes to be recognized by Wallet.

### $\bigcirc$ Website Push IDs

Register a Website Push Identifier (Website Push ID). Registering your Website Push IDs lets you generate Apple-issued certificates which are used to digitally sign and send push notifications from your website to macOS.

#### $\bigcirc$ iCloud Containers

Registering your iCloud Container lets you use the iCloud Storage APIs to enable your apps to store data and documents in iCloud, keeping your apps up to date automatically.

### App Groups

Registering your App Group allows access to group containers that are shared among multiple related apps, and allows certain additional interprocess communication between the apps.

5. Select **App**, click **Continue** to proceed to the next step.

			refiles	
ertifica	ates, Identif	iers & P	rofiles	
< All Identifiers				
Registe	r a new identifie	r		Back
Select a t	type			
		(=	->	
	Å	í (	Ĵ	
	Арр	Арр	Clip	
the Devision	le TD and other in	formation alia	K Continue to proceed to the next sta	n
			·	
É Doveloper				
Developer				
	tes, Identifiers {	& Profiles		
Certificat < All Identifiers Register a	t <b>es, Identifiers &amp;</b> an App ID	& Profiles		Back
Certificat < All Identifiers Register a Platform	t <b>es, Identifiers &amp;</b> an App ID	& Profiles	App ID Prefix	Back
Certificat < All Identifiers Register a Platform iOS, macOS, tvOS	t <b>es, Identifiers &amp;</b> an App ID	& Profiles	App ID Prefix	Back
Certificat < All Identifiers Register a Platform iOS, macOS, tvOS Description IMSDK Demo	t <b>es, Identifiers &amp;</b> an App ID	& Profiles	App ID Prefix Bundle ID • Explicit • Wildcard com.imsdk.pushdemo	Back
Certificat < All Identifiers Register a Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp	tes, Identifiers ک an App ID ;, watchOS	& Profiles	App ID Prefix Bundle ID • Explicit • Wildcard com.imsdk.pushdemo We recommend using a reverse-domain name style string (i.e.,	Back
Certificat < All Identifiers Register : Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp	tes, Identifiers & an App ID ;, watchOS	& Profiles	App ID Prefix Bundle ID  Explicit Wildcard Com.imsdk.pushdemo We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat < All Identifiers Register : Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities	tes, Identifiers & an App ID ;, watchOS pecial characters such as @, &, *, *, *, -, . App Services	& Profiles	App ID Prefix Bundle ID  Explicit Wildcard Com.imsdk.pushdemo We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat < All Identifiers Register : Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities ENABLED	tes, Identifiers & an App ID 3, watchOS Decial characters such as @, &, *, *, *, -, - App Services	& Profiles	App ID Prefix Bundle ID • Explicit • Wildcard com.imsdk.pushdemo We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat < All Identifiers Register : Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities ENABLED	tes, Identifiers & an App ID 3, watchOS Decial characters such as @, &, *, ', ", -, . App Services NAME © Access WiFi Informat	& Profiles	App ID Prefix         Bundle ID • Explicit • Wildcard         com.imsdk.pushdemo         We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat < All Identifiers Register : Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities ENABLED 	tes, Identifiers & an App ID 3, watchOS Decial characters such as @, &, *, ', ", -, App Services NAME © Access WiFi Informat	& Profiles	App ID Prefix         Bundle ID • Explicit       Wildcard         com.imsdk.pushdemo         We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat < All Identifiers Register : Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities ENABLED 	tes, Identifiers &         an App ID         3, watchOS         becial characters such as @, &, *, ', ", -,         App Services         NAME <ul> <li>Access WiFi Informat</li> <li>App Attest ①</li> <li>App Groups ①</li> </ul>	& Profiles	App ID Prefix         Bundle ID • Explicit       Wildcard         com.imsdk.pushdemo         We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat  All Identifiers  Register a  Platform iOS, macOS, tvOS  Description IMSDK Demo You cannot use sp  Capabilities ENABLED	tes, Identifiers &         an App ID         3, watchOS         becial characters such as @, &, *, ', ", -, .         App Services         NAME         Image: Comparison of the services         App Groups Image: Comparison of the services         Image: Comparison of the services	& Profiles	App ID Prefix         Bundle ID • Explicit       Wildcard         com.imsdk.pushdemo         We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat  All Identifiers  Register :  Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities ENABLED	an App ID   3, watchOS   Decial characters such as @, &, *, ', ', -, .   App Services   NAME <ul> <li>Access WiFi Informat</li> <li>App Attest</li> <li>App Groups</li> <li>Apple Pay Payment P</li> <li>Associated Domains</li> </ul>	& Profiles	App ID Prefix         Bundle ID • Explicit       Wildcard         com.imsdk.pushdemo         We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat  All Identifiers  Register a  Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities ENABLED	an App ID   3, watchOS   becial characters such as @, &, *, ', ", -, .   App Services   NAME   Image: Services	& Profiles	App ID Prefix         Bundle ID • Explicit       Wildcard         corn.imsdk.pushdemo         We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back

7. Check the **Push Notifications** box to enable the remote push service.

< All Identifier	S	
Registe	er an App ID	Back Continu
	<b>√→</b> Multipath (j)	
	Network Extensions	
	N) NFC Tag Reading 🕕	
	VPN Personal VPN	
	Push Notifications	
	Sign In with Apple (i)     Configure	
	SiriKit	
	System Extension	
	OUSer Management	
	Wallet (1)	
	Wireless Accessory Configuration	
	Mac Catalyst (Existing Apps Only) (1)	

# **Certificate Generation**

1. Select your AppID and choose **Configure**.

< All Identifie	rs	
Edit yo	our App ID Configuration	Remove Save
	Network Extensions 🕦	
	N) NFC Tag Reading	
	VPN Personal VPN	
	Push Notifications (i)	Configure Certificates (0)
	Sign In with Apple 👔	Configure
	SiriKit 👔	
	System Extension	
	OUser Management	
	Wallet	
	S Wireless Accessory Configuration	
	Mac Catalyst (Existing Apps Only) (j)	Configure

2. In the **Apple Push Notification service SSL Certificates** window, there are two SSL Certificates for the development environment (Development) and the production environment (Production), as shown below:

É Developer	Apple Push Notification service SSL Certificates	Yardian N San Mu - 954735703
Certificat	To configure push notifications for this App ID, a Client SSL Certificate that allows your notification server to connect to the Apple Push Notification Service is required. Each App ID requires its own Client SSL Certificate. Manage and generate your certificates below.	
< All Identifiers	Development SSL Certificate	
Edit your App	Create an additional certificate to use for this App ID.	Remove
Platform	Create Certificate	
iOS, macOS, tvOS, watchO Description	Production SSL Certificate	
TPNS SDK demo	Create an additional certificate to use for this App ID.	
You cannot use special cha	Create Certificate	
Capabilities		
ENABLED NAME	Done	
	cess WiFi Information (1)	
- (2) Ap	p Attest (i)	
[	p Groups (i) Configure	
- 🖻 Ap	ple Pay Payment Processing (i) Configure	
As	sociated Domains ①	
<u> </u>		

# 3. We

first select the **Create Certificate** for the Development environment, the system will prompt us that we need a Certificate Signing Request (CSR).

< All Certificates
Create a New Certificate Back Conti
<b>Certificate Type</b> Apple Push Notification service SSL (Sandbox)
Diatform
iOS v
Upload a Certificate Signing Request To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Ma Learn more > Choose File

Request a Certificate From a Certificate Authority ).

Keychain Access File	Edit	View Window Help
About Keychain Access		
Settings	¥,	
Certificate Assistant	>	Open
Ticket Viewer	ΖЖК	Create a Certificate
Services	>	Create a Certificate Authority Create a Certificate For Someone Else as a Certificate Authority
Hide Keychain Access	ж н	Request a Certificate From a Certificate Authority
Hide Others	ъжн	Set the default Certificate Authority
Show All		Evaluate a Certificate
Quit Keychain Access	жQ	Kevchain Access

5. Enter your email address, Common Name (your name or company name), select **Save to disk**, click **continue**, the system will generate a \*.certSigningRequest file.

	Certificate Assistant
	Certificate Information
	Enter information for the certificate you are requesting. Click Continue to request a certificate from the CA.
Certo	User Email Address: youre mail@example.com   Common Name: IMSDK   CA Email Address: Required   Request is: Image: Emailed to the CA   Saved to disk   Let me specify key pair information

Continue

6. Go back to the page on the Apple Developer website mentioned in Step 3, click **Choose File** to upload the generated \*.certSigningRequest file.

С	ertificates Identifiers & Profiles
Ŭ	er moutes, raentmers ar romes
	< All Certificates
	Create a New Certificate Back Continue
	Certificate Type Apple Push Notification service SSL (Sandbox)
	Platform:
	iOS
	Upload a Certificate Signing Request To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac. Learn more > Choose File

< All Certificates
Create a New Certificate Back Continue
<b>Certificate Type</b> Apple Push Notification service SSL (Sandbox)
Platform:
iOS
Upload a Certificate Signing Request To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac. Learn more >

É Developer		
Certificates, Ide	entifiers & Profiles	
< All Certificates Download Your Certif	ficate	
Certificate Details		
Certificate Name com.tpnssdk.pushdemo	Certificate Type APNs Development iOS	Download your certificate to your Mac, then double clic Keychain Access. Make sure to save a backup copy of
Expiration Date 2021/09/20	Created By	somewnere secure.

9. Repeat steps 1 - 8 above to download the Production SSL Certificate for the production environment to

your local machine.

Note:

The certificate for the production environment is actually a combined certificate of Development (Sandbox) +

Production, and it can be used as a certificate for both the development and production environments.

## É Developer

<ul> <li>All Certificates</li> </ul>		
Create a New Certi	ficate	
Certificate Type Apple Push Notification service SSL (	Sandbox & Production)	
Platform:		
iOS	•	
Choose File	CertificateSigningRequest.certSigningRequ	est
Development		
Developer		
Certificates, Id	entifiers & Profiles	
Developer	entifiers & Profiles	
Developer Certificates, Id	entifiers & Profiles	
Developer Certificates, Id All Certificates Download Your Certi	entifiers & Profiles	
Developer Certificates, Id All Certificates Download Your Certi Certificate Details	entifiers & Profiles ficate	
Developer     Certificates, Id     All Certificates     Download Your Certi     Certificate Details     rertificate Name     om.tpnssdk.pushdemo	entifiers & Profiles ficate Certificate Type Apple Push Services	Download your certificate to your Mac, then Keychain Access. Make sure to save a backu

system will import it into the keychain.

11. Open the Keychain App, go to log in to > My Certificates, right-click to export the newly created Apple Development IOS Push Services and Apple Push Services for the development and production environments as p12 files respectively.

©2013-2022 Tencent Cloud. All rights reserved.

	Keychain Access						λ Search	)
Default Keychains	All Items Passwords Secure	Notes My Certificate	s Keys Certificates					
🗗 login	Angle Duck Con							
Local Items	Certificate Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Conduct Co	orldwide Developer Re December 5, 2024 at 1 s valid	lations Certification Authority 15:52:39 China Standard Time	e <b>rvice</b> / e				
🔒 System								
System Roots	Name			~	Kind	Date Modified		Expir
	Apple Persistent State End	ryption			application password	Yesterday, 15:09		
	Apple Push Services: com	).xa	1		certificate			Oct 2
	Apple Push Services: com	xa	1		certificate			Dec 8
	🔀 Apple Push Services: com	.xa	i.db		certificate	~ <del>~-</del>	976	Mar 2
	Apple Push Services: com	d Nev	v Identity Preference					Dec 5
	apple Worldwide Develope	r Relations Cert	and a second sec					Feb 2
	Apple Worldwide Develope	r Relations Cerl Cop	y "Apple Push Services: c	com.	p	oushservice"		Mar 1
	Apple Worldwide Develope	r Relations Cer Delf	ete "Apple Push Services:	com.		pushservice"		Feb 8
	Apple Worldwide Develope	r Relations Ceri	and the second s	- Children			- SNON	Jan 2
	Apple Worldwide Develope	r Relations Cerl Exp	ort "Apple Push Services:	com.t		pushservice"		Dec 1
	AppleIDClientIdentifier						7:54	
	Basic Attestation User Suc	Get Get	Info				.00	Mar 2
	Beaconstore	Eval	luate "Apple Push Service	s: com		pushservice"	.02	

## Note

Set a password when saving the .p12 file.

# Step 2: Upload the certificate to the console

- 1. Log in to the Chat Console.
- 2. Click Plugin Service-Push-Access Settings to enter the access settings page



- 3. Click Add Certificate at the bottom of iOS in Vendor Configuration.
- 4. Select the certificate type, upload the iOS Certificate (.p12), set the certificate password, and click Confirm.

Access settings 2 <sup>r</sup> ,st •		ingapore 🚯 Telegram group WhatsApp group		
① The push plug-in trial is about to expire and push serv				
Push Overview free trial Remaining 3 days All mem				
				_
Normal push     All	Add iOS Certificate		×	Push dat
	Push Type	APNs Push     VolP Push		<ul> <li>Activat</li> </ul>
Purchase Now free trial	Certificate Type	Production environment     Development Environment		
		Please check that the uploaded certificate type is Apple Push Notification service SSL (Sandbox & Production), and test it after the Archive releases the Release package. Note: It cannot be tested in Xcode.		
1 Manufacturer configuration	Configuration type	<b>O</b> p12		
IM supports online and offline push notifications. On	iOS certificate (.p12) *	Select file		-level push char
		How to generate an APNs certificate?		
Android				
Add Certificate	mutable-content 🚯			
	Certificate password *	Enter the certificate password		
		Confirm		
No certific				
2 Local deployment				
After purchasing the push plug-in, you can use the *c	uick configuration" and "do	wnload configuration file" functions. With 3 lines of code and 2 minutes, you can complete push access from m	ultiple ma	nufacturers at on

# Note:

We recommend naming the uploaded certificate in English (special characters such as brackets are not allowed). You need to set a password for the uploaded certificate. Without a password, push notifications cannot be received. For an app published on App Store, the environment of the certificate must be the production environment. Otherwise, push notifications cannot be received.

The uploaded .p12 certificate must be your own authentic and valid certificate.

5. After the pending certificate information is generated, record the certificate's ID.



# Second, using a p8 certificate (supports Dynamic Island push notifications)

p8 Certificate: A p8 certificate does not have an expiration date, so you don't have to worry about the certificate expiring. Moreover, using a p8 certificate can simplify certificate management, as you can use a single p8 certificate to provide push notification services for multiple applications. In addition, p8 certificates support Dynamic Island push notifications.

# Step 1: Apply for an APNs certificate

To create a p8 certificate file, first log in to Apple Developer Center.



1. Enter Certificates, Identifiers & Profiles: In the top right corner of the page, click **Account**, then select **Certificates**, **Identifiers & Profiles** from the dropdown menu.

2. To create a new App ID: in the left-hand menu, click Identifiers, then click the + on the right to create a new App

ID. Fill in the relevant information and click Continue.

3. To create a new key: in the left-hand menu, click **Keys**, then click the **+** on the right to create a new key. Enter the name of the key, then check **Apple Push Notifications service (APNs)** and click **Continue**.



Download Your Key  After downloading your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key in a secure place.	Download Your Key  After downloading your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your key in a secure place.  Name:	Download Your Key  After downloading your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download y time, click Done and download it at a later time. Be sure to save a backup of your key in a secure place.  Name: Key ID:	< All Key	\$
After downloading your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your litime, click Done and download it at a later time. Be sure to save a backup of your key in a secure place.	After downloading your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your lime, click Done and download it at a later time. Be sure to save a backup of your key in a secure place.	After downloading your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download y time, click Done and download it at a later time. Be sure to save a backup of your key in a secure place. Name:	Dow	nload Your Key
	Name:ber	Name: per Kev ID:		After downloading your key, it cannot be re-downloaded as the server copy is removed. If you are not prepared to download your time, click Done and download it at a later time. Be sure to save a backup of your key in a secure place.
	Name: per	Name: Der		time, click Done and download it at a later time. Be sure to save a backup of your key in a secure place.

Confirm and generate the key: On the confirmation page, verify your key information, then click Register. Next, you'll see a page prompting you to download the key. Click **Download** and save the generated .p8 file to your computer.

## Note:

The p8 certificate can only be downloaded once; please save it properly.

Please safeguard the downloaded p8 file, as you will not be able to download it again. You can use this p8 certificate to configure your iOS applications to receive push notifications.

# Step 2: Upload the p8 certificate to the IM console

- 1. Log in to the Chat Console.
- 2. Click Plugin Service-Push-Access Settings to enter the access settings page
- 3. Click Add Certificate at the bottom of iOS in Vendor Configuration.
- 4. Select the .p8 certificate

# **Operation step**

# Step 1. Register your app with vendor push platforms

To utilize the offline push feature, you need to register your app on each vendor's push platform to obtain parameters such as AppID and AppKey. Currently, the mobile manufacturers supported in China include: Mi, Huawei, Honor, OPPO, Vivo, Meizu, and internationally Google FCM is supported.

# Step 2. Create resources in the IM console

Log in to Tencent Cloud Chat Console, then in the Push Management > Access Settings feature section, add each vendor's push certificate, and configure the AppID, AppKey, AppSecret, and other parameters obtained in Step 1 to the added push certificate.

# Explanation of the Subsequent Actions option:

Open Application: Clicking the notification bar launches the app, by default starting the app's Launcher interface;

Open Web Page: Clicking the notification bar will redirect to the configured web link;

Open the specified interface within the app: clicking the notification bar will redirect the interface based on the configured self Definition, see Custom Redirect on Click.

Huawei	
OPPO	
vivo	
Meizu	
HONOR	
Google FCM	
Vendor Push Platform	Configuring in the IM console

vendor Push Plat		Conliguning in the	
		Add Mi certifica	ite
📶 小米开放平台	推送這帶平台 Tuxa • 请前: a	Package Name *	Enter Package Name
☞ 推送工具	TUIKit	AppID *	Enter AppID
2 推送统计	应用类型 Android	AppKey *	Enter AppKey
<ul> <li>• 应用信息</li> </ul>	的建計同	AppSecret *	Enter AppSecret
通知类别 角色管理	主包名	Region	China India
∾ 调查工具	AppRey 124	ChannellD	Enter a channel ID
三 法律文档		Response after Click	Open application *Note: the Mi "onNotific
			using this method.

Vendor Push Platform	Configuring in the IM cons



Vendor Push Platform	Configuring in the IM cons
	Add OPPO certificate
OPPO官网 ColorOS社区 井虹平台 用户中心 OPPO○   推送运营平台 := 10 5.5 tulkit ✓	Package Name * Enter F
	AppKey • Enter /
☆ 创建推进 / 首页 > 应用配置	AppID • Enter /
④ 数据统计 V tuikit	AppSecret *
	MastarSecret *
記置管理   Appld	ChannellD Enter a
检查工具     AppKey cf     查看       新時建遊道     AppSecret 75************************************	Response after Oper Click
通道設置 推送链 MasterSecret 重 重 二 二 二 二 二 二 二 二 二 二 二 二 二 二 二 二 二	

Vendor Push Platform

Configuring in the IM cons

	一件法法律可公	Add vivo certifi	cate
		Package Name *	
38 云週信IM ◇		AppKey *	
	应用名称: 云通信IM	AppID *	
⑦ 应用管理 ^	应用类别:移动应用 推送权限:正式	Receipt ID	
应用信息	审核状态: 已通过	Category	
测试设备	回建时间:	AppSecret *	
⊘ 标签管理	AppiD:	Description	
♀。在线诊断	AppKey:       AppSecret:	Response after Click	

For receipt configuration, please refer to: Message reach statistics configuration - vivo

Vendor Push Platform	Configuring in the IM cons
Flyme 推送平台 首页 创建推送 数据统计 配置管理	
应用配置 标签用户 问题排查 黑名单 回执管理 常用设备 多包名 任务备注	Add Meizu certificate
TUIKit	Package Name   Enter F
应用名称 TUIKit	AppID • Enter A
应用形态 普通应用 应用包名 透加多包名	AppKey • Enter A
应用类型 通讯社交 ~ 应用图标 更换图片 尺寸为480*480,500KB以内	receipt switch After turn configure
	AppSecret • Enter A
<ul> <li>O App ID</li> <li>O App Key</li> <li>O App Secret</li> </ul>	Response after O Oper Click
快速集成 下载代码 扫描下载DemoAPK	

# For receipt configuration, please refer to: Message reach statistics configuration - Meizu

Vendo	or Push	Platform	Configuring in the IM cons
	НОМ		
	1101		
	٢	开放能力 / 推送服务 / 查看推送服务	Add Honor certificate
	ංදි	■ 查看推送服务	Package Name * Enter F
	<u></u>	应用类型: 移动应用	AppID • Enter A
	8	应用名称: 腾讯云通信IM	ClientID * Enter t
		应用包名:	ClientSecret * Enter ti
		SHA256证书指纹1:  由语时间:	ChannellD Enter a
			Badge Parameter Please
		APP Secret:	*Note: It
		Client ID:	Response after Oper
		Client Secret:	Click
		Android講SDK: 《点击下载荣耀PUSH Android講SDK》	
		Android講接入文档: 《点击下载荣耀PUSH Android講接入文档》	
		服务端接入文档:《点击下载荣耀PUSH服务端接入文档》	

Vendor Push Platform	Configuring in the IM cons				
Firebase	1.		6 + 0 B A		
-----------------------	-------------------------------------------------------	-----------------------------------------------------------------------------------------------	-----------	--------------------	---------
A Project Overview	Project settings				ioato
Generative AI	General Cloud Messaging Integrations Service accounts	Data privacy Users and permissions			oute
+ Build with Gemini 💌					
Project shortcuts		Manage service account permissions		Adding Method	aU 🔿
Authentication	Or Firebase Admin SDK	Firebase Admin SDK		, teaming the tree	• - F
Test Lab		Your Firebase service account can be used to authenticate multiple Firebase features, such as		Message type	
CS Messaging	Legacy credemias	Database, Storage and Auth, programmatically via the unified Admin SDK. Learn more 🗹		Message type	
App Check	Database secrets	Firebase service account			Transpa
What's new	All service accounts				availab
App Hosting 💌	S 7 service accounts	Admin SDK configuration snippet			SDK er
Se Data Connect (NW)		Node js      Java      Python      Go			ODITO
Product categories					
uild v		<pre>var admin = require("firebase-admin");</pre>		Package Name *	Enter
		<pre>var serviceAccount = require("path/to/serviceAccountKey.json");</pre>			
un V		admin.initializeApp({ credential: admin.credential.cert(serviceAccount)			
Analytics ~		)); [		Upload certificate	
All products					
		Generate new private key			How to
				ChappellD	Entor
				ChannenD	Enti
				onumone	

#### Note:

Regarding **Click for Subsequent Actions** supports the Report Statistics feature:

1. If you choose to open an app or a web page, purchasing the plugin will by default support reporting statistics.

2. If you choose to open a specified interface within the application:

For new certificate status, please directly use the auto-fill default value to support click statistics reporting.

If there was a certificate configured previously, continue using the old certificate and modify it to the default value to support reporting statistics, or regenerate a new certificate.

#### About FCM Data Messaging

FCM provides two push methods: Notification Message and Data Messaging.

Notification Message, with a simple style not differentiated by device, can support offline push once successfully integrated;

Data Messaging, offering rich customization for specific devices, supports reach and click reporting, and requires testing on the device before going live after integration.

The console defaults to Notification Message, and switching between modes can be done in the IM Console:



#### Note:

FCM Data Messaging capability is only supported on Pixel phones with TIMPush 7.8 and above, other manufacturers' devices need to be self-tested for support;

# **React-Native**

Last updated : 2024-06-24 15:45:32

The current messaging push plugin, when utilized within React Native, exclusively supports dispatching notifications to Android devices (including channels from various manufacturers) and iOS devices.

iOS

#### Android

Before integrating the TIMPush component, you need to apply for an APNs Push Certificate from Apple and then upload the Push Certificate to the IM console. After that, you can proceed with the quick access steps.

There are currently two mainstream types of certificates for Apple Manufacturer Configuration: p12 certificates and p8 certificates. Each type of certificate has its advantages and drawbacks, and you can choose one according to your needs.

#### **Certificate Type:**

p12 Certificate: A p12 certificate is a binary file containing both a public key and a private key, used for certificatebased authentication. It bundles the public key certificate and the private key into one file, with extensions .p12 or .pfx. p8 Certificate: A p8 certificate is an Auth Key, used for token-based authentication. It is a text file containing a private key, with an extension of .p8.

#### Validity and Management:

p12 Certificate: A p12 certificate typically has a one-year validity period, after which it needs to be regenerated and deployed. Each application requires a separate P12 certificate to handle push notifications.

p8 Certificate: A p8 certificate does not have an expiration date, so you don't have to worry about the certificate expiring. Moreover, using a P8 certificate can simplify certificate management, as you can use a single p8 certificate to provide push notification services for multiple applications.

#### Security:

p12 Certificate: A p12 certificate uses certificate-based authentication and requires the private key to be stored on the server. This could increase security risks, as the private key could be accessed by unauthorized users.

p8 Certificate: A p8 certificate uses token-based authentication, which means your server periodically generates a JSON Web Token (JWT) to establish a connection with APNs. This method is more secure, as it doesn't require storing a private key on the server.

#### **Dynamic Island:**

p12 Certificate: Not supported.

p8 Certificate: Supports Dynamic Island push notifications.

# 1. Using a p12 certificate (traditional push certificate)

## Step 1: Apply for an APNs certificate

#### Enable remote push for the app

1. log in to Apple Developer Center website, click Certificates, Identifiers & Profiles or the sidebar's Certificates, IDs & Profiles, enter the Certificates, IDS & Profiles page.



## É Developer

# **Certificates, Identifiers & Profiles**

Certificates	Identifiers 😌	
Identifiers	NAME ~	IDENTIFIER
Devices		
Profiles		
Keys		
More		
	Copyright © 2020 Apple Inc. All r	ights reserved. Terms of Use Privacy Policy

3. You can follow the steps below to create a new AppID or add a Push Notification Service to your existing AppID.

#### Note:

Your App's Bundle ID cannot use the wildcard \*, otherwise, the remote push service cannot be used. 4. Check the **App IDs** box, click **Continue** to proceed to the next step.

#### Ceveloper

## **Certificates, Identifiers & Profiles**

#### < All Identifiers

## **Register a new identifier**

#### $\bigcirc$ App IDs

Register an App ID to enable your app, app extensions, or App Clip to access available services and identify your app in a provisioning profile. You can enable app services when you create an App ID or modify these settings later.

#### $\bigcirc$ Services IDs

For each website that uses Sign in with Apple, register a services identifier (Services ID), configure your domain and return URL, and create an associated private key.

#### $\bigcirc$ Pass Type IDs

Register a pass type identifier (Pass Type ID) for each kind of pass you create (i.e. gift cards). Registering your Pass Type IDs lets you generate Apple-issued certificates which are used to digitally sign and send updates to your passes, and allow your passes to be recognized by Wallet.

#### $\bigcirc$ Website Push IDs

Register a Website Push Identifier (Website Push ID). Registering your Website Push IDs lets you generate Apple-issued certificates which are used to digitally sign and send push notifications from your website to macOS.

#### $\bigcirc$ iCloud Containers

Registering your iCloud Container lets you use the iCloud Storage APIs to enable your apps to store data and documents in iCloud, keeping your apps up to date automatically.

#### App Groups

Registering your App Group allows access to group containers that are shared among multiple related apps, and allows certain additional interprocess communication between the apps.

5. Select **App**, click **Continue** to proceed to the next step.

				Tariblac Mu
Certifica	ates, Identifie	ers & P	rofiles	
< All Identifiers				
Registe	r a new identifier			Back
Select a t	type			
		ſ		
	App	App	 Clip	
Certificat	tes. Identifiers & I	Profiles		
Certificat	tes, Identifiers & I	Profiles		
Certificat	tes, Identifiers & I	Profiles		
Certificat	t <b>es, Identifiers &amp; I</b> an App ID	Profiles		Back
Certificat < All Identifiers Register a Platform iOS, macOS, tyOS	tes, Identifiers & I an App ID	Profiles	App ID Prefix	Back
Certificat < All Identifiers Register a Platform iOS, macOS, tvOS Description	tes, Identifiers & I an App ID	Profiles	App ID Prefix Bundle ID • Explicit • Wildcard	Back
Certificat < All Identifiers Register a Platform iOS, macOS, tvOS Description IMSDK Demo	tes, Identifiers & I an App ID	Profiles	App ID Prefix Bundle ID • Explicit • Wildcard com.imsdk.pushdemo	Back
Certificat < All Identifiers Register a Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp	<b>tes, Identifiers &amp; I</b> an App ID 3, watchOS	Profiles	App ID Prefix Bundle ID  Explicit Wildcard Com.imsdk.pushdemo We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat < All Identifiers Register a Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities	tes, Identifiers & I an App ID S, watchOS pecial characters such as @, &, *, ', ", -, . App Services	Profiles	App ID Prefix Bundle ID  Explicit Wildcard Com.imsdk.pushdemo We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back C
Certificat < All Identifiers Register a Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities ENABLED	tes, Identifiers & I an App ID S, watchOS pecial characters such as @, &, *, ', ", -, . App Services NAME	Profiles	App ID Prefix Bundle ID  Explicit Wildcard Com.imsdk.pushdemo We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back C
Certificat < All Identifiers Register a Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities ENABLED	an App ID   S, watchOS     pecial characters such as @, &, *, ', ", -, .   App Services   NAME   Q   Access WiFi Information	Profiles 	App ID Prefix Bundle ID  Explicit Wildcard Com.imsdk.pushdemo We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat	an App ID   S, watchOS     pecial characters such as @, &, *, ', ", -, .   App Services   NAME <ul> <li>Access WiFi Information</li> <li>App Attest ①</li> </ul>	Profiles 	App ID Prefix         Bundle ID • Explicit       Wildcard         com.imsdk.pushdemo         We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat	an App ID   S, watchOS   Decial characters such as @, &, *, ', ", -,   App Services   NAME <ul> <li>Access WiFi Information</li> <li>App Attest</li> <li>App Groups</li> </ul>	Profiles	App ID Prefix         Bundle ID • Explicit       Wildcard         com.imsdk.pushdemo         We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat All Identifiers Register a Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities ENABLED Image: Specified structure of the specified s	an App ID   S, watchOS   App Services   NAME <ul> <li>App Attest</li> <li>App Groups</li> <li>App Payment Proce</li> </ul>	Profiles	App ID Prefix Bundle ID • Explicit • Wildcard com.imsdk.pushdemo We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat	App ID   S, watchOS   App Services   NAME   Image: Constraint of the services   Image: Constraint of the services </td <td>Profiles</td> <td>App ID Prefix         Bundle ID • Explicit       Wildcard         com.imsdk.pushdemo         We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).</td> <td>Back</td>	Profiles	App ID Prefix         Bundle ID • Explicit       Wildcard         com.imsdk.pushdemo         We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back
Certificat  All Identifiers  Register a  Platform iOS, macOS, tvOS Description IMSDK Demo You cannot use sp Capabilities ENABLED	an App ID   S, watchOS   Decial characters such as @, &, *, ', ", -, .   App Services   NAME <ul> <li>Access WiFi Information</li> <li>App Attest</li> <li>App Groups</li> <li>App I Apple Pay Payment Proced</li> <li>Associated Domains</li> <li>AutoFill Credential Provide</li> </ul>	Profiles	App ID Prefix         Bundle ID • Explicit       Wildcard         com.imsdk.pushdemo         We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).	Back

7. Check the **Push Notifications** box to enable the remote push service.

< All Identifier	S	
Registe	er an App ID	Back Continu
	<b>√→</b> Multipath (j)	
	Network Extensions	
	N) NFC Tag Reading 🕕	
	VPN Personal VPN	
	Push Notifications	
	Sign In with Apple (i)     Configure	
	SiriKit	
	System Extension	
	OUSer Management	
	Wallet (1)	
	Wireless Accessory Configuration	
	Mac Catalyst (Existing Apps Only) (1)	

#### **Certificate Generation**

1. Select your AppID and choose **Configure**.

< All Identifie	rs	
Edit yo	our App ID Configuration	Remove Save
	Network Extensions 🕦	
	N) NFC Tag Reading	
	VPN Personal VPN	
	Push Notifications (i)	Configure Certificates (0)
	Sign In with Apple 👔	Configure
	SiriKit 👔	
	System Extension	
	OUser Management	
	Wallet	
	S Wireless Accessory Configuration	
	Mac Catalyst (Existing Apps Only) (j)	Configure

2. In the **Apple Push Notification service SSL Certificates** window, there are two SSL Certificates for the development environment (Development) and the production environment (Production), as shown below:

É Developer	Apple Push Notification service SSL Certificates	Yardian N San Mu - 954735703
Certificat	To configure push notifications for this App ID, a Client SSL Certificate that allows your notification server to connect to the Apple Push Notification Service is required. Each App ID requires its own Client SSL Certificate. Manage and generate your certificates below.	
< All Identifiers	Development SSL Certificate	
Edit your App	Create an additional certificate to use for this App ID.	Remove
Platform	Create Certificate	
iOS, macOS, tvOS, watchO Description	Production SSL Certificate	
TPNS SDK demo	Create an additional certificate to use for this App ID.	
You cannot use special cha	Create Certificate	
Capabilities		
ENABLED NAME	Done	
	cess WiFi Information (1)	
- (2) Ap	p Attest (i)	
[	p Groups (i) Configure	
- 🖻 Ap	ple Pay Payment Processing (i) Configure	
As	sociated Domains ①	
<u> </u>		

### 3. We

first select the **Create Certificate** for the Development environment, the system will prompt us that we need a Certificate Signing Request (CSR).

< All Certificates
Create a New Certificate Back Conti
<b>Certificate Type</b> Apple Push Notification service SSL (Sandbox)
Diatform
iOS v
Upload a Certificate Signing Request To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Ma Learn more > Choose File

Request a Certificate From a Certificate Authority ).

Keychain Access F	ile Edit	View Window Help
About Keychain Acces	S	
Settings	ж,	Non-temperature
Certificate Assistant	>	Open
Ticket Viewer	∕Σ ೫ K	Create a Certificate
Services	>	Create a Certificate Authority
Hide Keychain Access	ж н	Request a Certificate From a Certificate Authority
Hide Others	∼жн	Set the default Certificate Authority
Show All		Evaluate a Certificate
Quit Keychain Access	жQ	Keychain Access

5. Enter your email address, Common Name (your name or company name), select **Save to disk**, click **continue**, the system will generate a \*.certSigningRequest file.

	Certificate Assistant
	Certificate Information
	Enter information for the certificate you are requesting. Click Continue to request a certificate from the CA.
Certo	User Email Address: youre mail@example.com Common Name: IMSDK CA Email Address: Required Request is: • Emailed to the CA Saved to disk Let me specify key pair information

Continue

6. Go back to the page on the Apple Developer website mentioned in Step 3, click **Choose File** to upload the generated \*.certSigningRequest file.

Cha	at
-----	----

< All Certificates
Create a New Certificate Back Continu
<b>Certificate Type</b> Apple Push Notification service SSL (Sandbox)
Platform:
iOS
<b>Upload a Certificate Signing Request</b> To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac. Learn more >

٢A	NI Certificates	
С	create a New Certificate	ck Continue
<b>Ce</b> Ap	ertificate Type Iple Push Notification service SSL (Sandbox)	
Pla	atform:	
i	os 🗸 🗸	
Ur To Lea	bload a Certificate Signing Request manually generate a Certificate, you need a Certificate Signing Request (CSR) fil arn more >	le from your Mac.

É Developer		
Certificates, Ide	entifiers & Profiles	
< All Certificates Download Your Certif	ficate	
Certificate Details		
Certificate Name com.tpnssdk.pushdemo	Certificate Type APNs Development iOS	Download your certificate to your Mac, then double clic Keychain Access. Make sure to save a backup copy of
Expiration Date 2021/09/20	Created By	somewnere secure.

9. Repeat steps 1 - 8 above to download the Production SSL Certificate for the production environment to

your local machine.

Note:

The certificate for the production environment is actually a combined certificate of Development (Sandbox) +

Production, and it can be used as a certificate for both the development and production environments.

#### É Developer

< All Certificates		
Create a New Certi	ificate	
Certificate Type Apple Push Notification service SSL (	(Sandbox & Production)	
Platform:		
iOS	~	
Choose File	CertificateSigningRequest.certSigningRec	quest
É Developer		
Developer Certificates, Id	entifiers & Profiles	
Developer Certificates, Id	entifiers & Profiles	
Developer Certificates, Id All Certificates Download Your Certificates	entifiers & Profiles	
Developer Certificates, Id All Certificates Download Your Certificates	entifiers & Profiles	
Developer Certificates, Id All Certificates Download Your Certificate Certificate Details	entifiers & Profiles	
Developer Certificates, Id All Certificates Download Your Certificate Certificate Details Certificate Name com.tpnssdk.pushdemo	entifiers & Profiles ificate Certificate Type Apple Push Services	Download your certificate to your Mac, then o Keychain Access. Make sure to save a backu

system will import it into the keychain.

11. Open the Keychain App, go to log in to > My Certificates, right-click to export the newly created Apple Development IOS Push Services and Apple Push Services for the development and production environments as p12 files respectively.

	Keychain Access						λ Search	)
Default Keychains	All Items Passwords Secure	Notes My Certificate	s Keys Certificates					
🗗 login	Angle Duck Con							
Local Items	Certificate Standard Conduct Certificate Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Subset Sub	orldwide Developer Re December 5, 2024 at 1 s valid	lations Certification Authority 15:52:39 China Standard Time	e <b>rvice</b> / e				
🔒 System								
System Roots	Name			~	Kind	Date Modified		Expir
	Apple Persistent State End	ryption			application password	Yesterday, 15:09		
	Apple Push Services: com	).xa	1		certificate			Oct 2
	Apple Push Services: com	xa	1		certificate			Dec 8
	🔀 Apple Push Services: com	.xa	i.db		certificate	~ <del>~-</del>	976	Mar 2
	Apple Push Services: com	d Nev	v Identity Preference					Dec 5
	apple Worldwide Develope	r Relations Cert	and a second sec					Feb 2
	Apple Worldwide Develope	r Relations Cerl Cop	y "Apple Push Services: c	com.	p	oushservice"		Mar 1
	Apple Worldwide Develope	r Relations Cer Delf	ete "Apple Push Services:	com.		pushservice"		Feb 8
	Apple Worldwide Develope	r Relations Ceri	and the second s	- Children			- Sherr	Jan 2
	Apple Worldwide Develope	r Relations Cerl Exp	ort "Apple Push Services:	com.t		pushservice"		Dec 1
	AppleIDClientIdentifier						7:54	
	Basic Attestation User Suc	Get Get	Info				.00	Mar 2
	Beaconstore	Eval	luate "Apple Push Service	s: com		pushservice"	.02	

#### Note

Set a password when saving the .p12 file.

#### Step 2: Upload the certificate to the console

- 1. Log in to the Chat Console.
- 2. Click Plugin Service-Push-Access Settings to enter the access settings page



- 3. Click Add Certificate at the bottom of iOS in Vendor Configuration.
- 4. Select the certificate type, upload the iOS Certificate (.p12), set the certificate password, and click Confirm.

Access settings 2 <sup>r</sup> ,st •		ingapore 🚯 Telegram group WhatsApp group		
① The push plug-in trial is about to expire and push serv				
Push Overview free trial Remaining 3 days All mem				
				_
Normal push     All	Add iOS Certificate		×	Push dat
	Push Type	APNs Push VolP Push		<ul> <li>Activat</li> </ul>
Purchase Now free trial	Certificate Type	Production environment     Development Environment		
		Please check that the uploaded certificate type is Apple Push Notification service SSL (Sandbox & Production), and test it after the Archive releases the Release package. Note: It cannot be tested in Xcode.		
1 Manufacturer configuration	Configuration type	<b>O</b> p12		
IM supports online and offline push notifications. On	iOS certificate (.p12) *	Select file		-level push char
		How to generate an APNs certificate?		
Android				
Add Certificate	mutable-content 🚯			
	Certificate password *	Enter the certificate password		
		Confirm		
No certific				
2 Local deployment				
After purchasing the push plug-in, you can use the *c	uick configuration" and "do	wnload configuration file" functions. With 3 lines of code and 2 minutes, you can complete push access from m	ultiple ma	nufacturers at on

#### Note:

We recommend naming the uploaded certificate in English (special characters such as brackets are not allowed). You need to set a password for the uploaded certificate. Without a password, push notifications cannot be received. For an app published on App Store, the environment of the certificate must be the production environment. Otherwise, push notifications cannot be received.

The uploaded .p12 certificate must be your own authentic and valid certificate.

5. After the pending certificate information is generated, record the certificate's ID.



# Second, using a p8 certificate (supports Dynamic Island push notifications)

p8 Certificate: A p8 certificate does not have an expiration date, so you don't have to worry about the certificate expiring. Moreover, using a p8 certificate can simplify certificate management, as you can use a single p8 certificate to provide push notification services for multiple applications. In addition, p8 certificates support Dynamic Island push notifications.

### Step 1: Apply for an APNs certificate

To create a p8 certificate file, first log in to Apple Developer Center.



1. Enter Certificates, Identifiers & Profiles: In the top right corner of the page, click **Account**, then select **Certificates**, **Identifiers & Profiles** from the dropdown menu.

2. To create a new App ID: in the left-hand menu, click Identifiers, then click the + on the right to create a new App

ID. Fill in the relevant information and click Continue.

3. To create a new key: in the left-hand menu, click **Keys**, then click the **+** on the right to create a new key. Enter the name of the key, then check **Apple Push Notifications service (APNs)** and click **Continue**.



Download Your Key
time, click Done and download it at a later time. Be sure to save a backup of your key in a secure place.

Confirm and generate the key: On the confirmation page, verify your key information, then click Register. Next, you'll see a page prompting you to download the key. Click **Download** and save the generated .p8 file to your computer.

#### Note:

The p8 certificate can only be downloaded once; please save it properly.

Please safeguard the downloaded p8 file, as you will not be able to download it again. You can use this p8 certificate to configure your iOS applications to receive push notifications.

#### Step 2: Upload the p8 certificate to the IM console

- 1. Log in to the Chat Console.
- 2. Click Plugin Service-Push-Access Settings to enter the access settings page
- 3. Click Add Certificate at the bottom of iOS in Vendor Configuration.
- 4. Select the .p8 certificate

## **Operation step**

#### Step 1. Register your app with vendor push platforms

To utilize the offline push feature, you need to register your app on each vendor's push platform to obtain parameters such as AppID and AppKey. Currently, the mobile manufacturers supported in China include: Mi, Huawei, Honor, OPPO, Vivo, Meizu, and internationally Google FCM is supported.

#### Step 2. Create resources in the IM console

Log in to Tencent Cloud Chat Console, then in the Push Management > Access Settings feature section, add each vendor's push certificate, and configure the AppID, AppKey, AppSecret, and other parameters obtained in Step 1 to the added push certificate.

#### Explanation of the Subsequent Actions option:

Open Application: Clicking the notification bar launches the app, by default starting the app's Launcher interface;

Open Web Page: Clicking the notification bar will redirect to the configured web link;

Open the specified interface within the app: clicking the notification bar will redirect the interface based on the configured self Definition, see Custom Redirect on Click.

Mi	
Huawei	
OPPO	
vivo	
Meizu	
HONOR	
Google FCM	
Vendor Push Platform	Configuring in the IM console

Vendor Push Plat	form	Configuring in the	IIVI console
		Add Mi contifica	
	· 推送运营平台 Tuke •	Package Name *	Enter Package Name
✔ 推送工具	TUIKit	AppID *	Enter AppID
☆ 推送统计 Ⅲ 应用管理 ~	自用类型 Android	АррКеу *	Enter AppKey
<ul> <li>应用信息</li> <li>通知类別</li> </ul>	<b>主告名 设置多信名</b> 了解多信名使用力法	AppSecret *	Enter AppSecret
角色管理	AppKey â.â	Region	Enter a channel ID
<ul> <li>○ 调查工具</li> <li>[] 法律文档</li> </ul>	AppSourcet 848	Response after	Open application
		Cilor	*Note: the Mi "onNotific using this method.

Vendor Push Platform	Configuring in the IM cons



Vendor Push Platform	Configuring in the IM cons
	Add OPPO certificate
OPPO官网 ColorOS社区 井虹平台 用户中心 OPPO○   推送运营平台 := 10 5.5 tulkit ✓	Package Name * Enter F
	AppKey • Enter /
✓ 创建推进 ✓ 首页 > 应用配置	AppID • Enter /
④ 数据统计 V tuikit	AppSecret *
	MastarSecret *
記置管理   Appld	ChannellD Enter a
检查工具     AppKey cf     查看       新時建遊道     AppSecret 75************************************	Response after Oper Click
通道設置 推送链 MasterSecret 重 重 二 二 二 二 二 二 二 二 二 二 二 二 二 二 二 二 二	

Vendor Push Platform

Configuring in the IM cons

		Add vivo certif
		Package Name *
88 云通信IM ∨	应用信息	AppKey *
₩ 推送统计	应用名称: 云通信IM	AppID *
ⓒ 应用管理 ^	应用类别:移动应用 推送权限:正式	Receipt ID
应用信息	审核状态: 已通过 创建时间:	Category
测试设备	应用包名:	AppSecret *
◇ 标签管理 ◇ 在线诊断	AppID: AppKey: ApoSecret: 重音 0 恢复 0	Response after Click

For receipt configuration, please refer to: Message reach statistics configuration - vivo.

Vendor Push Platform	Configuring in the IM cons
Flyme 推送平台 首页 创建推送 数据统计 配置管理	
应用配置 标签用户 问题排查 黑名单 回执管理 常用设备 多包名 任务备注	Add Meizu certificate
TUIKit	Package Name * Enter F
应用名称 TUIKit	AppID • Enter A
应用形态 普通应用 应用包名 汤加多包名	AppKey • Enter A
应用类型 通讯社交 ∨ 应用图标 更換图片 R寸为480*480, 500KB5(内	receipt switch Offer turn configure
	AppSecret • Enter A
© App ID	Response after Oper Click
О Арр Кеу     О Арр Secret     重置	
快速集成 下载代码 扫描下载DemoAPK	

#### For receipt configuration, please refer to: Message reach statistics configuration - Meizu.

Vend	or Push	Platform	Configuring in the IM cons
	но	NOR	
	٢	开放能力 / 推译职名 / 查罢推祥服名	Add Honor certificate
	ංදී	查看推送服务	Package Name * Enter F
	0		AppID • Enter A
	6	应用条型:移动应用 应用名称:腾讯云通信IM	ClientID • Enter t
		应用包名:	ClientSecret *
		SHA256证书指纹1: • · · · · · · · · · · · · · · · · · ·	ChannellD Enter a
		APP ID:	Badge Parameter Please
		APP Secret:	*Note: It
		Client ID:	Response after Open
		Client Secret:	Click
		Android端SDK: 《点击下载荣耀PUSH Android端SDK》	
		Android 讲接入文档: 《点击下载荣耀PUSH Android 端接入文档》	
		服务靖接入文档:《点击下载荣耀PUSH服务靖接入文档》	

Vendor Push Platform	Configuring in the IM cons

Firebase	1 ×		6 + 0 B A		
A Project Overview	Project settings General Cloud Metasging Integrations Service accounts Data privacy Users and permissions			Add ECM cortif	ioato
Generative AI				Add i Oim certificate	
+ Build with Gemini (NTM)					
Project shortcuts		Manage service account permissions	Ø	Adding Method	O Up
Authentication	Ov Firebase Admin 5	K Firebase Admin SDK		, is a line line line line line line line line	•
Test Lab		Your Firebase service account can be used to authenticate multiple Firebase features, such as		Message type	
C Remote Config	Legacy credentials	Database, Storage and Auth, programmatically via the unified Admin SDK. Learn more Z		Message type	
App Check	Database secrets	Firebase service account			Transpa
What's new	All service accounts				availab
App Hosting (NW)	S 7 service account	Admin SDK configuration snippet			SDK er
Se Data Connect (NW)		💿 Nodejs 🔘 Java 🔘 Python 🔘 Go			001101
Product categories					
uild v		<pre>var admin = require("firebase-admin");</pre>		Package Name *	Enter
		<pre>var serviceAccount = require("path/to/serviceAccountKey.json");</pre>			
un v		<pre>admin.initializeApp({     credential: admin.credential.cert(serviceAccount)</pre>			
Analytics ~		)); <b>(</b>		Upload certificate	
All products					
		Generate new private key			How to
				ChappellD	Entor
				GhanneilD	Ent
				onamone	

#### Note:

Regarding **Click for Subsequent Actions** supports the Report Statistics feature:

1. If you choose to open an app or a web page, purchasing the plugin will by default support reporting statistics.

2. If you choose to open a specified interface within the application:

For new certificate status, please directly use the auto-fill default value to support click statistics reporting.

If there was a certificate configured previously, continue using the old certificate and modify it to the default value to support reporting statistics, or regenerate a new certificate.

#### About FCM Data Messaging

FCM provides two push methods: Notification Message and Data Messaging.

Notification Message, with a simple style not differentiated by device, can support offline push once successfully integrated;

Data Messaging, offering rich customization for specific devices, supports reach and click reporting, and requires testing on the device before going live after integration.

The console defaults to Notification Message, and switching between modes can be done in the IM Console:



#### Note:

FCM Data Messaging capability is only supported on Pixel phones with TIMPush 7.8 and above, other manufacturers' devices need to be self-tested for support;

# Quick Integration Android

Last updated : 2024-08-01 11:13:26

## Step 1: Download and add the configuration file

After completing the console manufacturer push information, download and add the configuration file to the project. Add the downloaded timpush-configs.json file to the assets directory of the application module:

Step 2: Integrate the TIMPush plugin





```
// Integration of the push main package is mandatory
implementation 'com.tencent.timpush:timpush:8.0.6897'
// Integrate the corresponding manufacturer as needed
implementation 'com.tencent.timpush:fcm:8.0.6897'
implementation 'com.tencent.timpush:huawei:8.0.6897'
implementation 'com.tencent.timpush:xiaomi:8.0.6897'
implementation 'com.tencent.timpush:vivo:8.0.6897'
implementation 'com.tencent.timpush:honor:8.0.6897'
implementation 'com.tencent.timpush:honor:8.0.6897'
implementation 'com.tencent.timpush:honor:8.0.6897'
implementation 'com.tencent.timpush:honor:8.0.6897'
implementation 'com.tencent.timpush:meizu:8.0.6897'
// Choose one of the two below
// For the China region, choose to integrate this package
```

```
implementation 'com.tencent.timpush:oppo:8.0.6897'
// For other regions, choose to integrate this package
implementation 'com.tencent.timpush:oppo-intl:8.0.6897'
```

#### Note:

1. TIMPush requires integration with IMSDK version 7.6.5011 or above.

2. For users without UI or who haven't integrated other plugins, it is necessary to add integration with TUICore. It supports both source and Maven integration, as follows:



```
def projects = this.rootProject.getAllprojects().stream().map { project -> project.
api projects.contains("tuicore") ? project(':tuicore') : "com.tencent.liteav.tuikit
```



#### vivo and Honor configuration

According to vivo and Honor manufacturer's access guidance, it is necessary to add the APPID and APPKEY to the manifest file, otherwise compilation issues will occur.

Method 1

Method 2



android { . . . defaultConfig { . . .

```
manifestPlaceholders = [
    "VIVO_APPKEY" : "`APPKEY` of the certificate assigned to your applicati
    "VIVO_APPID" : "`APPID` of the certificate assigned to your applicatio
    "HONOR_APPID" : "`APPID` of the certificate assigned to your applicatio
    ]
}
```



// vivo begin
<meta-data tools:replace="android:value"
 android:name="com.vivo.push.api\_key"</pre>

android:value="`APPKEY` of the certificate assigned to your application" />
<meta-data tools:replace="android:value"
 android:name="com.vivo.push.app\_id"
 android:value="`APPID` of the certificate assigned to your application" />
// vivo end
// honor begin
<meta-data tools:replace="android:value"
 android:name="com.hihonor.push.app\_id"
 android:value="`APPID` of the certificate assigned to your application" />
// honor end

#### Huawei, HONOR, and Google FCM Adaptation

Follow the vendor method to integrate the corresponding plugin and JSON configuration file.

#### Note:

The following adaptation for HONOR only requires configuration for version 7.7.5283 and above.

1.1 Download the configuration file and place it under the root directory of the project:

Huawei

HONOR

Google FCM

**Operation Path** 

AppGallery Connect	全部服务 ~ 我的项目 ~	TUIKit 🗸 📵 TUIKit 🗸	ф <sup>0</sup> (?)
项目设置	常规 API管理	Server SDK 项目套餐 项目配额 项目费用	
盈利			
20 应用联运	应用		
₽₽ 游戏联运	SDK配置:	下载最新的配置文件(如果您修改了项目、应用信息或者更改了某个开发服务设置,可能需要更新该文件)	
其○ 付费下载		agconnect-services.json 不包含密钥 ⑦	
🖳 应用内支付服务		添加SDK	
	包名:	,	
增长 ^	APP ID:		
<i>③</i> 推送服务	SHA256证书指纹: ⑦	E 🖉 🔟	
[a] A/B测试			
₀S₀ 动态标签管理			
🕤 远程配置			
📄 应用内消息			
	OAuth 2.0客户端ID(凭据):		
Indexing		Client Secret	
(2) 智能运营	回调地址: ⑦		
<u> </u>		删除应用	
查 直达应用快服务			

HONOR Developers			
⊘ 生态服务 ∧	生态服务 > 应用管理 > <b>应用基础信息查看</b>		
应用服务	应用基础信息		
游戏服务	应用基础信息做任何更改,将在提交保存后生效		
智慧服务			
内容服务	应用名称:		
应用管理	更新时间:		
A HONOR Connect	App ID:		
6 HONOK Connect			
☆ 开放能力 ∨	平台类型: 安卓		
🔗 开发者中心 🗸 🗸	< 应用类型: 应用		
	支持设备: 手机/平板		
	默认语言: 简体中文 🦉		
	SDK 配置: <u>下载最新的配置文件(如果您</u> 修改了应用信息或者更改了某个开发服务设置,可能需要更新该文件)		
	(↓ mcs-services.ison) 添加 SDK		

🔶 Firebase	I		
Project Overview			
Generative AI Build with Gemini (NEW) Project shortcuts Authentication Test Lab C> Messaging C Remote Config		Your apps Android apps	SDK setup and configuration Need to reconfigure the Firebase SDKs for your app? F download the configuration file containing keys and id
<ul> <li>App Check</li> <li>What's new</li> <li>App Hosting NEW</li> <li>Data Connect NEW</li> <li>Product categories</li> <li>Build ~</li> <li>Run ~</li> <li>Analytics ~</li> </ul>		Apple apps	Image: See SDK instructions <ul> <li>google-service</li> <li>App ID ⑦</li> <li></li> <li>App nickname</li> <li>Add a nickname</li> <li>Add a nickname</li> <li>Package name</li> <li></li></ul>
III All products			Add fingerprint Remove this a
Spark Upgrade			i Delete project



1.2 Add the following configuration under buildscript -> dependencies in your project-level build.gradle file:



For Gradle version 7.1 and above

Gradle version 7.0

Versions Below Gradle 7.0

Add the following configuration under buildscript -> dependencies in your project-level build.gradle file:



```
buildscript {
    dependencies {
        ...
        classpath 'com.google.gms:google-services:4.3.15'
        classpath 'com.huawei.agconnect:agcp:1.6.0.300'
        classpath 'com.hihonor.mcs:asplugin:2.0.1.300'
```



```
}
```

In the project-level settings.gradle file, add the following repository configurations under pluginManagement -> repositories and dependencyResolutionManagement -> repositories:



```
pluginManagement {
    repositories {
        gradlePluginPortal()
        mavenCentral()
        maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
```
```
// Configure the Maven repository address for HMS Core SDK.
        maven {url 'https://developer.huawei.com/repo/'}
        maven {url 'https://developer.hihonor.com/repo'}
    }
}
dependencyResolutionManagement {
    . . .
    repositories {
       mavenCentral()
       maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
        // Configure the Maven repository address for HMS Core SDK.
       maven {url 'https://developer.huawei.com/repo/'}
       maven {url 'https://developer.hihonor.com/repo'}
    }
    }
}
```

Add the following configuration under buildscript in your project-level build.gradle file:





```
buildscript {
    repositories {
        mavenCentral()
maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
// Configure the Maven repository address for HMS Core SDK.
maven {url 'https://developer.huawei.com/repo/'}
maven {url 'https://developer.hihonor.com/repo'}
    }
    dependencies {
        ...
        classpath 'com.google.gms:google-services:4.3.15'
```

```
classpath 'com.huawei.agconnect:agcp:1.6.0.300'
classpath 'com.hihonor.mcs:asplugin:2.0.1.300'
}
```

Add the following repository configurations under dependencyResolutionManagement -> repositories in your projectlevel settings.gradle file:



```
dependencyResolutionManagement {
```

```
...
repositories {
```

```
mavenCentral()
maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
// Configure the Maven repository address for HMS Core SDK.
maven {url 'https://developer.huawei.com/repo/'}
maven {url 'https://developer.hihonor.com/repo'}
}
```

Add the following configuration under buildscript and allprojects in the project-level build.gradle file:



```
buildscript {
    repositories {
        mavenCentral()
maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
// Configure the Maven repository address for HMS Core SDK.
maven {url 'https://developer.huawei.com/repo/'}
maven {url 'https://developer.hihonor.com/repo'}
    }
    dependencies {
        . . .
        classpath 'com.google.gms:google-services:4.3.15'
        classpath 'com.huawei.agconnect:agcp:1.6.0.300'
        classpath 'com.hihonor.mcs:asplugin:2.0.1.300'
    }
}
allprojects {
    repositories {
        mavenCentral()
maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
// Configure the Maven repository address for HMS Core SDK.
maven {url 'https://developer.huawei.com/repo/'}
maven {url 'https://developer.hihonor.com/repo'}
    }
}
```

1.3 Add the following configuration in the app-level build.gradle file:





apply plugin: 'com.google.gms.google-services'
apply plugin: 'com.huawei.agconnect'
apply plugin: 'com.hihonor.mcs.asplugin'

After the above steps are performed, offline push notifications can be received.

#### Note:

If you want to integrate the TIMPush component as simply as possible, you need to log in and log out using the login/logout APIs provided by TUILogin of the TUICore component, and the TIMPush component automatically

senses the log in and log out events. If you don't want to use the APIs provided by TUILogin, after completing the login/logout operation, you need to manually call the registerPush/unRegisterPush APIs of TIMPushManager.

# Step 3: Set obfuscation rules

In the proguard-rules.pro file, add TIMPush-related classes to the non-obfuscation list:



```
-keep class com.tencent.qcloud.** { *; }
-keep class com.tencent.timpush.** { *; }
```

# **Step 4: Message Delivery Statistics Configuration**



If you need to collect data reach statistics, please complete the configuration as follows:

Huawei

HONOR

vivo

Meizu

AppGallery Connect	全部服务 ~ 我的项目 ~	TUIKit ~
目设置	推送通知 自动推送通知(Beta) 推送报告 自助分析(Beta)	R
	推送服务	
□ 游戏联运	建立云端铁线编织清音路送逝进,为你提供东村,高效,满高的清启得还服务。 缓醒:如果不透祥数据存储位置,基于主题/设备组Weoh推送发送功能不可用。如果不开透高设分析功能,AS测起/预测/	回执配置 <sub>①</sub> ×
➡。 付费下载 ➡ 应用内支付服务	数据存储位置:中国	• 回执名称 0.50
□ 华为钱包	发送者ID: 9105385871708321887 章音度成功算	• 回调地址 https://
K AD FIDE HIS V	项目回换状态 3 未开通 开道	<b>回调用户名</b> 0.50
推送服务     A/B测试	精准推送能力 未开通 开道	回荷松明 (2010) ● V2 ○
P。动态标签管理	选择应用	
這程配置 应用內消息	O TUIKet 未配置 ~	
App Linking	包名 com.tencent.qcloud.tim.tuikit	

### **Receipt Address:**

Singapore	https://apisgp.im.qcloud.com/v3/offline_push_report/huawei
South Korea	https://apikr.im.qcloud.com/v3/offline_push_report/huawei
USA https:/	/apiusa.im.qcloud.com/v3/offline_push_report/huawei
Germany	https://apiger.im.qcloud.com/v3/offline_push_report/huawei
Indonesia	https://apiidn.im.qcloud.com/v3/offline_push_report/huawei
Others in Chin	a https://api.im.qcloud.com/v3/offline_push_report/huawei

#### Note:

Huawei Push Certificate ID <= 11344, using Huawei Push v2 interface, does not support Reach and Click Receipt, please regenerate and update Certificate ID.

но	NOF	R Developers					文档	智能客服	158*****40
0		开放能力 / 推送服务 / 应用回执. 【 <b>应用回执</b>							
€		应用类型:移动应用 应用名称: <b>4</b>	回执配置			×			
		应用包名:	* 回执名称:	测试回执	4/50				
	>		* 回调地址: 回调用户名:	https://test.tim.qq.com/v3/offline_push_report/honor 只能输入英文字母、数字和下划线	0/50				
			回调密钥:	<ul> <li>长度不超过 128, 不小于 16 字符</li> <li>2 到达问执 2 点击问执</li> </ul>					
			HINGE .	取消 提交		ł			

# Receipt Address:

Singapore	https://apisgp.im.qcloud.com/v3/offline_push_report/honor
South Korea	https://apikr.im.qcloud.com/v3/offline_push_report/honor
USA https:/	/apiusa.im.qcloud.com/v3/offline_push_report/honor
Germany	https://apiger.im.qcloud.com/v3/offline_push_report/honor
Indonesia	https://apiidn.im.qcloud.com/v3/offline_push_report/honor
Others in Chin	ha https://api.im.qcloud.com/v3/offline_push_report/honor

Callback Address Configuration	Configure Receipt ID in the
▶ ▼ 1 ▼ 0 开放平台   推送运营平台	添加Android证书
88 云通信M V 应用信息	AppKey • 请输入AppK
✓ 推送工具 ✓ 应用名称: ************************************	AppID • 请输入AppIC
du 推送统计 ~ 应用类别: 移动应用	
③ 应用管理         ^         推送权限:正式	
应用信息 审核状态: 已通过	Category 请输入 Cate
測试设备 应用包名: 000000000000000000000000000000000000	AppSecret • 请输入AppS
√ 标签管理 AppID: ●LULIT	□ 古王后续动作 ● 打开应用
AppKey: ① ご 「 「 「 」 : 「 」 : 「 」 : 「 」 : 「 」 : 」 : 」	MULTERENT F
Receipt Address:	

# 🔗 Tencent Cloud

Singapore
https://apisgp.im.gcloud.com/v3/offline push report/vivo
South Korea
https://apikr.im.qcloud.com/v3/offline_push_report/vivo
USA
https://apiusa.im.qcloud.com/v3/offline_push_report/vivo
Germany
https://apiger.im.qcloud.com/v3/offline_push_report/vivo
Indonesia
https://apiidn.im.qcloud.com/v3/offline_push_report/vivo
Others in China
https://api.im.qcloud.com/v3/offline_push_report/vivo

le Receipt Sw	tch			Configure Receipt A
添加魅族证书			×	
应用包名称 •	请输入应用包名称	如何生成魅族证书? 🕻		
AppID •	请输入AppID			<b>立田紀章 指称田户 问题</b>
AppKey •	请输入AppKey			2017年1月1日 11日 11日 11日 11日 11日 11日 11日 11日 11日
回执开关	•			① 图执地址
AppSecret *	打开回执开关后,请务必参照文档	2,正确配置回执地址,否则将会导致离线推	送失败。	回执列表 回执地址
点击后续动作		打开应用内指定页面		图换地址
应用内指定界面・	com.tencent.qcloud.tim.push.TIM 此处会回调点击通知栏事件,通过广	/ 播或者回调给到应用,App可以在回调中自行	<b>示</b> 处理打开应用	
		确定		

# Receipt Address:

Singapore	https://apisgp.im.qcloud.com/v3/offline_push_report/meizu
South Kore	https://apikr.im.qcloud.com/v3/offline_push_report/meizu
USA htt	ps://apiusa.im.qcloud.com/v3/offline_push_report/meizu
Germany	https://apiger.im.qcloud.com/v3/offline_push_report/meizu
Indonesia	https://apiidn.im.qcloud.com/v3/offline_push_report/meizu
China	https://api.im.qcloud.com/v3/offline_push_report/meizu
Note:	

After turning on the receipt toggle, please make sure the receipt address is correctly configured. Not configuring it or configuring the wrong address will affect the push notification feature.

#### Note:

No configuration for message delivery statistics is required for other supported manufacturers. FCM currently does not support the push notification statistics feature.

# Step 5: Set the offline push parameters when sending a message

When calling sendMessage to send a message, you can set offline push parameters through V2TIMOfflinePushInfo. By calling V2TIMOfflinePushInfo's setExt, you can set custom Definition ext data, which allows you to access the ext field in the callback of the click notification jump when the user starts the App after receiving an offline push. Then, based on the content of the ext field, navigate to the specified UI interface. See the sendMessage() method of ChatProvider for reference:





```
V2TIMOfflinePushInfo v2TIMOfflinePushInfo = new V2TIMOfflinePushInfo();
v2TIMOfflinePushInfo.setTitle("Push Title");
v2TIMOfflinePushInfo.setDesc("Push Content");
OfflinePushExtInfo offlinePushExtInfo = new OfflinePushExtInfo();
offlinePushExtInfo.getBusinessInfo().setSenderId("senderID");
offlinePushExtInfo.getBusinessInfo().setSenderNickName("senderNickName");
if (chatInfo.getType() == V2TIMConversation.V2TIM_GROUP) {
    offlinePushExtInfo.getBusinessInfo().setChatType(V2TIMConversation.V2TIM_GROUP)
    offlinePushExtInfo.getBusinessInfo().setSenderId("groupID");
```

```
v2TIMOfflinePushInfo.setExt(new Gson().toJson(offlinePushExtInfo).getBytes());
// For OPPO, you must set the `ChannelID` to receive push messages. The `ChannelID`
v2TIMOfflinePushInfo.setAndroidOPPOChannelID("tuikit");
v2TIMOfflinePushInfo.setAndroidHuaWeiCategory("IM");
v2TIMOfflinePushInfo.setAndroidVIVOCategory("IM");
final V2TIMMessage v2TIMMessage = message.getTimMessage();
String msqID = V2TIMManager.getMessageManager().sendMessage(v2TIMMessage, isGroup ?
   V2TIMMessage.V2TIM_PRIORITY_DEFAULT, false, v2TIMOfflinePushInfo, new V2TIMSend
        @Override
        public void onProgress(int progress) {
        }
        @Override
        public void onError(int code, String desc) {
            TUIChatUtils.callbackOnError(callBack, TAG, code, desc);
        }
        @Override
        public void onSuccess(V2TIMMessage v2TIMMessage) {
            TUIChatLog.v(TAG, "sendMessage onSuccess:" + v2TIMMessage.getMsgID());
            message.setMsgTime(v2TIMMessage.getTimestamp());
            TUIChatUtils.callbackOnSuccess(callBack, message);
        }
    });
```

# Step 6: Parsing Offline Push Messages

When a push is received, by clicking on the notification bar event, the component will notify the application in the form of a callback or broadcast. The application can then configure the App's redirect page in the callback. It is recommended to register the callback during the application's oncreate() function in the Application. The callback method is as follows:





TUICore.registerEvent(TUIConstants.TIMPush.EVENT\_NOTIFY, TUIConstants.TIMPush.EVENT @Override public void onNotifyEvent(String key, String subKey, Map<String, Object> pa Log.d(TAG, "onNotifyEvent key = " + key + "subKey = " + subKey); if (TUIConstants.TIMPush.EVENT\_NOTIFY.equals(key)) { if (TUIConstants.TIMPush.EVENT\_NOTIFY\_NOTIFICATION.equals(subKey)) if (param != null) { String extString = (String)param.get(TUIConstants.TIMPush.N // Getting ext for Definition redirect // Example: Redirect to the corresponding chat interface



The broadcast method is as follows:





```
// Dynamic Broadcast Registration
IntentFilter intentFilter = new IntentFilter();
intentFilter.addAction(TUIConstants.TIMPush.NOTIFICATION_BROADCAST_ACTION);
LocalBroadcastManager.getInstance(context).registerReceiver(localReceiver, intentFi
// Broadcast Receiver
public class OfflinePushLocalReceiver extends BroadcastReceiver {
    public static final String TAG = OfflinePushLocalReceiver.class.getSimpleName()
    @Override
    public void onReceive(Context context, Intent intent) {
```

}

```
DemoLog.d(TAG, "BROADCAST_PUSH_RECEIVER intent = " + intent);
    if (intent != null) {
        String ext = intent.getStringExtra(TUIConstants.TIMPush.NOTIFICATION_EX
        // Getting ext for Custom Redirect
        // Example: Redirect to the corresponding chat interface
        OfflinePushExtInfo offlinePushExtInfo = null;
        try {
            offlinePushExtInfo = new Gson().fromJson(extString, OfflinePushExtI
            if (offlinePushExtInfo.getBusinessInfo().getChatAction() == Offline
                String senderId = offlinePushExtInfo.getBusinessInfo().getSende
                if (TextUtils.isEmpty(senderId)) {
                    return;
                }
                TUIUtils.startChat(senderId, offlinePushExtInfo.getBusinessInfo
            }
         } catch (Exception e) {
             Log.e(TAG, "getOfflinePushExtInfo e: " + e);
         ł
    } else {
        Log.e(TAG, "onReceive ext is null");
}
```

Congratulations! You have completed the integration of the push plugin. Please be reminded: After the trial period or subscription expires, the push service (including regular message offline push, all-staff/Tag push, etc.) will automatically cease. To avoid affecting the normal use of your services, please make sure to purchase/renew in advance.

# uniapp

Last updated : 2024-06-13 10:21:45

# Preconditions

# 1. To integrate Chat TUIKit, see UI Integration Solution (Recommended) -> Integrate Basic feature -> uni-app (Vue2/Vue3)

### Note:

- 1. If you have already integrated Chat TUIKit, you can ignore this step.
- 2. If you are integrating Chat without UI, you can ignore this step.

## 2. Upgrade @tencentcloud/chat to the latest version.

#### Note:

@tencentcloud/chat is backward compatible, feel free to upgrade. If you are currently using tim-js-sdk or tim-wx-sdk, please refer to our Upgrade Guide.

Chat





npm install @tencentcloud/chat@latest

Check the version number of TencentCloudChat.VERSION in the HBuilder log to confirm @tencentcloud/chat ≥ 3.2.5 as shown:

10:38:25.663	Project 'sample-uniapp' start compiling
10:38:26.478	Please note that in running mode, due to log output, sourcemap, and uncompressed sour
10:38:26.481	Compiler version: 4.15 (vue3)
10:38:26.481	Compiling
10:38:34.541	Project 'sample-uniapp' compile <u>d successfully.</u>
10:38:34.543	ready in 8464ms.
10:38:34.553	Connecting device
10:38:34.938	Mobile app playground is already installed, version is 1.0.0, version is not changed,
10:38:37.319	Synchronizing app resources
10:38:37.531	[adapter-vue]: vue version is 3 <u>at TUIKit/adapter-vue.ts:13</u>
10:38:37.551	Chat 10:38:36 GMT+0800 (CST).820 TencentCloudChat.VERSION:3.3.5
10:38:37.579	TUIChatEngine.VERSION:2.1.4 at node modules/@tencentcloud/chat-uikit-engine/index.js
10:38:37.590	TUICore.VERSION:2.1.4 at node modules/@tencentcloud/tui-core/index.js:1
10:38:37.590	TUICore.getInstance ok. <u>at node modules/@tencentcloud/tui-core/index.js:1</u>
10:38:37.591	UniversalAPI.VERSION:2.1.5 <u>at node modules/@tencentcloud/universal-api/index.js:1</u>
10:38:37.591	TUICustomerServer.init ok <u>at TUIKit/tui-customer-service-plugin/server.ts:13</u>
10:38:37.592	TUIServiceManager.registerService serviceName:TUICustomerServicePlugin at node module
10:38:37.592	TUIExtensionManager.registerExtension extensionID:contactList at node modules/@tencer

## 3. Activate the Push Plugin

Go to IM Console > Push, click **Purchase Now** or **Free Trial**. (Each application can try it for free once, valid for 7 days.)

Chat	Access settings	Current data center: Singapore (i)     Telegram gro	up WhatsApp group	
표준 Application Management 편 Overview 은 Users 문 Groups	Push Step 2: Select SE	All members/tag push	Function Preview O	unction Preview ○ E Push Records ● Not activated
Configuration     ✓     Webhook     C Statistics     Plugin	3: Purchase Now/free 1	rial	nnel, which is fast and reliable; for offline push, it is recommended that you	u use the system-level push channel provided by each manufacturer. The system The level p
Push     Access settings     Access test     Push Record	Android IOS M Huawei Meizu vivo Add Contificate	OPPO Honor FCM NIO		
Step 1: Push acc Troubleshooting	ess settings	sertificate yet		
Tools           Monitor           Dev Tools         ~           Integration Guide         -	Local deployment     After purchasing the push plug-in, you can use     Quick configuration     One-click integrated configuration Average     O	the "quick configuration" and "download configuration file" funct Not activated consumption is 1 hour	ions. With 3 lines of code and 2 minutes, you can complete push access f Manual configuration Need to integrate configuration one by one Average consumption	rom multiple manufacturers at once.
	•	1	•	

#### Note:

After the trial or purchase of the Push Plugin expires, the push service (including normal message offline push, all member/Tag push, etc.) will automatically stop. To avoid affecting the normal use of your business, please

Chat

purchase/renew in advance.

# 4. Manufacturer Configuration

### Note:

The uniapp Manufacturer Configuration includes Android Manufacturer Configuration and iOS Manufacturer Configuration, see uniapp.

# Integrating TencentCloud-TIMPush

# Step 1: manifest.json App Module Configuration

In the project [manifest.json] > [App Module Configuration], configure the Message Push Module as shown:

	Step 2: Open the App module conf	iguration
<ul> <li>I chat-example</li> <li>I node_modules</li> <li>I pages</li> </ul>	基础配置	App模块配置 App包含的扩展模块(影响包体积)。云打包后生效。本地打包需另行在原生量
> Du static > Du TUIKit > Du unpackage	App图标配置 App启动界面配置	打包模块配置 □ Barcode(扫码)
₪ App.vue <> index.html וח main.is	App模块配置	□ Bluetooth(低功耗蓝牙)
[] package-lock.json	App <sup>校政配</sup> 直 App原生插件配置	□ Camera&Gallery(相机和相册)
[] package.json [] pages.json ]] uni.promisify.adaptor.js	App常用其它设置	☐ FaceID(人脸识别,仅支持iOS)
₽ uni.scss	WeD配置 微信小程序配置	□ FacialRecognitionVerify(实人认证) 金融级实人认证,身份证和人脸活体匹配检测 <u>开通配置</u> ,使用指南
Step 1: Open manifest.json	百度小程序配置	□ Fingerprint(指纹识别)
	抖音小程序配置 支付宝小程序配置	☐ Geolocation(定位) 定位功能依赖三方SDK,上架到国内应用市场需要在隐私协议中添加相应的条款 ☐ iBeacon
	QQ小程序配置	□ LivePusher(直播推流)
	快手小程序配置 飞书小程序配置	Maps(地图,只能选一个)     部分国产Android手机获取位置必须申请定位服务商的SDK。 <u>地图配置指南</u> 地图
	京东小程序配置	☐ Messaging(短彩邮件消息) ☐ 0Auth(登录鉴权)
	快应用配置 uni统计配置	第二方登录量权配置指南 登录 查找功能依赖二方SDK, 上架到国内应用市场需要 □ Payment(支付) 支付配置指南 支付功能依赖三方SDK, 上架到国内应用市场需要在隐私协议中语
	Source Code	<mark> Push(消息推送)</mark> ● uniPush是DCloud联合个推公司推出、全端的、云端一体的统一推送服务,使
		□ uniPush 1.0 (老版本,不推荐新项目使用) <u>配置,使用指南</u>
	Step 3: Check Push	□ uniPush 2.0 (全端支持) 配置、使用指南 nd uniPush2.0 are not checked

# Step 2: Activate the TencentCloud-TIMPush cloud packaging service and fill in the relevant parameters.

1. Go to the plugin marketplace and activate the TencentCloud-TIMPush cloud packaging service. As shown in the image:

#### Note:

1. In the plugin marketplace, the **appld** for the cloud packaging service project must match the **appid** in the project's **manifest.json**.

2. Activation of the **TencentCloud-TIMPush** cloud packaging service is for a single project only. It only concerns the project.



2. In the project's [manifest.json] > [App Native Plugin Configuration] > [Cloud Plugins], select TencentCloud-

**TIMPush** and set the relevant parameters.

#### Note:

1. Note that parameters might appear out of order in HBuilderX. Please fill them out carefully and accurately.

2. Each parameter is required; otherwise, a compilation error will occur. The default is 0.

com.hihonor.push.app\_id

com.vivo.push.app\_id && com.vivo.push.api\_key

TIMPushAppGroupID

#### Note:

com.hihonor.push.app\_id corresponds to the appID of hihonor.

When not enabling hihonor push, com.hihonor.push.app\_id can be set to 0 by default.

Console configuration	manifest.json Honor Configu



#### Note:

com.vivo.push.api\_id corresponds to vivo's appID.

com.vivo.push.api\_key corresponds to vivo's appKey.

When not enabling vivo push, com.vivo.push.api\_id and com.vivo.push.api\_key can be set to 0 by default.

Console configuration	manifest.json vivo Configura



TIMPushAppGroupID corresponds to iOS's appGroupID. It is a configuration item for iOS reach reporting, refer to configuring-app-groups to generate appGroupID.

### Note:

Not configuring TIMPushAppGroupID will not affect the normal push feature.

When iOS reach reporting is not enabled, TIMPushAppGroupID can be set to `0` by default;

You can refer to configuring-app-groups to generate appGroupID	iOS appGroupID Generation Guidelines	manifest.json iOS Configuration
	You can refer to configuring-app-groups to generate appGroupID	



# Step 3: manifest.json Android Permission Configuration

Append the following permissions in **manifest.json** > **Source View** > **app-plus** > **distribute** > **android** > **permissions**, as shown:



```
"<uses-permission android:name=\\"android.permission.INTERNET\\" />",
"<uses-permission android:name=\\"android.permission.ACCESS_NETWORK_STATE\\" />",
"<uses-permission android:name=\\"android.permission.ACCESS_WIFI_STATE\\" />",
"<uses-permission android:name=\\"android.permission.WRITE_EXTERNAL_STORAGE\\" />"
```

Y III chat-evample	manifest.json	
> In orde_modules	基础配置	1 ⊡ { 2
> in static	App图标配置	3 "appid" : " <mark></mark> ", 4 "description" : "",
Step 1: Open manifest.json	App启动界面配置	5 "versionName" : "1.0.0", 6 "versionCode" : "100",
<> index.html	App模块配置	7 "transformPx": false, /* 5+App性有相关 */
i main.js	App权限配置	9 □ "app-plus" : {
[4] manifest.json		10 "usingComponents" : true,
[] package_iock.json [] package.json	Appl示土细汁配直	11 "nvueStyleCompiler" : "uni-app", 12 "compilerVersion" : 3
[] pages.json	App常用其它设置	13 III "splashscreen" : {
uni.promisify.adaptor.js \$\vec{uni.scss}	Web配置	14"alwaysShowBeforeRender" : true,15"waiting" : true,
	微信小程序配置	16 "autoclose" : true, 17 "delay" · 0
	百度小程序配置	18 - <b>},</b> 19 /* 模块配置 */
	抖音小程序配置	20 modules": {
	支付宝小程序配置	22 - <b>},</b> 22 - <b>}</b> ,
	QQ小程序配置	23 / / 迎历众问后志 */ 24 回 "distribute": {
	快手小程序配置	25 /* android打包配置 */ 26 □ "android": {
	飞书小程序配置	27 □  "permissions" : [    28  " <uses-permission android:name='\"&lt;/td'></uses-permission>
	<京东小程序配置	29       " <uses-permission android:name='\"&lt;/td'>         30       "<uses-permission android:name='\"&lt;/td'></uses-permission></uses-permission>
	快应用配置	31       " <uses-permission android:name='\"&lt;/td'>         32       "<uses-feature android:name='\"and&lt;/td'></uses-feature></uses-permission>
	uni统计配置	33 " <uses-permission android:name='\"&lt;/th'></uses-permission>
	Source Code	35 " <uses-permission android:name-)"<="" th=""></uses-permission>
		36 " <uses-permission android:ne="" ste<="" th=""></uses-permission>
		37 " <uses-permission android:na<br="">29 "<uses-permission android:name='\"&lt;/th'></uses-permission></uses-permission>
S	tep 2: Select sou	ource code " <uses-feature android:name='\"and&lt;/th'></uses-feature>
		" <uses-permission android:name='\"&lt;/th'></uses-permission>
		41 <ul> <li>41 <ul> <li>42 <ul> <li>42 <ul> <li>42 <ul> <li>42 <ul></ul></li></ul></li></ul></li></ul></li></ul></li></ul>
		43 ' <uses-permission android:name='\"&lt;/th'></uses-permission>
		44 <pre>'<uses-permission android:name='\"&lt;/pre'></uses-permission></pre>
		46 },

# Step 4. Register TencentCloud-TIMPush

#### Note:

@tencentcloud/chat  $\geq$  3.2.5 supports TencentCloud-TIMPush.

androidConfig is the Android push configuration. If you don't need to package an Android App, you can pass null.

iOSConfig is the iOS push configuration. If you don't need to package an iOS App, you can pass null.

Retrieve the Android configuration timpush-configs.json

Retrieve iOS configuration iOSBusinessID

timpush-configs.json can be downloaded from IM console > Push > Access Settings, and placed in the same directory

as App.vue , as shown:

Application	Access settings - test-push v	Current data center: Singapore (i) Telegram group	WhatsApp group	Step 3	: Select the cor
agement Overview	Push Overview free trial Remaining 7 days All member	rs/tag push interface call frequency100 times/day Edit	Function Provide O	Function Preview O	Function Preview O
Users Groups Configuration ~	Activated  Purchase Now  free trial	<ul> <li>Activated</li> </ul>	Activated	Activated	
Nebhook Statistics n Push ^	Manufacturer configuration     M supports online and offline push notifications. Online     Android IOS     Mo Martine Martine using Info	push is delivered through the instant messaging IM chann	which is fast and reliable; for offline push, it is recommended	I that you use the system-level push channel provided by each m	inufacturer. The system The level (
Access settings Access test Push Record p 1: Access S	Add Cartificate	Edit Deixte			
Push Iroubleshooting Al Chatbot Desk	Region AppID AppSecret AppKey ChannellD Response after <b>Open application</b>				
Non Step 2: Cli	ck Android to download th	e configuration file			
lev Tools ~	2 Local deployment After purchasing the push plug-in, you can use the "qui	ck configuration" and "download configuration file" function	ns. With 3 lines of code and 2 minutes, you can complete push	access from multiple manufacturers at once.	
	Quick configuration One-click integrated configuration Average consur	nption is 1 hour	Manual configuration Need to integrate configuration one by one Average cons	umption is 1 week	



iOS iOSBusinessID can be obtained from IM console > Push > Access Settings, as shown:

Concept Cloud	Overview Products - +	
Chat	Access settings - test-push	<ul> <li>Current data center: Sir</li> </ul>
크는 Application	Push Overview free trial Remaining 7 days	All members/tag push interface call fr
Management	Normal push	All membe
은 Users 몲 Groups	Activated	Activate
Configuration Step 2	: Switch to iOS manufacturer o	onfiguration
<ul> <li>Webhook</li> <li>Statistics</li> </ul>	1 Manufacturer configuration	



App.vue

Integration (UI Included)

Integration Without UI

In App.vue, import TencentCloud-TIMPush, and mount it to uni.

In App.vue, import timpush-configs.json, and mount it to uni.





```
// #ifdef APP-PLUS
import TIMPushConfigs from "./timpush-configs.json";
const TIMPush = uni.requireNativePlugin("TencentCloud-TIMPush");
console.warn(`TencentCloud-TIMPush: uni.requireNativePlugin ${!!TIMPush ? 'success'
uni.$TIMPush = TIMPush;
uni.$TIMPushConfigs = TIMPushConfigs || {};
// #endif
```

Ensure uni.requireNativePlugin successfully imports TencentCloud-TIMPush, as shown in the figure:

10:57:00 Confirm that uni.requireNativePlugin("TencentCloud-TIMPush") succeeded barmore <u>at node mode</u> 10:57:09.204 TUIServiceManager.registerService serviceName:TUIGroupService <u>at node modules</u> 10:57:09.204 TUIServiceManager.registerService serviceName:TUIGroupService <u>at node modules</u> 10:57:09.204 TUIExtensionManager.registerExtension extensionID:chatHeader <u>at node modules</u> 10:57:09.204 TUIEventManager.registerEvent eventName:loginStateChanged subKey:userLoginSucce 10:57:09.213 <u>TencentCloud-TIMPush: uni.requireNativePlugin success at App.vue:6</u> 10:57:09.213 TUIEventManager.registerEvent eventName:loginStateChanged subKey:userLoginSucce 10:57:09.213 TUIExtensionManager.registerEvent eventName:loginD:inputToolBarMore <u>at node module</u>

When logging into uikit, register TencentCloud-TIMPush into uikit.



```
import { TUILogin } from "@tencentcloud/tui-core";
TUILogin.login({
    SDKAppID: 0, // Replace 0 with the SDKAppID of your IM app during integration.
    userID: '',
    // UserSig is the cipher for users to sign in to Instant Messaging, essentially
    // This method is only suitable for running Demo locally and debugging function
    userSig: '',
    // Should you require to send image, voice, video, file and other rich media me
    useUploadPlugin: true,
    framework: `vue${vueVersion}` // Current development uses framework vue2 / vue3
```

```
TIMPush: uni.$TIMPush, // APP registers Push Plugin
pushConfig: {
    androidConfig: uni.$TIMPushConfigs, // Android push configuration, pass nul
    iOSConfig: {
        "iOSBusinessID": "" // iOS push configuration, pass null if not needed.
        }
    }
})
```

Before logging into chat, register TencentCloud-TIMPush in the chat.







```
import TencentCloudChat from '@tencentcloud/chat';
const chat = TencentCloudChat.create({SDKAppID: 0}) // Replace 0 with the SDKAppID
chat.registerPlugin({
    'tim-push': uni.$TIMPush,
    pushConfig: {
        androidConfig: uni.$TIMPushConfigs, // Android push configuration, pass null if
        iOSConfig: {
            "iOSBusinessID": "xxx" // iOS push configuration, pass null if not needed.
        }
    }
})
```
### 🔗 Tencent Cloud

```
chat.login({
    userID: '', // User ID.
    userSig: '' // The password for logging into Chat, which is essentially the cip
})
```

#### Step 5. Create a Custom Base

click HBuilderX's Run> Run on Mobile or Emulator> Make a Custom Debugging BaseCreate a Custom Base. Note:

Configure Native Plugins, it is necessary to package a Custom Base for testing.

When creating a base, the Android package name is the application package name bound during the activation of cloud packaging for plugins.

Use cloud certificates.

	Step 7	1: Click Run			
<b>HBuilderX</b> File Edit Select Find	Goto	Run Build	d View	Tool	Help
		Browser		>	
🗸 🛄 chat-example		Run Built-in	Browser		
> 🖿 node_modules		Mobile App	Playground	>	Android Run
> 🖿 pages		Miniprogram	n	>	Android Run Pa
> 🖿 static	Step	2: Mobile A	pp Playgrou	nd	iOS Run
> 🖿 TUIKit			, , , , , , , , , , , , , , , , , , ,		iOS Run Page
> 🖿 unpackage					iOS-Simulator
App.vue					IOS-Simulator
<> index.html					Show RemoteD
🗊 main.js					Generate Cust
[•] manifest.json					Android Simula
[] package-lock.json				Step 3	3: Generate Cust
[] package.json					
[] pages.json					Faq
uni.promisify.adaptor.js					Install Simulate
🖇 uni.scss					Chrome Debug

Choose
arket
/larket
[detail]
more revenue]
p 7: Subi

# Step 6. Run and log in to the project, confirm the integration of TencentCloud-TIMPush is successful.

During runtime, choose to run with a Custom Base. In HBuilder's log, confirm the printing of

TIMPushModule.\_setToken ok , indicating successful integration of TencentCloud-TIMPush, as shown in the picture:

Confir	m th	nat Tend	centClou	Jd-TIMP	Push integration is successful
19:46.					sationList ok. completeFlag:1 count:13 cost:286 ms
19:46:20.482	Chat	19:46:21	GMT+0800	(CST).473	ProfileHandler.onConversationsProfileUpdated toAccountList:im_demo_admin
19:46:20.495	Chat	19:46:21	GMT+0800	(CST).485	ConversationModuleupdateLocalConversationList cost:1 ms
19:46:20.495	Chat	19:46:21	GMT+0800	(CST).487	UserStatusHandler.subscribeUserStatus userID count:8
19:46:20.495	Chat	19:46:21	GMT+0800	(CST).499	ConversationModulediffAndDeleteConversation list:
19:46:20.508	Chat	19:46:21	GMT+0800	(CST).502	ConversationModule.svncConversationList. count:13 sum:286 avg:286
19:46:20.508	Chat	19:46:21	GMT+0800	(CST).505	TIMPushModulesetToken ok
19:46:20.575	Chat	19:46:21	CMT:0000	(CST).577	<u>-CommonGroupHandlerpagingGet</u> GroupList ok. offset:0 limit:200 totalCount:0 isComplete
19:46:20.575	Chat	19:46:21	GMT+0800	(CST).578	CommonGroupHandler.syncGroupList ok. count:47 sum:594 avg:198
19:46:20.587	Chat	19:46:21	GMT+0800	(CST).581	CommonGroupHandler.handleUpdateGroupLastMessage conversation count:13, local group co
19:46:20.597	Chat	19:46:21	GMT+0800	(CST).591	ConversationModuleupdateLocalConversationList cost:0 ms
19:46:20.598	Chat	19:46:21	GMT+0800	(CST).592	ConversationModulediffAndDeleteConversation list:
19:46:20.598	Chat	19:46:21	GMT+0800	(CST).592	SignalingModule.onReady
19:46:20.598	Chat	19:46:21	GMT+0800	(CST).592	CommonGroupHandler.handleUpdateGroupLastMessage conversation count:13, local group co
19:46:20.598	Chat	19:46:21	GMT+0800	(CST).599	ConversationModule.emitTotalUnreadMessageCountUpdate from 0 to 3

#### Step 7. Receive your first push notification.

Go to the IM Console Push > Push Test to send your first push notification. As shown in the figure:

#### Note:

1. The receiver's application must be in the background or the process terminated.

2. Live broadcast groups do not support offline message push.

Tencent Cloud	Overview Products - +
Chat	Access test - test-push - Current data center: Singapore (i) Telegram group WhatsApp
<sup>3</sup> Step 2: Enter t	e userID of the user who will receive the push notifications.
Management	
E Overview	When an exception occurs when you connect to the native manufacturer's offline push, you can use this tool to troubleshoot.
🚊 Users	Username (UserID) 123
品 Groups	Get token binding status
Configuration ~	
🙃 Webhook	Results The information of the device currently used by 123 is as follows:
	Step 3: Get token binding status
C Otatistics	Last Updated: 2024-06-05 10:19:46
Step 1: Access Te	Certificate ID (Device Vendor): . XiaoMi
₽ Push ^	Token or regID: aTUPA*****ZMI2w; Length: 64 Last Updated: 2024-06-05 10:20:39
<ul> <li>Access Settings</li> </ul>	
Access Test	Certificate ID Mi
· Pur Step 4: Sele	t the corresponding manufacturer certificate ID
Push Data	
• Push	Send a nuch test
Troubleshooting	
O Al Chatbot	Results token: **** MI2w 成功推送。如果仍无法接收,请确认接收方手机已经打开您APP的通知功能。
	Step 5: Send a push test
Tools	
C Monitor	
📅 Dev Tools 🛛 👻	Access to the testing tool. Currently, it only supports testing whether the push is sent successfully and calling the SDK inte currently support testing jumps, ringtones and other features.
Integration Guide	

# More Advanced Features (Highly Recommended)

#### 1. Set Push Content

Integration (UI Included) Integration Without UI When using TUIChatService to send messages in uikit, set the related parameters for offlinePushInfo. For example, when sending a standard text message, the code is as follows:







```
// Send Plain Text Message
let promise = TUIChatService.sendTextMessage(
{
    payload: { text: 'Hello world!' }
},
{
    // If the recipient is offline, the message will be stored for roaming and an off
    offlinePushInfo: {
      title: '', // Offline Push Title.
      description: '', // Offline Push Content.
      extension: '', // Offline Push Transparent Content
```

}
}
);
promise.catch((error) => {
 // Business side can handle errors by catching exceptions through promise.catch
});

Reference document: uikit - TUIChatService message sending documentation When sending a message in chat, set the related fields for offlinePushInfo as follows:







```
// Message Sending Options
chat.sendMessage(message, {
    // If the recipient is offline, the message will be stored for roaming and an off
    offlinePushInfo: {
        title: '', // Offline Push Title.
        description: '', // Offline Push Content.
        extension: '', // Offline Push Transparent Content
    }
});
```

Reference Documentation:

#### Chat SDK - sendMessage Documentation

#### 2. Retrieving Click-through Transmission Content

Retrieve the transmission content in the App.vue file, and configure the specified jump page.



```
onLaunch: function () {
    // #ifdef APP-PLUS
    // In App.vue, listen in the lifecycle hook onLaunch
    if (uni.$TIMPush) {
        uni.$TIMPush.setOfflinePushListener((data) => {
            // The content within the pass-through entity, excluding the pushed Message
```

```
console.log('setOfflinePushListener', data);
    // Redirect to a specified interface within the application
    uni.navigateTo({
        url: "/pages/xxx/xxx",
        });
    });
    });
} #endif
}
```

#### Note:

Here you can retrieve the pass-through content configured during the push notification.

Here you can configure the application redirect interface.

During cross-platform interoperability, ensure that extension remains consistent, extension must include the entity field.

### Anomaly Troubleshooting

If users do not receive offline push notifications when the App is offline, they can use the Troubleshooting Tools for end-to-end troubleshooting of push details.

	Р	ush troubleshooting
	Push status	
		Task ID
		User ID:
		L
		0
		Submit instant mess 2024-05-07 1
Information		
		↓
2024-05-07 14:56:06		0
RMX3370		Submit online channel 2024-05-07 14:56:06
Android		
11		*
7.9.5000		Push message to device
rec		2024-05-07 14:56:06
0000	000100010010101000	4
oppo		() Lice alick
		ck status statistics for online push will be supported soon, so stay tuned
	Push	full link
Enable		
Bound		
	<ol> <li>The application</li> </ol>	on is switched to the background and the process is alive. Online and offline push will be ser
	Information           2024-05-07 14:56:06           RMX3370           Android           11           7.9.5666           7.9.5666           rer ' CN 60000000 - 0000           Oppo           Oppo           Enable           Bound	Push status Information 2024-05-07 14:56:06 RMX3370 Android 11 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.5666 7.9.566

# **Statistics Collection**

Query all Statistics for a user, organize and analyze daily push metric data of various types, and generate recent **Sent** > **Reached** > **Clicked** funnel conversion charts, with support for viewing by manufacturer channel.



# **Device Notification Bar Settings**

The direct manifestation of a push is a notification bar prompt, which, like other notifications, is affected by device notification-related settings, taking Huawei as an example:

"Settings - Notifications - Notifications (Lock Screen) - Hide or Do not Disturb" will affect the display of offline push notifications when the screen is locked.

"Settings - Notifications - Advanced Settings - Show Notification Icons (Status Bar)" will affect the showing of the offline push notification icon in the status bar.

"Settings - Notifications - Application Notifications - Allow Notifications" will directly affect the display of offline push notifications.

"Settings - Notifications - Application Notifications - Notification Sound" and "Settings - Notifications - Application Notifications - Notification Mute" will affect the offline push notification sound.

10:31 @ * 춗 B 🗩	10:33 <b>:</b> ④ % 穴 D 💌	10:35 🖬 💿 巻 ਵ
Settings	$\leftarrow$ Notifications & status bar $\bigcirc$	$\leftarrow$ Notifications & status bar
I Mobile network	Q Search	Q Search
Super Device >	Status bar >	SkyTone Data Service No recent notifications
More connections >	NOTIFICATIONS	Software update No recent notifications
Home screen & style	App icon badges	Tencent Cloud Media Service No recent notifications
Display & brightness	while your phone is locked. Wake screen Turp	Tencent Cloud Tools No recent notifications
Sounds & vibration	<sup>a</sup> Step 2: Enable lock screen notifications	Tencent Meeting No recent notifications
Notifications & status bar >	Most recent 💌	Step 3: Enter applies
Step 1: Notification & status bar	<mark>美団</mark> Meituan > 1 minute ago	Themes No recent notifications
Battery	maimai     >       5 minutes ago     >	Tips No recent notifications

### Vendor's push restrictions

1. All vendors in China have adopted message classification mechanisms, and different push policies are assigned for different types of messages. To make the push timely and reliable, you need to set the push message type of your app as the system message or important message with a high priority based on the vendor's rules. Otherwise, offline push messages are affected by the vendor's push message classification and may vary from your expectations.

2. Additionally, some manufacturers impose limits on the daily number of pushes for an application. You can check the daily push quantity limits in the Manufacturer Console. If offline push messaging is not timely or is occasionally not received, consider the following:

### 🔗 Tencent Cloud

Huawei	viv	OPPO	Mi	Meizu	FCM
Push messages are divided into	Push messages	Push messages are	Push messages are	The number of push	The frequency of push uplink
Service and	are divided	divided into	divided into	messages is	messages is
Communication	into system	private	Important	limited. For	limited. For
Category and	messages	messages and	messages and	more details,	details, please
Information	and	public trust	general	please see	see
Marketing	operational	messages,	messages,	Manufacturer	Manufacturer
Category, with	messages,	with different	with different	Description.	Description.
different push	with different	pusn	pusn		
effectiveness	push	effectiveness	effectiveness		
and policies.	effectiveness	and policies.	and policies.		
Additionally,	and policies.	Private	The important		
message	System	messages are	message type		
classification is	messages	for users who	is reserved		
also related to	will also	are	exclusively for		
Self-	undergo a	significantly	instant .		
categorization	secondary	attentive to	messaging,		
benefits:	correction by	and wish to	personal		
Kales a la sa	the vendor's	receive	tollow-up		
If there is no	Intelligent	Information	dynamic		
seit-neip	classification.	promptiy.	alerts,		
message	If a message	Privileges for	personal		
classification	Is identified	the private	matters		
permission, the	not as a	message	reminders,		
	system	channel need	personal order		
periorm	message, it		status		
secondary	WIII	for via email.	changes,		
Intelligent	automatically	The number of	personal		
message	be corrected	pusnes	Tinancial		
classification on	to an	through the	reminders,		
push messages.	operational	public trust	personal		
If you have	message. If	channel Is	status		
applied for sell-	there is a	limited. For	changes,		
calegorization	misjudgment,	more details,	personal		
benefits, push	you can	please see	resource		
messages will	provide	Manufacturer	changes, and		
			personal		
		UI Monufacturer	uevice alerts.		
your own	Auditionally,		triese eight		
		Description 2.	types of		
categories for			messages can		
pushing. For	messages is		be activated		
more details,	restricted by		through an		

please see Manufacturer	a daily limit, which is determined	application in the manufacturer's
Decemption	by the app's	console. The
	subscription	number of
	statistics with	general
	the vendor.	message type
	For more	pushes is
	details,	limited. For
	please see	more details,
	Manufacturer	please see
	Description 1	Manufacturer
	or	Description 1
	Manufacturer	or
	Description	Manufacturer
	2.	Description 2.

### FAQs

# 1. TencentCloud-TIMPush and uniPush2 are in conflict and cannot be shared. How should this be addressed?

Reason: TencentCloud-TIMPush does not support sharing with other offline push channels. Solution: It is recommended to use only TencentCloud-TIMPush.

# 2. Error in push: "TIMPushModule.\_getDeviceToken failed. error: { "code": -1, "msg": "huawei ApiException: com.huawei.hms.common.ApiException: 907135000: arguments invalid"}".

Reason: The Huawei agconnect-services.json file has not been imported. Solution: Check if the Huawei agconnect-services.json file has been imported.

#### Note:

Huawei Push requires the agconnect-services.json file downloaded from the official website to be placed in the

nativeResources/android/assets directory. For details, see uniapp.

# 3. Error in push: "TIMPushModule.\_getDeviceToken failed. error:{"code":-1, "msg": "callback is not String"}".

Reason: Huawei version is too low, EMUI version needs to be greater than 10.

Solution: Upgrade or switch to a Huawei with EMUI version > 10 for push.

# 4. Error in push: "TIMPushModule.\_getDeviceToken failed. error:{"code":22022, "msg": "xiaomi error code: 22022"}".

Reason: The application package name is invalid.

Solution: Go to the Xiaomi Push Platform to check if the application's package name, appld, and appKey match.

#### 5. Can't receive push notifications on OPPO mobile phones?

Cause: The notification bar display feature is disabled by default for applications installed on the OPPO device. Solution: Enable the notification bar display feature.

#### 6. Followed the integration process but still can't receive push notifications?

Cause:

Device status anomaly, IM console settings not configured, initialization not registered, etc.

Push is dependent on vendor capabilities; some simple characters may be filtered by the vendor and not be transparently pushed.

If push messages are not sent promptly or are occasionally not received, you need to check the vendor's push restrictions.

Solution:

Investigation can be conducted using the Troubleshooting Tools.

Check the content to be sent, avoid using simple characters.

Check the manufacturer's push restrictions, details can be seen in Manufacturer Push Restrictions.

#### 7. Why aren't regular messages receiving offline push notifications on iOS?

Reason: The inconsistency between the app's running environment and the certificate environment. Solution:

Check if the app's running environment matches the certificate environment. If they do not match, offline push notifications will not be received.

Check whether the app and certificate environment is set to production. If it's in the development environment, requesting a deviceToken from Apple might fail, but this issue hasn't been observed in the production environment, please switch to the production environment for testing.

# 8. In the iOS development environment, registration occasionally does not return a deviceToken or indicates an APNs request token failure?

Cause: This issue arises due to instability in the APNs service.

Solution:

Insert a SIM card into the phone and use the 4G network.

Uninstall and reinstall the application, restart the application, or shut down and restart the phone.

Use a package for the production environment.

Use another iPhone.

#### 9. Why is there no token in iOS?



Cause:

Emulators do not generate tokens.

Real Device, you need to enable push permissions on the phone.

Real Device, you need to add push notification's entitlement.

Solution: Use a real device to enable push permissions, and add push notification's entitlement.

#### 10. Can tim-js-sdk use TIMPush?

Cause: Does not support TencentCloud-TIMPush;

Solution: Upgrade @tencentcloud/chatAccess, @tencentcloud/chat is downward compatible, and will not affect the business implemented by tim-js-sdk.

# Flutter

Last updated : 2024-06-13 10:21:45

# Operation step

#### Step 1: Integrate message push plugin

This plugin is known as tencent\_cloud\_chat\_push on pub.dev. You can include it in your pubspec.yaml dependency directory, or run the following command for automatic installation.





flutter pub add tencent\_cloud\_chat\_push

#### Step 2: Push parameter configuration

iOS

Android

Please upload the iOS APNs push certificate obtained during the vendor configuration step to the IM console.

The IM console will assign a certificate ID for you, as shown in the figure below:

Tencent Cloud	Overview Products Cloud Security Center Video on Demand Media Processing Service Anti-DE	DoS Chat Short Message Service
Chat	Access settings Current data center: Jakarta () Telegram gr	roup WhatsApp group
∃≟ Application		
Management	ID: 1( 5 Edit Delete	ID: 15
🔠 Overview	Certificate information 1700191466613.ir	Certificate information 1700191482842
은 Users	F%E4%BA%A7%E7%8E%AF%E5%A2%83,p12	1%E7%8E%AF Push Type APNs Push
器 Groups	Push Type APNs Push	Certificate Type Development Er
Configuration ·	Please test after the Release package is released from	Please run the tr push), and do n
🕸 Webhook	mutable-content Enabled	mutable-content Enabled
C Statistics	Certificate password	Expiration time 2024-12-05 07:
Pluain	Expiration time 2024-05-22 02:37:36	
E Push _^		
Access settings	ID: \ Edit Delete	ID: 1

#### As early as possible after your application starts, call the

TencentCloudChatPush().setApnsCertificateID method to pass in the certificate ID.





#### TencentCloudChatPush().setApnsCertificateID(apnsCertificateID: Certificate ID);

After completing the manufacturer push information on the console, download and add the configuration file to the project. Add the downloaded timpush-configs.json file to the android/app/src/main/assets directory. If the directory does not exist, please create it manually.

1. Choose to download the configuration file timpush-configs.json	1. Add to the project



#### **Step 3: Client Code Configuration**

In this step, you'll need to write some native code, such as: Swift, Java, XML, etc.

Please don't worry, just follow the instructions and copy the code we provide into the specified file.

iOS

Android

You can use Xcode for editing, or you can directly edit in Visual Studio Code or Android Studio.

Open the ios/Runner/AppDelegate.swift file, paste the following outlined code into it, as shown in the figure. The code is attached after the picture.









```
import UIKit
import Flutter
// Add these two import lines
import TIMPush
import tencent_cloud_chat_push
// Add `, TIMPushDelegate` to the following line
@UIApplicationMain
@objc class AppDelegate: FlutterAppDelegate, TIMPushDelegate {
    override func application(
```

}

```
_ application: UIApplication,
    didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKe
) -> Bool {
    GeneratedPluginRegistrant.register(with: self)
    return super.application(application, didFinishLaunchingWithOptions: launch
}
// Add this function
func offlinePushCertificateID() -> Int32 {
    return TencentCloudChatPushFlutterModal.shared.offlinePushCertificateID();
}
// Add this function
func applicationGroupID() -> String {
    return TencentCloudChatPushFlutterModal.shared.applicationGroupID()
}
// Add this function
func onRemoteNotificationReceived(_ notice: String?) -> Bool {
    TencentCloudChatPushPlugin.shared.tryNotifyDartOnNotificationClickEvent(not
    return true
}
```

It is recommended to use Android Studio to complete this part of the editing.

In the same directory as MainActivity under your project's Android path, create a new Application file class, for example, it can be named MyApplication.java .

If you have already defined an Application class, you can directly reuse it without having to create it again.



Paste the following code into the file, as shown above:





```
Replace 'package' with your own, usually Android Studio will generate it automatica
import com.tencent.chat.flutter.push.tencent_cloud_chat_push.application.TencentClo
public class MyApplication extends TencentCloudChatPushApplication {
    @Override
    public void onCreate() {
        super.onCreate();
    }
}
```

#### Note:

If you have already created your own Application for other purposes, please directly extend TencentCloudChatPushApplication and ensure in the onCreate() function, you call super.onCreate(); .

**Open the** android/app/src/main/AndroidManifest.xml **file, add a specified** android:name

parameter for the <application> tag, pointing to the newly made custom Application class. As shown in the figure:



Step 4: Client OEM Configuration

iOS

Android

No need to proceed with this step on the iOS side.

Open the android/app/build.gradle file, at the end, add a dependencies configuration, and as required, include all or part of the following vendor push packages. Only by including the corresponding vendor's push package can you enable that vendor's Native Push Capability.

The version numbers mentioned below should be consistent with the version number of this Flutter Push Plugin (tencent\_cloud\_chat\_push).





```
dependencies {
    // Huawei
    implementation 'com.tencent.timpush:huawei:${Push Plugin version number}'
    // XiaoMi
    implementation 'com.tencent.timpush:xiaomi:${Push Plugin version number}'
    // vivo
    implementation 'com.tencent.timpush:vivo:${Push Plugin version number}'
    // Honor
```

```
implementation 'com.tencent.timpush:honor:${Push Plugin version number}'
// Meizu
implementation 'com.tencent.timpush:meizu:${Push Plugin version number}'
// Google Firebase Cloud Messaging (Google FCM)
implementation 'com.tencent.timpush:fcm:${Push Plugin version number}'
// Choose one of the two below for OPPO
// For the China Region, choose to integrate this package
implementation 'com.tencent.timpush:oppo:${Push Plugin version number}'
// For other regions, choose to integrate this package
implementation 'com.tencent.timpush:oppo-intl:${Push Plugin version number}'
```

#### Vivo and Honor Adaptation

According to Vivo and Honor Vendor Access Guide, it is necessary to add APPID and APPKEY to the Manifest File.

Method 1

}

Method 2









// android/app/src/main/AndroidManifest.xml

```
// Vivo begin
<meta-data tools:replace="android:value"
    android:name="com.vivo.push.api_key"
    android:value="`APPKEY` of the certificate assigned to your application" />
<meta-data tools:replace="android:value"
    android:name="com.vivo.push.app_id"</pre>
```

```
android:value="`APPID` of the certificate assigned to your application" />
// Vivo end
// Honor begin
<meta-data tools:replace="android:value"
    android:name="com.hihonor.push.app_id"
    android:value="`APPID` of the certificate assigned to your application" />
// Honor end
```

#### Adapting to Huawei, Honor, and Google FCM

Follow the vendor's method to integrate the corresponding plugin and JSON configuration file.

#### Note:

The following adaptation for Honor is only needed for version 7.7.5283 and above.

1.1 Download the configuration file and add it to the root directory of the project/Android/app.

Huawei

HONOR

Google FCM

**Operation Path** 



HONOR Developers	
⊘ 生态服务 ∧	生态服务 > 应用管理 > <b>应用基础信息查看</b>
应用服务	应用基础信息
游戏服务	应用基础信息做任何更改,将在提交保存后生效
智慧服务	
内容服务	应用名称:
应用管理	更新时间:
	App ID:
HONOR Connect	应用包名: 🗩 🖬 🖬 📰 📕 👘
☆ 开放能力 ∨	平台类型: 安卓
● 开发者中心 ~	✓ 应用类型: 应用
	支持设备: 手机/平板
	默认语言: 简体中文 🧷
	SDK 配置: 下载最新的配置文件(如果您修改了应用信息或者更改了某个开发服务设置,可能需要更新该文件) 」 mcs-services.json     添加 SDK

붣 Firebase		Tencent-IM 👻 项目设置			
↑ 项目概览	¢		您的应用		
构建					添加应用
Authentication     Firestore Dat     Realtime Dat     Storage     Hosting     () Functions	on abase abase		Android 应用	SDK 设置和配置         需要重新为您的应用配置 Firebase SDK? 请再次访问 SDK 设置说明,或直接下载:         10 宣看 SDK 说明         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・         ・ </th <th>包含应用密钥</th>	包含应用密钥
<ul> <li>Wachine Lea</li> <li>发布与监控</li> <li>Crashlytics</li> <li>② Performance</li> <li>☑ Test Lab</li> <li>.10 App Distribut</li> </ul>	rning ;			広用別名 広用別名 添加別名 ✓ SHA 正书指纹 ②	au O
分析 I Dashboard (○ Realtime				添加指纹 移時此应用	
Extensions					



1.2 Add the following configuration under buildscript -> dependencies in the project-level build.gradle file:

#### Gradle 7.1 and above

Add the following configuration under buildscript -> dependencies in the project-level build.gradle file:




```
buildscript {
    dependencies {
        ...
        classpath 'com.huawei.agconnect:agcp:1.6.0.300'
        classpath 'com.hihonor.mcs:asplugin:2.0.1.300'
        classpath 'com.google.gms:google-services:4.4.0'
    }
}
```

Add the following repository configuration under buildscript -> repositories and allprojects -> repositories in the project-level settings.gradle file:



```
pluginManagementbuildscript {
    repositories {
        gradlePluginPortal()
        mavenCentral()
        maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
        // Configure the Maven repository address for the HMS Core SDK.
```

## Step 5: Handle the message click callback, and parse the parameters

Please define a function to receive the push message click callback event.

Define the function with the following parameters: {required String ext, String? userID, String? groupID} .

Among them, the ext Field carries the complete ext information specified by the sender. If not specified, a default value is assigned. You can navigate to the corresponding page by parsing this field.

The userID and groupID fields are automatically attempted by the plugin to parse the ext Json String, retrieving the single chat partner userID and group chat groupID information. If you have not defined the ext Field yourself, and the ext Field is default specified by the SDK or UIKit, then you can use the default parsing provided here. If parsing fails, it will be null.

You can define a function to receive this callback and navigate to the corresponding session page or your business page accordingly.

Example below:



```
void _onNotificationClicked({required String ext, String? userID, String? groupID})
print("_onNotificationClicked: $ext, userID: $userID, groupID: $groupID");
if (userID != null || groupID != null) {
    // Navigate to the corresponding Message Page based on userID or groupID.
    } else {
    // Write your own parsing method based on the ext Field and navigate to the cor
    }
}
```

#### Step 6: Register Push Plugin

Please register the push plugin immediately after logging into IM and before using other plugins (such as CallKit).

Invoke the TencentCloudChatPush().registerPush method, passing in a callback function defined for clicks.

Furthermore, you have the option to also pass in appsCertificateID for the iOS push certificate ID and androidPushOEMConfig for the Android push vendor configuration. These two configurations should have been specified in previous steps, and if no modification is necessary, they do not need to be passed again.



TencentCloudChatPush().registerPush(onNotificationClicked: \_onNotificationClicked);

#### Note:

If your application requires the use of **push plugin for business message notifications**, and it does not immediately start and log in to the IM module after launching <3>, or if you need to handle business navigation by obtaining message click callbacks before logging in to the IM module <5>, it is recommended that you call the <7>TencentCloudChatPush().registerOnNotificationClickedEvent</7> method as soon as possible to manually mount the message click callback, so that you can promptly obtain the message parameters.

In this scenario, you can execute this function before calling TencentCloudChatPush().registerPush and place it as early as possible in the code.



TencentCloudChatPush().registerOnNotificationClickedEvent(onNotificationClicked: \_o

# Step 7: Message Push Delivery Statistics

If you need to collect data on delivery, please complete the setup as follows:

Huawei
HONOR
vivo
Meizu
iOS



Receipt Address: https://api.im.qcloud.com/v3/offline\_push\_report/huawei

#### Note:

Huawei Push Certificate ID <= 11344, using Huawei Push v2 version interface does not support reach and click receipt, please regenerate and update the certificate ID.

но	NOF	R Develop	ers					文档	智能客服	158****40
0		开放能力 / 推送	服务 / 应用回执							
රි			应用类型:移动应用 应用名称: ••••••••••••••••••••••••••••••••••••	回执配置	aitanta	4/50	×			
	>		应用回执:测试回执 编辑 删除	* 回调地址: 回调用户名:	源和LUEI外 https://test.tim.qq.com/v3/offline_push_report/honor 只能输入英文字母、数字和下划线	0/50				
				回调密钥: * 回执范围:	<ul><li>长度不超过 128, 不小于 16 字符</li><li>✓ 到达回执 </li><li>✓ 点击回执</li></ul>					
					取消 機交					

Receipt Address: https://api.im.qcloud.com/v3/offline\_push\_report/honor

Callback Address (	Receipt	ID Config	guration i	
マママ チャック マンマン	→ 推送运营平台			
88 云通信M ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	应用信息		添加Android证	书
<ul><li></li></ul>	应用名称:		AppKey • AppID •	请输入Appŀ 请输入AppⅡ
应用信息 测试设备	审核状态: 已通过 创建时间:		回执 ID	请输入
◇ 标签管理 ♀。在线诊断	Applo: Wellemann AppKey: :::::::::::::::::::::::::::::::::::		AppSecret •	请输入 AppS
	APP回执地址: 已开通 修改 关闭		点击后续动作	● 打开应用
Receipt Address: https://api.i	m.qcloud.com/v3/offline_push_report/vivo			

Enable Receipt Switch	Configure Receipt Addres

<b>浴加施肤</b> 证节				×	新建同执 每个00000
应用包名称★	请输入应用包名称	如何生成魅族证书? 🖸	l		<ul> <li>② 回执地址</li> </ul>
AppID •	请输入AppID				回执列表
АррКеу •	请输入AppKey				回换地址
				-	
回执开关	① ① 打开回执开关后,请务必参照文档	2,正确配置回执地址,否则	则将会导致离线推送失败。		
回执开关 AppSecret *	①     ①       打开回执开关后,请务必参照文档       请输入AppSecret	<b>记</b> ,正确配置回执地址,否则	则将会导致离线推送失败。		
回执开关 AppSecret • 点击后续动作	①     ①       打开回执开关后,请务必参照文档       请输入AppSecret       〕打开应用     打开网页	2 , 正确配置回执地址, 否约 打开应用内指定页面	则将会导致离线推送失败。	J	
回执开关 AppSecret • 点击后续动作 应用内指定界面 •	<ul> <li>③</li> <li>打开回执开关后,请务必参照文档</li> <li>请输入AppSecret</li> <li>打开应用</li> <li>打开网页</li> <li>com.tencent.qcloud.tim.push.TIM</li> <li>此处会回调点击通知栏事件,通过//</li> </ul>	2,正确配置回执地址,否则 打开应用内指定页面 / 播或者回调给到应用,App	则将会导致高线推送失败。	应用	

Receipt Address: https://api.im.qcloud.com/v3/offline\_push\_report/meizu

#### Note:

After enabling the Receipt Switch, please make sure the Receipt Address is configured correctly. Failing to configure or configuring the wrong address will affect the push feature.

For iOS Push Notification Reach Statistics Configuration, please refer to Statistics Push Arrival Rate.

No configuration is needed for other supported manufacturers; FCM does not support the push notification statistics feature.

Congratulations, you have completed the integration of the push plugin. A reminder: After the trial or purchase of the push plugin expires, the push service will automatically stop (including regular message offline push, all-member/Tag push, etc.). To avoid affecting the normal use of your business, please purchase/ renew in advance.

# **React-Native**

Last updated : 2024-06-24 15:47:33

# Operation step

## Step 1: Integrate message push plugin

To install the react-native-tim-push package, execute the following code in the terminal.



# ठ Tencent Cloud

```
# use yarn
yarn add react-native-tim-push
# use npm
npm install react-native-tim-push
```

### Step 2: Push parameter configuration

iOS

Android

Please upload the iOS APNs push certificate obtained during the vendor configuration step to the IM console. The IM console will assign a certificate ID for you, as shown in the figure below:



As early as possible after your application starts, call the

TimPushPlugin.getInstance().setApnsCertificateID method to pass in the certificate ID.





```
import { TimPushPlugin } from 'react-native-tim-push';
const certificateID = 'certificate id';
TimPushPlugin.getInstance().setApnsCertificateID(certificateID);
```

After completing the manufacturer push information on the console, download and add the configuration file to the project. Add the downloaded timpush-configs.json file to the android/app/src/main/assets directory. If the directory does not exist, please create it manually.

1. Choose to download the configuration file timpush-configs.json	1. Add to the project



#### **Step 3: Client Code Configuration**

In this step, you'll need to write some native code, such as: Swift, Java, etc.

Please don't worry, just follow the instructions and copy the code we provide into the specified file.

iOS

Android

You can use Xcode for editing, or you can directly edit in Visual Studio Code or Android Studio.

1. Create a file named "TencentIMPush.swift" in the iOS directory and copy the following code into it.





```
import Foundation
import react_native_tim_push
@objc class TencentImPush: NSObject{
    @objc func getOfflinePushCertificatedID() -> Int32 {
    return TencentCloudPushModal.shared.offlinePushCertificateID();
  }
```

```
@objc func getApplicationGroupID() -> String {
   return TencentCloudPushModal.shared.applicationGroupID();
  }
  @objc func onRemoteNotificationReceived(_ notice: String?) -> Void {
   TencentCloudPushModal.shared.onRemoteNotificationReceived(notice);
  }
}
```

2. Locate the "AppDelegate.h" file and add the following code to it.





```
#import <Your-Project-Name-Swift.h>
// My project Name is `TimPushExample`. So it should be `#import <TimPushExample-Sw
...</pre>
```

After adding the code, it should look like this:

1	<pre>#import <rctappdelegate.h></rctappdelegate.h></pre>
	<pre>#import <uikit uikit.h=""></uikit></pre>
3+	<pre>#import <timpushexample-swift.h></timpushexample-swift.h></pre>
	@interface AppDelegate : RCTAppDelegate
	@end
8	

3. Locate the "AppDelegate.mm" file and add the following code to it.





```
- (int)offlinePushCertificateID {
   TencentImPush *instance = [[TencentImPush alloc] init];
   return [instance getOfflinePushCertificatedID];
}
- (NSString *)applicationGroupID {
   TencentImPush *instance = [[TencentImPush alloc] init];
   return [instance getApplicationGroupID];
}
```

```
- (BOOL)onRemoteNotificationReceived:(NSString *)notice {
    TencentImPush *instance = [[TencentImPush alloc] init];
    [instance onRemoteNotificationReceived:notice];
    return YES;
}
```

After adding the code, it should look like this:

```
#import "AppDelegate.h"
   #import <React/RCTBundleURLProvider.h>
  @implementation AppDelegate

    - (BOOL)application: (UIApplication *)application didFinishLaunchingWithOptic

     self.moduleName = @"TimPushExample";
    // They will be passed down to the ViewController used by React Native.
    self.initialProps = @{};
     return [super application:application didFinishLaunchingWithOptions:launch
   – (NSURL *)sourceURLForBridge:(RCTBridge *)bridge
   {
     return [self getBundleURL];
2+ - (int)offlinePushCertificateID {
       TencentImPush *instance = [[TencentImPush alloc] init];
       return [instance getOfflinePushCertificatedID];
+ }
 + - (NSString *)applicationGroupID {
      TencentImPush *instance = [[TencentImPush alloc] init];
       return [instance getApplicationGroupID];
 +
 + - (BOOL) on Remote Notification Received: (NSString *) notice {
       TencentImPush *instance = [[TencentImPush alloc] init];
       [instance onRemoteNotificationReceived:notice];
```



In the "MainApplication.java" file, import "com.timpush.RNTencentIMPushApplication" and replace "Application" with "RNTencentIMPushApplication" as shown in the following example:



# Step 4: Client OEM Configuration

iOS

Android

No need to proceed with this step on the iOS side.

Open the android/app/build.gradle file, at the end, add a dependencies configuration, and as required, include all or part of the following vendor push packages. Only by including the corresponding vendor's push package can you enable that vendor's Native Push Capability.

The version numbers mentioned below should be consistent with the version number of this React Native Push Plugin (react-native-tim-push).



```
dependencies {
    // Huawei
    implementation 'com.tencent.timpush:huawei:${Push Plugin version number}'
```

# S Tencent Cloud

```
// XiaoMi
implementation 'com.tencent.timpush:xiaomi:${Push Plugin version number}'
// vivo
implementation 'com.tencent.timpush:vivo:${Push Plugin version number}'
// Honor
implementation 'com.tencent.timpush:honor:${Push Plugin version number}'
// Meizu
implementation 'com.tencent.timpush:meizu:${Push Plugin version number}'
// Google Firebase Cloud Messaging (Google FCM)
implementation 'com.tencent.timpush:fcm:${Push Plugin version number}'
// Choose one of the two below for OPP0
// For the China Region, choose to integrate this package
implementation 'com.tencent.timpush:oppo:${Push Plugin version number}'
// For other regions, choose to integrate this package
implementation 'com.tencent.timpush:oppo-intl:${Push Plugin version number}'
```

#### Vivo and Honor Adaptation

According to Vivo and Honor Vendor Access Guide, it is necessary to add APPID and APPKEY to the Manifest File. Method 1

Method 2

}









// android/app/src/main/AndroidManifest.xml

```
// Vivo begin
<meta-data tools:replace="android:value"
    android:name="com.vivo.push.api_key"
    android:value="`APPKEY` of the certificate assigned to your application" />
<meta-data tools:replace="android:value"
    android:name="com.vivo.push.app_id"</pre>
```

```
android:value="`APPID` of the certificate assigned to your application" />
// Vivo end
// Honor begin
<meta-data tools:replace="android:value"
    android:name="com.hihonor.push.app_id"
    android:value="`APPID` of the certificate assigned to your application" />
// Honor end
```

# Adapting to Huawei, Honor, and Google FCM

Follow the vendor's method to integrate the corresponding plugin and JSON configuration file.

#### Note:

The following adaptation for Honor is only needed for version 7.7.5283 and above.

1.1 Download the configuration file and add it to the root directory of the project/Android/app.

Huawei

HONOR

Google FCM

**Operation Path** 



HONOR Developers	
⊘ 生态服务 ∧	生态服务 > 应用管理 > <b>应用基础信息查看</b>
应用服务	应用基础信息
游戏服务	应用基础信息做任何更改,将在提交保存后生效
智慧服务	
内容服务	应用名称:
应用管理	更新时间:
	App ID:
HONOR Connect	应用包名: 🗩 🖛 🗰 🗰 👘 💼
☆ 开放能力 ∨	平台类型: 安卓
♀ 开发者中心 ∨	< 应用类型: 应用
	支持设备: 手机/平板
	默认语言: 简体中文 🖉
	SDK 配置: 下载最新的配置文件(如果您修改了应用信息或者更改了某个开发服务设置,可能需要更新该文件) 山 mcs-services.json

붣 Firebase	Tencent-IM 👻 项目设置			
✿ 项目概览		您的应用		
构建			添加应用	
Authentication     Firestore Database     Realtime Database     Storage     Mosting     Hosting     Functions		Android 应用	SDK 设置和配置 需要重新为您的应用配置 Firebase SDK? 请再次访问 SDK 设置说明,或直接下载包含应用密钥 和标识符的配置文件。 2 查看 SDK 说明	
<ul> <li>(2) Machine Learning</li> <li>发布与监控</li> <li>Crashlytics</li> <li>② Performance</li> <li>☑ Test Lab</li> <li>④ App Distribution</li> </ul>			M H B (0) 应用别名 添加别名 / 软件包名称	
分析 .1 Dashboard 〇 Realtime			SUR LL ITHEN 0 深加指纹 移陸此应用	
Extensions				



1.2 Add the following configuration under buildscript -> dependencies in the project-level build.gradle file:

For Gradle version 7.1 and above

Gradle version 7.0

Versions Below Gradle 7.0

Add the following configuration under buildscript -> dependencies in your project-level build.gradle file:





```
buildscript {
    dependencies {
        ...
        classpath 'com.google.gms:google-services:4.3.15'
        classpath 'com.huawei.agconnect:agcp:1.4.1.300'
        classpath 'com.hihonor.mcs:asplugin:2.0.1.300'
    }
}
```

In the project-level settings.gradle file, add the following repository configurations under pluginManagement -> repositories and dependencyResolutionManagement -> repositories:



```
pluginManagement {
    repositories {
        gradlePluginPortal()
        mavenCentral()
        maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
        // Configure the Maven repository address for HMS Core SDK.
        maven {url 'https://developer.huawei.com/repo/'}
        maven {url 'https://developer.hihonor.com/repo'}
```

```
}
}
dependencyResolutionManagement {
    ....
    repositories {
        mavenCentral()
        maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
        // Configure the Maven repository address for HMS Core SDK.
        maven {url 'https://developer.huawei.com/repo/'}
        maven {url 'https://developer.hihonor.com/repo'}
    }
}
```

Add the following configuration under buildscript in your project-level build.gradle file:





```
buildscript {
    repositories {
        mavenCentral()
maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
// Configure the Maven repository address for HMS Core SDK.
maven {url 'https://developer.huawei.com/repo''}
maven {url 'https://developer.hihonor.com/repo'}
    }
    dependencies {
        ...
        classpath 'com.google.gms:google-services:4.3.15'
```

```
classpath 'com.huawei.agconnect:agcp:1.4.1.300'
classpath 'com.hihonor.mcs:asplugin:2.0.1.300'
}
```

Add the following repository configurations under dependencyResolutionManagement -> repositories in your projectlevel settings.gradle file:



```
dependencyResolutionManagement {
```

```
...
repositories {
```

```
mavenCentral()
maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
// Configure the Maven repository address for HMS Core SDK.
maven {url 'https://developer.huawei.com/repo/'}
maven {url 'https://developer.hihonor.com/repo'}
}
```

Add the following configuration under buildscript and allprojects in the project-level build.gradle file:



```
buildscript {
    repositories {
        mavenCentral()
maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
// Configure the Maven repository address for HMS Core SDK.
maven {url 'https://developer.huawei.com/repo/'}
maven {url 'https://developer.hihonor.com/repo'}
    }
    dependencies {
        . . .
        classpath 'com.google.gms:google-services:4.3.15'
        classpath 'com.huawei.agconnect:agcp:1.4.1.300'
        classpath 'com.hihonor.mcs:asplugin:2.0.1.300'
    }
}
allprojects {
    repositories {
        mavenCentral()
maven { url "https://mirrors.tencent.com/nexus/repository/maven-public/" }
// Configure the Maven repository address for HMS Core SDK.
maven {url 'https://developer.huawei.com/repo/'}
maven {url 'https://developer.hihonor.com/repo'}
    }
}
```

1.3 Add the following configuration in the app-level build.gradle file:





apply plugin: 'com.google.gms.google-services'
apply plugin: 'com.huawei.agconnect'
apply plugin: 'com.hihonor.mcs.asplugin'

#### Step 5: Handle the message click callback, and parse the parameters

Please define a function to receive the push message click callback event.

```
Define the function with the following parameters: (ext: string, userID?: string, groupID?:
string): void; .
```

Among them, the ext Field carries the complete ext information specified by the sender. If not specified, a default value is assigned. You can navigate to the corresponding page by parsing this field.

The userID and groupID fields are automatically attempted by the plugin to parse the ext Json String, retrieving the single chat partner userID and group chat groupID information. If you have not defined the ext Field yourself, and the ext Field is default specified by the SDK or UIKit, then you can use the default parsing provided here. If parsing fails, it will be null.

You can define a function to receive this callback and navigate to the corresponding session page or your business page accordingly.

## Step 6: Register Push Plugin

# Please register the push plugin immediately after logging into IM and before using other plugins (such as CallKit).

Invoke the TimPushPlugin.getInstance().registerPush method, passing in a callback function defined for clicks.

Furthermore, you have the option to also pass in appsCertificateID for the iOS push certificate ID and androidPushOEMConfig for the Android push vendor configuration. These two configurations should have been specified in previous steps, and if no modification is necessary, they do not need to be passed again.





#### Note:

If your application requires the use of **push plugin for business message notifications**, and it does not immediately start and log in to the IM module after launching, or if you need to handle business navigation by obtaining message click callbacks before logging in to the IM module, it is recommended that you call
the TimPushPlugin.getInstance().registerOnNotificationClickedEvent method as soon as possible to manually mount the message click callback, so that you can promptly obtain the message parameters. In this scenario, you can execute this function before calling

TimPushPlugin.getInstance().registerPush and place it as early as possible in the code.



TimPushPlugin.getInstance().registerOnNotificationClickedEvent({onNotificationClick

### Step 7: Message Push Delivery Statistics

If you need to collect data on delivery, please complete the setup as follows:



Huawei	
HONOR	
vivo	
Meizu	
iOS	



Receipt Address: https://api.im.qcloud.com/v3/offline\_push\_report/huawei

### Note:

Huawei Push Certificate ID <= 11344, using Huawei Push v2 version interface does not support reach and click receipt, please regenerate and update the certificate ID.

но	NOF	R Develope	ers					文	档	智能客服	158*****4
0 0		开放能力 / 推送的 【 <b>应用回执</b>	服务 / 应用回执								
S S S S S S S S S S S S S S S S S S S			应用类型:移动应用 应用名称: ••••••••••••••••••••••••••••••••••••	回执配置			×				
	>		应用回执:测试回执 编辑 删除	* 回訊名称: * 回调地址: 回调用户名:	测试回热 https://test.tim.qq.com/v3/offline_push_report/honor 只能输入英文字母、数字和下划线	4/50					
				回调密钥: * 回执范围:	长度不超过 128, 不小于 16 字符 ☑ 到达回执   ☑ 点击回执						
					取消 提交						

Receipt Address: https://api.im.qcloud.com/v3/offline\_push\_report/honor

Callback Address Configuration	Receipt ID Configuration i
↓ ↓ ◇ 开放平台 推送运営平台	
88 云通信M · 应用信息	添加Android证书
① 推送規计	AppKey・     请输入AppK       AppID・     请输入AppII       回执 ID     请输入       Category     请输入 Cate       AppSecret・     请输入AppS       点击后续动作     打开应用
Receipt Address: https://api.im.qcloud.com/v3/offline_push_report/vivo	

Enable Receipt Switch	Configure Receipt Addres

浴加施肤证节				×	新建回执 每个业务网友
应用包名称 *	请输入应用包名称	如何生成魅族证书? 🛽			② 回执地址
AppID •	请输入AppID				回执列表
АррКеу •	请输入AppKey				回换地址
•					
回执开关	① ① 打开回执开关后,请务必参照文档	<b>2</b> ,正确配置回执地址,否则:	将会导致离线推送失败。		
回执开关 AppSecret *	<ul> <li>①</li> <li>打开回执开关后,请务必参照文档</li> <li>请输入AppSecret</li> </ul>	<b>2</b> ,正确配置回执地址,否则 <sup>3</sup>	将会导致离线推送失败。		
回执开关 AppSecret • 点击后续动作	<ul> <li>①</li> <li>打开回执开关后,请务必参照文档</li> <li>请输入AppSecret</li> <li>〕打开应用</li> <li>〕打开应用</li> <li>〕打开网页</li> </ul>	2 , 正确配置回执地址,否则 打开应用内指定页面	将会导致高线推送失败。		
回执开关 AppSecret • 点击后续动作 应用内指定界面 •	<ul> <li>③</li> <li>打开回执开关后,请务必参照文档</li> <li>请输入AppSecret</li> <li>打开应用</li> <li>打开应用</li> <li>打开网页</li> <li>Com.tencent.qcloud.tim.push.TIM</li> <li>此处会回调点击通知栏事件,通过广</li> </ul>	2, 正确配置回执地址, 否则 打开应用内指定页面 / 播或者回调给到应用, App可	将会导致高线推送失败。	立用	

Receipt Address: https://api.im.qcloud.com/v3/offline\_push\_report/meizu

### Note:

After enabling the Receipt Switch, please make sure the Receipt Address is configured correctly. Failing to configure or configuring the wrong address will affect the push feature.

For iOS Push Notification Reach Statistics Configuration, please refer to Statistics Push Arrival Rate.

No configuration is needed for other supported manufacturers; FCM does not support the push notification statistics feature.

Congratulations! You have completed the integration of the push plugin. Please be reminded: After the trial period or subscription expires, the push service (including regular message offline push, all-staff/Tag push, etc.) will automatically cease. To avoid affecting the normal use of your services, please make sure to purchase/renew in advance.

# Statistics

Last updated : 2024-06-13 10:21:45

This article aims to introduce the various statistics pages for push statistics data, enabling users to quickly query statistical data.

### Note:

The Statistics feature will only record the push data details of the last device logged in to, and does not support scenarios with multiple devices logged in to.

# **Regular Push**

# Push Records

You can query all push record data for users under this application. It is necessary to specify the time window, Sender ID, and Recipient ID. Querying user push data includes push time, Push ID, push title, and push content; it also supports locating specific push records by searching for push titles or push content.

### Query conditions

Time window: A specific time period within a specified date, maximum one hour, precise to minutes and seconds, required.

Sender ID: Required. Recipient ID: Required. Push Title: Optional. Push Content: Optional.

### Query results (Recorded in the last 7 days)

Push Time: The exact time of push reach.

Push ID: The unique ID of the push message, which can be used to locate the full push link situation of this push in the troubleshooting tool.

Push Title: Push display title.

Push Content: Push display content.



# Push Data

The statistics display various push metrics data of the application in recent days, mainly including an overview of yesterday's push, push data conversion funnel for multi-type time intervals, supports viewing classified by vendor dimensions, and details of daily push metrics data.

### Yesterday's Push Overview Metrics Data

Includes the number that can be sent, the number sent, the number reached, the number of clicks, the actual delivery rate, the reach rate, the click-through rate.

### **Time Range**

Supports yesterday, the last 7 days, the last 30 days, and specified time window types.

### Push Data Conversion Funnel

Includes the dimensions of sendable quantity, sent quantity, reached quantity, and clicked quantity.

### Support viewing by vendor dimension

Support viewing by all vendor categories.



# All-staff/Tag Push

# **Push Records**

You can query all the full/Tag push record data sent to users under this application, a specific time window must be specified, and the push data queried includes push time, task ID, and push request content; it also supports searching and locating specific push records based on push content.

### **Query conditions**

Time window: A specific time period within a specified date, up to 7 days, precise to minutes and seconds, required. Push Content: Optional.

## Query results (Recorded in the last 7 days)

Push Time: The exact time of push reach.

Task ID: The unique ID of the push message, which can be used to locate the full push notification link status in the troubleshooting tool.

Push Content: The detailed data in JSON format of the push.

Tencent Cloud	Overview Products + +
Chat	Push Records Current data center: China ① Telegram group WhatsApp group
∃∄ Application	All members/tag push Normal push
Management	2024-06-03 00:00:00 ~ 2024-06-03 23:59:59 📋 Push content Please enter the push content
III Overview	Query Reset
<u>ළ</u> Users	
器 Groups	Task ID Push time Push request content
Configuration ~	'{ "CloudCustomData": "", "Condition": { "TagsAnd": [ "push_plugin_test" ], "TagsOrt": null, "AttrsAnd": {}, "AttrsOrt": {}, "AstrsOrt": {}, "AstrsOrt: {}, "AstrsOrt
🕸 Webhook	2024-06-03 10:30:17 2024-06-03 10:30:17 2024-06-03 10:30:17 Services \n\",\"merit mediation and the session of the services \n\",\"merit mediation and the services \n\",\"
C Statistics	t markan na san yang tang tang tang tang tang tang tang t
Plugin	
🕞 Push 🔹 ^	
Access settings	
Access test	
Push Record	
Push Data	
<ul> <li>Push</li> <li>Troubleshooting</li> </ul>	
<ul> <li>Al Chatbot</li> </ul>	
🕡 Desk	
Tools	
⑦ Monitor	
🖬 Dev Tools 🛛 🗸	Total items: 1
Integration Guide	

# Push Data

The statistics display various push metrics data of the application in recent days, including the features page and the meaning of the metric data, same as standard push.

### Yesterday's Push Overview Metrics Data

Includes the number that can be sent, the number sent, the number reached, the number of clicks, the actual delivery rate, the reach rate, the click-through rate.

### **Time Range**

Supports yesterday, the last 7 days, the last 30 days, and specified time window types.

## Push Data Conversion Funnel

Includes the dimensions of sendable quantity, sent quantity, reached quantity, and clicked quantity.

### Support viewing by vendor dimension

Support viewing by all vendor categories.



## Task ID

All-staff/Tag Push supports viewing the detailed metrics data and details of a single push by Task ID.

Yesterday	Last 7 days	Last 30 days	2024-05-27 00:00:00 ~ 2024-	06-02 23:59:59 📩	Task ID 🕄	66221895_537529d8_2000005e8	Sending channel	Select all 😒	Query	Reset	
Trend				Details							
				2024.0	Date	Quantity that can be	e sent	Send quantity	Number of	contacts	Number of
		1134323		2024-0	o-zo (ruesuay)		34323	1074433		663433	
		883433									
Quant Quant	tity that can be ser per of clicks	nt 💼 Send quanti	ty Number of contacts	Total items: 9	)						

# **Metrics Data Calculation Method**

**Sendable Quantity:** The total number of valid devices that can be reached by filtering the target population selected for the push task (when a device is online and switches to the background, both online and offline pushes will be sent simultaneously, and it will be counted as two devices, not deduplicated).

**Sent Quantity:** The total number of valid devices that have been successfully distributed through the Instant Messaging (IM) Channel and the Manufacturer Channel among the valid devices that can be sent (when a device is online and switches to the background, both online and offline pushes will be sent simultaneously, and it will be counted as two devices, not deduplicated).

**Reached Quantity:** The total number of receipts successfully received by the device terminal through the Instant Messaging (IM) Channel and the Manufacturer Channel (when a device is online and switches to the background, both online and offline pushes will be sent simultaneously, and it will be counted as two devices, not deduplicated). **Clicked Quantity:** The total number of receipts for clicks after the push is successfully displayed.

Actual Delivery Rate: Sent Quantity / Sendable Quantity \* 100%.

Reach Rate: Reached Quantity / Sent Quantity \* 100%.

Click-through Rate: Clicked Quantity / Reached Quantity \* 100%.

# **Push Statistics Support Status by Manufacturer**



Manufacturer	Reach	Click
Huawei	Supported (Requires configuration of Acknowledgement)	Supported (Requires integration of TIMPush)
HONOR	Supported (Requires configuration of Acknowledgement)	Supported (Requires integration of TIMPush)
vivo	Supported (Requires configuration of Acknowledgement)	Supported (Requires integration of TIMPush)
OPPO	Supported	Supported (Requires integration of TIMPush)
Mi	Supported	Supported (Requires integration of TIMPush)
Meizu	Supported (Requires configuration of Acknowledgement)	Supported (Requires integration of TIMPush)
FCM	Not supported	Not supported
Apns	Supported (Requires integration of TIMPush)	Supported (Requires integration of TIMPush)

# **Troubleshooting Tool**

Last updated : 2024-06-13 10:21:45

This article aims to introduce the various pages and feature usage of the Push Notification Troubleshooting Tool, guiding users to troubleshoot the entire push notification link details when an offline push message is not received.

# **Regular Push**

The Regular Push Troubleshooting Tool is primarily for developer users to troubleshoot specific push notification reception issues. By finding the push record and copying the unique Push ID of that push, one can query the detailed push notification link details of that push.

# **Query Fields**

Push ID: The unique ID used to identify a regular push, which can be found in the push records. It is required.

# Viewing the query result

Basic Information: The basic information of the current device at the time of push delivery, including model, operating system, SDK, and Plugin Version Number, etc.

Device Status: The push's notification bar switch status and the device's token binding status at the time of dispatch. Push Status: The end-to-end information for the push delivery, including the entire link from IM server > Manufacturer Server > Terminal Device > User Click.

Content Cloud	Overview Products - +	
Chat	Push troubleshooting	
王는 Application	All members/tag push Normal push	
Management	Push ID ① 6655 Reset	
III Overview		
은 Users	Query results	
퉒 Groups	Basic information	Push status
Configuration ~		
🕸 Webhook	Device model RMX3370	Pust
C Statistics	Operating System Android	
Plugin	System version 11 SDK version 8.0.5808	
Fa Push	Push plug-in version 8.0.5808	
Access settings	Manufacturer token reality Manufacturer brand Oppo	
	Certificate ID	
Access test		
* Push Record	Equipment status	
• Push Data	Notification bar status () Enable	
<ul> <li>Push Troubleshooting</li> </ul>	token binding status Bound	
Al Chathat		
Ul Desk		
Tools		
⑦ Monitor		
📩 Dev Tools 🛛 🔪		
Integration Guide		
		Normal push only supports viewing offline pus

# All-staff/Tag Push

The All-staff/Tag Push Troubleshooting Tool is primarily for administrator users to investigate the reception status of a specific user in an All-staff/Tag push. By finding the push record and copying the unique Task ID of that push, and entering the recipient's UserID, one can query the detailed push notification link details for that UserID.

# Query Fields

Task ID: The unique ID used to identify an All-staff/Tag push, which can be found in the push records. It is required. UserID: A user ID for whom the All-staff/Tag push is intended, also the recipient of the push. It is required.

# Viewing the query result

Basic Information: At the time of push delivery for the specified UserID, the basic information of the current device includes model, operating system, SDK, and plugin version number, etc.

Device Status: At the time of push delivery for the specified UserID, the device's notification bar switch status, as well as the device's token binding status.

Push Status: The end-to-end information of the push delivered to the specified UserID, including the entire link situation from IM server > Manufacturer Server > Terminal Device > User Click.

Tencent Cloud	Overview Products + +	
Chat	Push troubleshooting	
국 Application	All members/tag push Normal push	
Management	Task ID 🕥 ( Cuery Reset	
III Overview		
은 Users	Query results	
볿 Groups	Basic information	Push status
Configuration ·	Durk time 0004.06.03.10/20/49	
🕸 Webhook	Device model RMX3370	Task II
() Statistics	Operating System Android	
Plugin	SDK version 8.0.5808	
🕞 Push 🔹 ^	Push plug-in version 8.0.5808	
Access settings	Manufacturer brand Oppo	
Access test	Certificate ID	
Push Record	Equipment status	
• Push Data	Notification bar status () Enable	$\odot$
<ul> <li>Push</li> <li>Troubleshooting</li> </ul>	token binding status Bound	Submit online channel 2024-06-03 10:30:18
O Al Chatbot		
🔘 Desk		$\odot$
Tools		Push message to device 2024-06-03 10:30:18
⑦ Monitor		
🖬 Dev Tools 🛛 👻		•
Integration Guide		()
		User click status statistics for online push will be support
		Ine application is switched to the background a

# Client APIs Android

Last updated : 2024-08-01 11:28:30

# TIMPushManager

public abstract class TIMPushManager: Push Plugin Interface Class.

# **API** overview

# **Register/Unregister Push Service Interface**

After initializing and successfully logging in to IM, you can register for push services.

API	Description
registerPush	To register for push services, read the configuration file timpush- configs.json from the project.
unRegisterPush	Unregister to close offline push services, call before logging out of the IM account.
disableAutoRegisterPush	To disable the plugin from automatically registering for push services after logging in, call before registering for push services.

# FCM Custom Ringtone Configuration Interface

Once configured, the custom ringtone is effective. The sender must include the channelld in the offline message sent.

API	Description
setCustomFCMRing	To configure FCM's custom ringtone, it needs to be called before registering for push services.
setCustomConfigFile	The custom replacement plugin by default reads the push service registration configuration file timpush-configs.json, which needs to be called before registering for push services

# Interface Details

## **Static Public Member Functions**

static TIMPushManager getInstance(): Retrieves the TIMPushManager manager instance.

## **Member Function Description**

### abstract void registerPush(Context context, TIMPushCallback callback)

Register offline push service, call upon successful login to the IM account. (To facilitate the easiest possible integration into the push service, the plugin will automatically read the configuration file timpush-configs.json in the project to obtain the information needed to register for the push service)

### Note:

You need to use the login API provided by TUILogin of the TUICore component to log in; the plugin will automatically detect this and register the push service.

If you do not wish to use the API provided by TUILogin, you need to manually call this interface to register the service after completing the login operation.

### abstract void unRegisterPush(TIMPushCallback callback)

Unregister to close offline push services, call before logging out of the IM account.

### Note:

If you do not wish to use the push service, you can manually call this interface to unregister the service. If you log out using the logout API provided by TUILogin of the TUICore component, the plugin will automatically detect this and unregister the push service.

### abstract void disableAutoRegisterPush()

To disable the plugin from automatically registering for the push service, it's necessary to call this before logging in. **Note:** 

If you log in using the login API provided by TUILogin in the TUICore component, the plugin will by default automatically register for the push service. Calling this interface can disable the automatic registration.

### abstract void setCustomFCMRing(String channelld, String ringName, boolean enable)

To configure FCM's custom ringtone, it needs to be called before registering for push services.

### Note:

Once configured, the custom ringtone is effective. The sender must include the channelld in the offline message sent. Parameter description:

API	Description
channelld	FCM channel uniquely defines the channel ID of the notification bar within the app.
ringName	FCM channel defines the name of the push ringtone for the notification bar, located in the raw directory and does not require a file suffix.



enable

Set whether the offline push prompt ringtone uses a custom ringtone.

### abstract void setCustomConfigFile(String configs)

The custom replacement plugin by default reads the push service registration configuration file timpush-configs.json, which needs to be called before registering for push services.

### Note:

Mainly used for dynamically switching different configuration files for push registration in multiple environments, for example: push feature integration and testing under different configuration files in production and test environments; For methods to switch during the static compilation period, please refer to: buildConfigField("String", "custom\_timpush\_configs", "\\"Custom Definition File Name\\"")

Parameter description:

Parameter	Description
configs	The name of the custom Definition configuration file should remain unchanged: "Engineering Root Directory/app/src/assets/"

# iOS

Last updated : 2024-06-13 10:21:45

# **API** Overview

# **Register/Unregister Push Service Interface**

After initialization and successful log in to IM, you can register for the push service.

API	Description
registerPush	Register the push service after log in to is complete
unRegisterPush	Unregister the offline push service when logging out of the IM account.
disableAutoRegisterPush	Disable the plugin after logging in to automatically register the push service; it needs to be called before registering the push service

# Statistics on the push arrival rate of TIMPush

If you need to track the arrival and click data of push notifications, you need to proactively call this function in the Notification Service Extension.

API	Description
onReceiveNotificationRequest:inAppGroupID:callback:	It is only supported to be called in the Notification Service Extension's '- didReceiveNotificationRequest:withContentHandler:' method; appGroup indicates the App Group shared between the main App and the Extension. It needs to be configured in the Capability of the main App to enable App Groups.

# Interface details

# **Function Description**



### + (void)registerPush

Register the offline push service when the IM account logs in successfully. (The information needed to register the push service comes from the offlinePushCertificateID of the TIMPushDelegate protocol implemented in your AppDelegate)

#### Note:

You need to log in using the login interface provided by TUILogin in the TUICore component, and the plugin will automatically perceive and register the push service.

If you don't want to use the interface provided by TUILogin, after completing the login operation, you need to manually call this interface to register the service.

Usage: [TIMPush registerPush];

### + (void)unRegisterPush

Unregister the offline push service when logging out of the IM account.

#### Note:

You need to log in using the login interface provided by TUILogin in the TUICore component, and the plugin will automatically perceive and register the push service.

If you don't want to use the interface provided by TUILogin, after completing the login operation, you need to manually call this interface to register the service.

**Usage:** [TIMPush unRegisterPush];

#### + (void)disableAutoRegisterPush

To turn off the plugin's automatic registration of the push service, you need to call this before logging in.

#### Note:

If you log in using the login interface provided by TUILogin in the TUICore component, the plugin automatically registers the push service by default. Calling this interface will turn off automatic registration.

**Usage:** [TIMPush disableAutoRegisterPush];

# + (void)onReceiveNotificationRequest:(UNNotificationRequest \*)request inAppGroupID:(NSString \*)appGroupID callback:(TIMPushNotificationExtensionCallback)callback

Statistics on the push arrival rate of TIMPush

You need to implement the `- applicationGroupID` method in the AppDelegate.m file, returning the App Group ID. And call this function in the Notification Service Extension's '- didReceiveNotificationRequest:withContentHandler:' method.

#### Note:

The appGroup identifier represents the App Group shared between the main App and its Extension. It needs to be configured in the main App's Capability as App Groups.

Parameter description:

request	Parameters carried by the UNNotificationServiceExtension callback
---------	-------------------------------------------------------------------

appGroupID	The appGroup identifier represents the App Group shared between the main App and its Extension. It needs to be configured in the main App's Capability as App Groups.
callback	typedef void(^TIMPushNotificationExtensionCallback)(UNNotificationContent *content) Statistical function Callback, carrying content information

Usage:



- (void)didReceiveNotificationRequest:(UNNotificationRequest \*)request withContentH

```
self.contentHandler = contentHandler;
NSString * appGroupID = kTIMPushAppGorupKey;
__weak typeof(self) weakSelf = self;
[TIMPush onReceiveNotificationRequest:request inAppGroupID:appGroupID callback:
    weakSelf.bestAttemptContent = [content mutableCopy];
    // Modify the notification content here...
    // self.bestAttemptContent.title = [NSString stringWithFormat:@"%@ [modifie
    weakSelf.contentHandler(weakSelf.bestAttemptContent);
  }];
}
```

# Flutter

Last updated : 2024-06-13 10:21:45

# TencentCloudChatPush

class TencentCloudChatPush: Push Plugin Interface Class.

# **API** Overview

# Register/Unregister Push Service Interface

After initialization and successful log in to IM, you can register for the push service.

API	Description
registerOnNotificationClickedEvent	Register message click callback in advance.
registerPush	Register Push Service, optionally override the push message from interface parameter json.
unRegisterPush	Unregister the offline push service when logging out of the IM account.

## FCM Custom Ringtone Configuration Interface

Once configured, the custom ringtone is effective. The sender must include the channelld in the offline message.

API	Description
configFCMPrivateRing	To configure the custom ringtone for FCM, it needs to be called before registering for the push service.

## **Special Configuration Interface for Push Channel**

API	Description
setPushBrandId	To specify the vendor channel type for device offline push, it needs to be called before registering for the push service.
getPushBrandId	Obtain the vendor channel type currently in use for device offline push.
checkPushStatus	After completing the access configuration for each vendor, you can use this interface to test the push status on the corresponding vendor

	devices.
setApnsCertificateID	Configure the push certificate ID for APNs separately.
setApplicationGroupID	Configure the Application Group ID for the iOS project.
getAndroidPushToken	Retrieve Android device manufacturer Token.
setAndroidPushToken	Manually specify the Android device manufacturer Token.
setAndroidCustomTIMPushConfigs	Manually replace the default push configuration file timpush-configs.json read by the plugin with a self-defined one. This needs to be invoked before registering the push service.

# Interface details

# **Push Plugin Class**

TencentCloudChatPush(): Retrieves the TencentCloudChatPush push plugin instance, which is a Static Singleton. All subsequent steps will invoke methods through this singleton instance.

# **Member Function Description**

### registerOnNotificationClickedEvent

Configure Message Click Callback Function.

You can call or adjust the callback function dynamically as needed, or directly pass it during registerPush . Sample Code:



void \_onNotificationClicked({required String ext, String? userID, String? groupID})
print("\_onNotificationClicked: \$ext, userID: \$userID, groupID: \$groupID");
/// Custom processing
}

#### TencentCloudChatPush().registerOnNotificationClickedEvent(onNotificationClicked: \_o

#### **Parameter Description:**

Parameter		Туре	Description
onNotificationClicked	ext	String	This is the complete ext information carried by the message,

			specified by the sender. If not specified, a default value is used. You can navigate to the corresponding page by resolving this field.
	userID	String?	This parameter corresponds to userID, automatically attempts to parse the ext Json String, to retrieve the One-on-one Chat counterpart's userID within. <b>Note:</b> If you have not defined the ext field yourself, and it is set by default by the SDK or UIKit, you can use the default parsing here. If the attempt to parse fails, it will be null empty.
	groupID	String?	This parameter corresponds to groupID, automatically attempts to parse the ext Json String, to retrieve the Group Chat groupID information within. <b>Note:</b> If you have not defined the ext field yourself, and it is set by default by the SDK or UIKit, you can use the default parsing here. If the attempt to parse fails, it will be null empty.

## registerPush

Register the offline push service, call after the IM account logs in successfully.

Sample Code:





```
TencentCloudChatPush().registerPush(
    onNotificationClicked: _onNotificationClicked,
    androidPushOEMConfig: "can be left null",
    apnsCertificateID: 0,
```

#### );

### **Parameter Description:**

Parameter		Туре	Description
onNotificationClicked	ext	String	This is the complete ext information carried by the message, spec



encent Cloud				
		0.1		

	userID	String?	This parameter corresponds to userID, automatically attempts to p <b>Note:</b> If you have not defined the ext field yourself, and it is set by defaul
	groupID	String?	This parameter corresponds to groupID, automatically attempts to <b>Note:</b> If you have not defined the ext field yourself, and it is set by defaul
androidPushOEMConfi	g	String	If the timpush-configs.json configuration file has been imported, th Optional Android-side parameters json:







		<pre>// google    "fcmPushBussinessId": "", // Certificat    // oppo    "oppoPushBussinessId": "", // Certifica    "oppoPushAppKey": "",// Application App:    "oppoPushAppSecret": "",// Application 2    // honor    "honorPushBussinessId": "", // Certifica }</pre>
apnsCertificateID	int	If setApnsCertificateID method has already been called separately

## unRegisterPush

Unregister the offline push service, to be called after the IM account logs out.

## Sample Code:





TencentCloudChatPush().unRegisterPush();

### configFCMPrivateRing

To configure the custom ringtone for FCM, it needs to be called before registering for the push service. **Sample Code:** 





TencentCloudChatPush().configFCMPrivateRing(channelId: channelId, ringName: ringNam

## Parameter Description:

Parameter Name	Туре	Description
channelld	String	FCM channel custom notification bar's channel ID, unique within the app.
ringName	String	FCM channel's custom push ringtone name, located in the raw directory



		and does not require a file suffix.
enable	bool	Setting whether the offline push prompt ringtone uses a custom ringtone.

### Note:

Once configured, the custom ringtone is effective. The sender must include the channelld in the offline message.

### setPushBrandId

To specify the vendor channel type for device offline push, it needs to be called before registering for the push service. **Note:** 

This interface allows specifying the use of a manufacturer's push channel type, for example, specifying the use of the FCM channel for push on Xiaomi devices abroad with setPushBrandld(TencentCloudChatPushBrandlD.FCM). Generally, there's no need to specify the channel type, as the component will automatically identify the device manufacturer category to register and use the corresponding manufacturer channel.

### Sample Code:





TencentCloudChatPush().setPushBrandId(brandID: brandID);

## Parameter description

:		
Parameter	Description	
brandID	Vendor	Device Type
	XiaoMi	TencentCloudChatPushBrandID.XiaoMi



	HuaWei	TencentCloudChatPushBrandID.HuaWei	
	FCM	TencentCloudChatPushBrandID.FCM	
		Meizu	TencentCloudChatPushBrandID.Meizu
	Орро	TencentCloudChatPushBrandID.Oppo	
	Vivo	TencentCloudChatPushBrandID.Vivo	
	Honor	TencentCloudChatPushBrandID.Honor	

## getPushBrandId

Obtain the vendor channel type currently in use for device offline push.

Sample Code:





```
final res = await TencentCloudChatPush().getPushBrandId();
if(res.code == 0) {
  final TencentCloudChatPushBrandID brandID = res.data;
}
```

## **Return Type:**

Future<TencentCloudChatPushResult<TencentCloudChatPushBrandID>>

### checkPushStatus

After completing the access configuration for each vendor, you can use this interface to test if the corresponding vendor devices can receive pushes.

### Sample Code:



```
final res = await TencentCloudChatPush().checkPushStatus(brandID: 2002);
if(res.code == 0) {
   final status = res.data;
}
```

### **Parameter Description:**
## STencent Cloud

Parameter Name	Туре	Description
brandID	TencentCloudChatPushBrandID	BrandID Definition as shown in the table above.

### Return Type:

Future<TencentCloudChatPushResult<String>>

If true, push notifications can be successfully sent.

#### setApnsCertificateID

Separately configure the APNs Push Certificate ID. You can call or dynamically adjust the Certificate ID as needed, or

directly pass it during registerPush .

Sample Code:





TencentCloudChatPush().setApnsCertificateID(apnsCertificateID: 0);

### Parameter Description:

Parameter Name	Туре	Description
apnsCertificateID	int	The Certificate ID assigned to the APNs certificate in the Tencent Cloud IM Console.

#### setApplicationGroupID

Configure the Application Group ID for the iOS project.

#### Sample Code:



TencentCloudChatPush().setApplicationGroupID(applicationGroupID: "");

### Parameter Description:

Parameter Name	Туре	Description
applicationGroupID	String	The format is: group + [Main Bundle ID] + key.



#### getAndroidPushToken

Retrieve Android device manufacturer Token.

Sample Code:



TencentCloudChatPush().getAndroidPushToken();

#### **Return Type:**

Future<TencentCloudChatPushResult<String>>

The returned String type data is the vendor's push Token..



#### setAndroidPushToken

Manually specify the Android device manufacturer Token.

#### Sample Code:



TencentCloudChatPush().setAndroidPushToken(businessID: 10000, pushToken: "pushToken

#### **Parameter Description:**

Parameter Name	Туре	Description



businessID	String	Push certificate ID, available from the Tencent Cloud IM console, under the push certificate card.
pushToken	String	Obtain the vendor's push Token in your own way.

#### setAndroidCustomTIMPushConfigs

Manually replace the default push configuration file timpush-configs.json read by the plugin with a custom one. This needs to be invoked before registering the push service (registerPush).

#### Note:

Mainly used for dynamically switching different configuration files for push registration in various environments, for example: integrating and testing push features under different configuration files in production and testing environments.

#### Sample Code:





TencentCloudChatPush().setAndroidCustomTIMPushConfigs(configs: "");

### Parameter Description:

Parameter Name	Туре	Description
configs	String	The name of the custom configuration file, the path must remain unchanged: "Engineering Root Directory/android/app/src/assets/".



# All Staff/Tag Push Pushing to All/Tagged Users

Last updated : 2024-07-23 17:57:52

Pushing to all/tagged users not only supports sending specific content to all users, but also can send personalized content to specific user groups based on tags and attributes, such as member events and regional notifications. This helps effectively attract, convert, and activate new users. It also supports pushing delivery reports and self-service troubleshooting tools. Refer to Feature Overview for the effect.

#### Note:

The account must have been logged in or manually imported before it can receive push messages.

## Feature Overview

Supports sending push messages to all users.

Supports sending push messages based on user attributes.

Supports sending push messages based on user tags.

The administrator pushes a message, and the recipient sees that the sender of the message is the administrator.

The administrator specifies an account to push a message to other accounts, and the recipient sees that the sender is not the administrator, but the account specified by the administrator.

Supports roaming, with the same storage duration as regular messages.

Due to the large number of accounts that need to be delivered for pushing to all/tagged users, it takes a certain amount of time (depending on the total number of accounts) to complete the delivery of all accounts.

By setting OnlineOnlyFlag to 1, push messages can only be sent without saving sessions, roaming, and unread messages.

## **API Call Description**

#### Sample Request URL





https://xxxxxx/v4/timpush/push?usersig=xxx&identifier=admin&sdkappid=88888888&rando

### **Request Parameters**

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
ХХХХХХ	The dedicated domain name corresponding to the country/region of the SDKAppID. China: console.tim.qq.com

	Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/timpush/push	Request API.
usersig	The signature generated by the app admin account. For details, see Generating UserSig.
identifier	The app admin account.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
random	A random 32-bit unsigned integer.
contenttype	The value is always json .

## **Call Frequency**

This API includes pushing to all users and pushing to users by attribute or tag. By default, it can be called 100 times per day. The interval between two pushes must be greater than 1 second.

#### Note:

By default, the API can be called 100 times per day for free. For each additional call per day, the cost will increase by 10 USD per day. If you need to adjust the frequency of API calls, go to the Chat console to modify it.

## **Sample Request Packets**

#### **Pushing Messages to All Users**

The administrator pushes messages to all users:







}

```
],
"OfflinePushInfo": {
    "PushFlag": 0, // 0: Enables offline push; 1: Disables offline push.
    "Title": "offline push title",
    "Desc": "offline push content"
}
```

The administrator specifies an account for pushing messages to all users. In the example, the sender account is

xiaoming :



```
{
 "From_Account": "xiaoming",
 "MsgRandom": 3674128,
 "OnlineOnlyFlag": 0, // 0: Saves roaming and unread messages. In this case, the p
                       // 1: Does not save roaming or unread messages. In this case
  "MsgBody": [
     {
          "MsgType": "TIMTextElem",
          "MsgContent": {
              "Text": "hi, beauty"
          }
     }
 ],
 "OfflinePushInfo": {
      "PushFlag": 0, // 0: Enables offline push; 1: Disables offline push.
      "Title": "offline push title",
      "Desc": "offline push content"
 }
}
```

#### Note:

From\_Account is the message sender account and can be specified as any existing account. If no sender is specified or the specified sender does not exist, the API caller account will be taken by default.

The administrator pushes messages only to online users (online push):





```
}
],
"OfflinePushInfo": {
    "PushFlag": 1, // 0: Enables offline push; 1: Disables offline push.
}
```

#### Note:

Setting OnlineOnlyFlag to 1 indicates only push (online push+offline push). Setting OfflinePushInfo.PushFlag to 1 indicates no offline push. Therefore, the above example indicates that messages are pushed only to online users.

#### Pushing Messages by User Tag

The administrator pushes messages to users tagged with "Stock A" and "Stock B":





```
"MsgContent": {
    "Text": "hi, beauty"
    }
    }
    // "OfflinePushInfo": {
    "PushFlag": 0, // 0: Enables offline push; 1: Disables offline push.
    "Title": "offline push title",
    "Desc": "offline push content"
    }
}
```

The administrator pushes messages to users tagged with "Stock A" or "Stock B":





```
"MsgContent": {
    "Text": "hi, beauty"
    }
    }
    // 
    // 
    // 
    // 
    // OfflinePushInfo": {
        "PushFlag": 0, // 0: Enables offline push; 1: Disables offline push.
        "Title": "offline push title",
        "Desc": "offline push content"
    }
}
```

## Pushing Messages by User Attribute

The administrator pushes messages to Platinum Premier membership in Shenzhen:





```
"MsgBody": [
    {
        "MsgType": "TIMTextElem",
        "MsgContent": {
            "Text": "hi, beauty"
        }
    }
    }
}
```

#### Note:

From\_Account is the message sender account and can be specified as any existing account. If no sender is specified or the specified sender does not exist, the API caller account will be taken by default.

The administrator pushes messages to Platinum Premier users in Shenzhen:





```
"MsgBody": [
    {
        "MsgType": "TIMTextElem",
        "MsgContent": {
            "Text": "hi, beauty"
        }
    }
    }
    // OfflinePushInfo": {
        "OfflinePushInfo": {
            "PushFlag": 0, // 0: Enables offline push; 1: Disables offline push.
            "Title": "offline push title",
            "Desc": "offline push content"
    }
}
```

## **Request Fields**

Field	Туре	Required	Description
From_Account	String	No	Account of the message sender, which can be specified as any existing account. <b>Note:</b> If no sender is specified or the specified sender does not exist, the API caller account will be taken by default.
MsgRandom	Integer	Yes	A random number (32-bit unsigned integer) of the message. It is used by the backend for message deduplication within the same second. Make sure a random number is entered.
OnlineOnlyFlag	Integer	No	The default value is 0, which means to save roaming and unread messages. In this case, the push concurrency limit is 200 users/second. 1 means not to save historical messages and not to count unread messages. In this case, there is no push concurrency limit.
Condition	Object	No	Valid values: The or condition of attributes: AttrsOr The and condition of attributes: AttrsAnd The or condition of tags: TagsOr The and condition of tags: TagsAnd <b>Note:</b> AttrsOr, AttrsAnd, TagsOr, and TagsAnd cannot coexist. If no condition is specified, messages are pushed to all users.
TagsOr	Array	No	The union of tag conditions. A tag is a string of up to 50 bytes. <b>Note:</b> Attributes and tags cannot be used as push conditions at the same time. The TagsOr condition can contain up to 10 tags.



TagsAnd	Array	No	The intersection of tag conditions. A tag is a string of up to 50 bytes. <b>Note:</b> Attributes and tags cannot be used as push conditions at the same time. The TagsAnd condition can contain up to 10 tags.
AttrsOr	Object	No	The union of attribute conditions.
AttrsAnd	Object	No	The intersection of attribute conditions.
MsgBody	Array	Yes	Message body. For more information on the message format, see Message Formats. Note that MsgBody is an array that can contain multiple message elements.
MsgType	String	Yes	TIMTextElem (text message) TIMTextElem (text message) TIMLocationElem (location message) TIMFaceElem (emoji message) TIMCustomElem (custom message) TIMSoundElem (voice message) TIMImageElem (image message) TIMFileElem (file message) TIMFileElem (video message)
MsgContent	Object	Yes	Different message object types ( MsgType ) have different formats ( MsgContent ). For more information, see the <b>TIMMsgElement</b> section in Message Formats.
OfflinePushInfo	Object	No	The information to be pushed offline. For more information, see Message Formats.

## Sample Response Packets





```
{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0,
    "TaskId": "53743040_144115212910570789_4155518400_15723514"
}
```

## **Response Fields**

Field Type	Description
------------	-------------

ActionStatus	String	Processing result. OK : succeeded. FAIL : failed.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error information.
Taskld	String	Push task ID.

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information. For common error codes (60000 to 79999), see Error Codes.

The following table describes the error codes specific to this API:

Error Code	Description	
90001	Failed to parse the JSON format. Check whether the request packets meet JSON specifications.	
90002	The MsgBody field in the JSON request packets does not meet message format requirements or it is not an array. For more information, see the <b>TIMMsgElement</b> section in Message Formats.	
90005	The MsgRandom field is missing in the JSON request packets or it is not an integer.	
90007	The MsgBody field in the JSON request packets is not an array. Change it to an array.	
90009	The request requires app admin permissions.	
90010	The JSON request packets do not meet message format requirements. For more information, see the <b>TIMMsgElement</b> section in Message Formats.	
90020	The tag length exceeds the limit (the maximum length allowed is 50 bytes).	
90022	TagsOr and TagsAnd in the push conditions contain repeated tags.	
90024	Pushes are too frequent. The interval between two pushes must be greater than 1 second.	
90026	Incorrect offline message storage period. The value cannot exceed 7 days.	
90032	The number of tags in the push conditions exceeds 10, or the number of tags in the tag adding request exceeds 10.	
90033	Invalid attribute.	

## 🔗 Tencent Cloud

90039	Message push by attribute and message push by tag are mutually exclusive.	
90040	A tag in the push conditions is null.	
90045	The feature of pushing to all/tagged users is not enabled.	
90047	The number of pushes exceeds the daily quota (default quota: 100).	
91000	Internal service error. Try again.	

## **API** Debugging Tool

Use the RESTful API online debugging tool to debug this API.

## References

Pushing to All/Tagged Users Setting Application Attribute Names Obtaining Application Attribute Names Setting User Attributes Deleting User Attributes Obtaining User Attributes Obtaining User Tags Adding User Tags Deleting User Tags Deleting All User Tags Recalling Push

# **Obtaining Application Attribute Names**

Last updated : 2024-07-16 11:40:03

## Feature Overview

This API is used by the administrator to obtain application attribute names. To call this API, you need to set application attribute names first.

### Sample Request URL





https://xxxxxx/v4/timpush/get\_attr\_name?usersig=xxx&identifier=admin&sdkappid=88888

### **Request Parameters**

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
XXXXXX	The dedicated domain name corresponding to the country/region of the SDKAppID.

	China: console.tim.qq.com Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com
	Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/timpush/get_attr_name	Request API.
usersig	The signature generated by the app admin account. For details, see Generating UserSig.
identifier	The app admin account.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
random	A random 32-bit unsigned integer.
contenttype	The value is always json .

## Maximum Call Frequency

100 times/second

## Sample Request Packets





{ }

### **Request Fields**

None.

## Sample Response Packets





```
{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0,
    "AttrNames": {
        "0": "sex",
        "1": "city",
        "2": "level"
    }
}
```

Field	Туре	Description
ActionStatus	String	Processing result. OK : succeeded. FAIL : failed.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error information.
AttrNames	Object	A series of key-value pairs. Each key-value pair indicates the name of the corresponding attribute. For example, "0":"xxx" indicates that the name of attribute 0 is xxx.

## Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information. For common error codes (60000 to 79999), see Error Codes.

The following table describes the error codes specific to this API:

Error Code	Description	
90001	Failed to parse the JSON format. Check whether the request packets meet JSON specifications.	
90009	The request requires app admin permissions.	
90018	The number of requested accounts exceeds the limit.	
91000	Internal service error. Try again.	

## **API** Debugging Tool

Use the RESTful API online debugging tool to debug this API.

## References

Pushing to All/Tagged Users Setting Application Attribute Names



Obtaining Application Attribute Names Setting User Attributes Deleting User Attributes Obtaining User Attributes Obtaining User Tags Adding User Tags Deleting User Tags

**Recalling Push** 

# Setting Application Attribute Names

Last updated : 2024-07-16 11:40:03

## Feature Overview

You can set a maximum of 10 custom user attributes for each application. This API is used to set the name of each attribute. After you set attribute names, they can be used for push by user attribute.

### Sample Request URL




https://xxxxxx/v4/timpush/set\_attr\_name?usersig=xxx&identifier=admin&sdkappid=88888

#### **Request Parameters**

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
XXXXXX	The dedicated domain name corresponding to the country/region of the SDKAppID.

	China: console.tim.qq.com	
	Singapore: adminapisgp.im.qcloud.com	
	Seoul: adminapikr.im.qcloud.com	
	Frankfurt: adminapiger.im.qcloud.com	
	Silicon Valley: adminapiusa.im.qcloud.com	
	Jakarta: adminapiidn.im.qcloud.com	
v4/timpush/set_attr_name	Request API.	
usersig	The signature generated by the app admin account. For details, see Generating UserSig.	
identifier	The app admin account.	
sdkappid	The SDKAppID assigned by the Chat console when an application is created.	
random	A random 32-bit unsigned integer.	
contenttype	The value is always json .	

### Maximum Call Frequency

100 times/second

### Sample Request Packets

Set attribute 0 of the application to sex , attribute 1 to city , and attribute 2 to country .





### **Request Fields**

©2013-2022 Tencent Cloud. All rights reserved.

Field	Туре	Required	Description
Digital key	String	Yes	Attribute number (0 to 9).
Attribute name	String	Yes	The attribute name cannot exceed the length limit of 50 bytes. An application can have a maximum of 10 push attributes (numbered from 0 to 9), and users can customize the meaning of each attribute.

### Sample Response Packets



```
"ActionStatus": "OK",
"ErrorInfo": "",
"ErrorCode": 0
```

#### **Response Fields**

}

Field	Туре	Description
ActionStatus	String	Processing result. OK : succeeded. FAIL : failed.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error information.

# **Error Codes**

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information. For common error codes (60000 to 79999), see Error Codes.

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the request packets meet JSON specifications.
90009	The request requires app admin permissions.
91000	Internal service error. Try again.

# **API Debugging Tool**

Use the RESTful API online debugging tool to debug this API.

### References

Pushing to All/Tagged Users Setting Application Attribute Names Obtaining Application Attribute Names Setting User Attributes Deleting User Attributes Obtaining User Attributes Obtaining User Tags Adding User Tags Deleting User Tags Deleting All User Tags

**Recalling Push** 

©2013-2022 Tencent Cloud. All rights reserved.

# **Obtaining User Attributes**

Last updated : 2024-07-23 18:06:24

# Feature Overview

This API is used by the administrator to obtain user attributes. Up to 100 users' attributes can be obtained at a time. To call this API, you need to set application attribute names first.

#### Sample Request URL





https://xxxxxx/v4/timpush/get\_attr?usersig=xxx&identifier=admin&sdkappid=888888888&r

#### **Request Parameters**

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
ХХХХХХ	The dedicated domain name corresponding to the country/region of the SDKAppID. China: console.tim.qq.com

	Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/timpush/get_attr	Request API.
usersig	The signature generated by the app admin account. For details, see Generating UserSig.
identifier	The app admin account.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
random	A random 32-bit unsigned integer.
contenttype	The value is always json .

### Maximum Call Frequency

100 times/second

### Sample Request Packets







### **Request Fields**

### 🔗 Tencent Cloud

Field	Туре	Required	Description
To_Account	Array	Yes	List of target user accounts.

#### Sample Response Packets



{

```
"ActionStatus": "OK",
"ErrorInfo": "",
"ErrorCode": 0,
```

```
"UserAttrs": [
        {
            "To_Account": "Jack",
            "Attrs": {
                "sex" : "Female",
                 "city": "New York"
            }
        },
        {
            "To_Account": "abc",
            "Attrs": {}
        },
        {
            "To_Account": "Daved",
            "Attrs": {
                "sex": "M",
                 "city": "Shenzhen"
            }
        }
    ]
}
```

#### **Response Fields**

Field	Туре	Description
ActionStatus	String	Processing result. OK : succeeded. FAIL : failed.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error information.
UserAttrs	Array	List of user attributes.
To_Account	String	User account.
Attrs	Object	Attribute content.

# Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information. For common error codes (60000 to 79999), see Error Codes.

The following table describes the error codes specific to this API:

### S Tencent Cloud

Error Code	Description
90001	Failed to parse the JSON format. Check whether the request packets meet JSON specifications.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error. Try again.

# API Debugging Tool

Use the RESTful API online debugging tool to debug this API.

## References

Pushing to All/Tagged Users Setting Application Attribute Names Obtaining Application Attribute Names Setting User Attributes Deleting User Attributes Obtaining User Attributes Obtaining User Tags Adding User Tags Deleting User Tags Deleting All User Tags Recalling Push

# Setting User Attributes

Last updated : 2024-07-16 11:40:03

# Feature Overview

This API is used by the administrator to set attributes for users. Up to 100 users' attributes can be set at a time. To call this API, you need to set application attribute names first.

#### Sample Request URL





https://xxxxxx/v4/timpush/set\_attr?usersig=xxx&identifier=admin&sdkappid=888888888&r

#### **Request Parameters**

Parameter	Description		
https	The request protocol is HTTPS and the request method is POST.		
ХХХХХХ	The dedicated domain name corresponding to the country/region of the SDKAppID. China: console.tim.qq.com		

	Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/timpush/set_attr	Request API.
usersig	The signature generated by the app admin account. For details, see Generating UserSig.
identifier	The app admin account.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
random	A random 32-bit unsigned integer.
contenttype	The value is always json .

### Maximum Call Frequency

100 times/second

### Sample Request Packets





```
"To_Account": "xiaojun013",
"Attrs": {
"city": "attr3",
"sex": "attr4"
}
}
}
```

#### **Request Fields**

Field	Туре	Required	Description
To_Account	String	Yes	Target user account.
Attrs	Object	Yes	Attribute set. Each attribute is a key-value pair, with the key being the attribute name and the value being the user's corresponding attribute value. A user attribute value cannot exceed 50 bytes.

### Sample Response Packets





```
{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0
}
```

### **Response Fields**

Field	Туре	Description

Chat



ActionStatus	String	Processing result. OK : succeeded. FAIL : failed.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error information.

# **Error Codes**

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information. For common error codes (60000 to 79999), see Error Codes.

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the request packets meet JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
90033	Invalid attribute.
91000	Internal service error. Try again.

# **API** Debugging Tool

Use the RESTful API online debugging tool to debug this API.

## References

Pushing to All/Tagged Users Setting Application Attribute Names Obtaining Application Attribute Names Setting User Attributes Deleting User Attributes Obtaining User Attributes Obtaining User Tags Adding User Tags



Deleting User Tags Deleting All User Tags Recalling Push

# **Deleting User Attributes**

Last updated : 2024-07-16 11:40:03

# Feature Overview

This API is used by the administrator to delete attributes for users. Up to 100 users' attributes can be deleted at a time. To call this API, you need to set application attribute names first.

#### Sample Request URL





https://xxxxxx/v4/timpush/remove\_attr?usersig=xxx&identifier=admin&sdkappid=8888888

#### **Request Parameters**

Parameter	Description		
https	The request protocol is HTTPS and the request method is POST.		
ХХХХХХ	The dedicated domain name corresponding to the country/region of the SDKAppID. China: console.tim.qq.com		

	Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/timpush/remove_attr	Request API.
usersig	The signature generated by the app admin account. For details, see Generating UserSig.
identifier	The app admin account.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
random	A random 32-bit unsigned integer.
contenttype	The value is always json .

### **Maximum Call Frequency**

100 times/second

#### Sample Request Packets





```
{
    "UserAttrs": [
        {
            "To_Account": "xiaojun013",
            "Attrs": [
               "sex",
               "city"
        ]
        },
        {
            "To_Account": "xiaojun012",
        }
}
```



```
"Attrs": [
"sex",
"city"
]
}
]
```

### **Request Fields**

Field	Туре	Required	Description
To_Account	String	Yes	Target user account.
Attrs	Array	Yes	Attribute set. Note that you only need to specify the attribute names here. For more information on the format and meanings of Attrs , see Setting Application Attribute Names.

### Sample Response Packets





```
{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0
}
```

### **Response Fields**

Field	Туре	Description



ActionStatus	String	Processing result. OK : succeeded. FAIL : failed.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error information.

# **Error Codes**

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information. For common error codes (60000 to 79999), see Error Codes.

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the request packets meet JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
90033	Invalid attribute.
91000	Internal service error. Try again.

# API Debugging Tool

Use the RESTful API online debugging tool to debug this API.

## References

Pushing to All/Tagged Users Setting Application Attribute Names Obtaining Application Attribute Names Setting User Attributes Deleting User Attributes Obtaining User Attributes Obtaining User Tags



Adding User Tags Deleting User Tags Deleting All User Tags Recalling Push

# **Obtaining User Tags**

Last updated : 2024-07-16 11:40:03

# Feature Overview

This API is used by the administrator to obtain user tags. Up to 100 users' tags can be obtained at a time.

#### Sample Request URL





https://xxxxxx/v4/timpush/get\_tag?usersig=xxx&identifier=admin&sdkappid=88888888&ra

#### **Request Parameters**

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
ХХХХХХ	The dedicated domain name corresponding to the country/region of the SDKAppID. China: console.tim.qq.com

	Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com
v4/timpush/get_tag	Request API.
usersig	The signature generated by the app admin account. For details, see Generating UserSig.
identifier	The app admin account.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
random	A random 32-bit unsigned integer.
contenttype	The value is always json .

### **Maximum Call Frequency**

100 times/second

#### Sample Request Packets





### **Request Fields**

Field	Туре	Required	Description



#### Sample Response Packets



```
{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0,
    "UserTags": [
        {
```

```
"To_Account": "xiaojun012",

"Tags": ["a", "b"]

},

{

"To_Account": "xiaojun013",

"Tags": ["a", "c"]

}

]
```

### **Response Fields**

Field	Туре	Description
ActionStatus	String	Processing result. OK : succeeded. FAIL : failed.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error information.
UserTags	Array	List of user tags.
To_Account	String	User account.
Tags	Array	Tag content.

# Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode

and ErrorInfo in the response packets represent the actual error code and error information. For common error codes (60000 to 79999), see Error Codes.

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the request packets meet JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error. Try again.

# **API** Debugging Tool

Use the RESTful API online debugging tool to debug this API.

# References

Pushing to All/Tagged Users Setting Application Attribute Names Obtaining Application Attribute Names Setting User Attributes Deleting User Attributes Obtaining User Attributes Obtaining User Tags Adding User Tags Deleting User Tags Deleting All User Tags Recalling Push
# Adding User Tags

Last updated : 2024-07-16 11:40:03

# Feature Overview

This API is used by the administrator to add tags for users.

#### Note:

Up to 100 users' tags can be added at a time. In the request body, a maximum of 10 tags can be added for each user. A maximum of 100 tags can be set for each user. If a user has more than 100 tags, you need to delete old tags before adding new ones for the user.

The maximum number of tags that can be set for an application is 1,000, that is, the deduplicated total number of tags for all users is up to 1,000.

The maximum length of a single tag is 50 bytes.

#### Sample Request URL





https://xxxxxx/v4/timpush/add\_tag?usersig=xxx&identifier=admin&sdkappid=88888888&kra

#### **Request Parameters**

Parameter	Description		
https	The request protocol is HTTPS and the request method is POST.		
XXXXXX	The dedicated domain name corresponding to the country/region of the SDKAppID. China: console.tim.qq.com		

	Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/timpush/add_tag	Request API.
usersig	The signature generated by the app admin account. For details, see Generating UserSig.
identifier	The app admin account.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
random	A random 32-bit unsigned integer.
contenttype	The value is always json .

### Maximum Call Frequency

100 times/second

### Sample Request Packets





```
{
    "UserTags": [
        {
            "To_Account": "xiaojun012",
            "Tags": ["a", "b"]
        },
        {
            "To_Account": "xiaojun013",
            "Tags": ["a", "b"]
        }
    ]
```



#### }

### **Request Fields**

Field	Туре	Required	Description
To_Account	String	Yes	Target user account.
Tags	Array	Yes	Tag set.

### Sample Response Packets





```
{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0
}
```

### **Response Fields**

Field	Туре	Description



ActionStatus	String	Processing result. OK : succeeded. FAIL : failed.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error information.

# **Error Codes**

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information. For common error codes (60000 to 79999), see Error Codes.

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the request packets meet JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error. Try again.

# **API Debugging Tool**

Use the RESTful API online debugging tool to debug this API.

## References

Pushing to All/Tagged Users Setting Application Attribute Names Obtaining Application Attribute Names Setting User Attributes Deleting User Attributes Obtaining User Attributes Obtaining User Tags Adding User Tags Deleting User Tags Deleting All User Tags



**Recalling Push** 

# **Deleting User Tags**

Last updated : 2024-07-16 11:40:03

# Feature Overview

This API is used by the administrator to delete tags for users. Up to 100 users' tags can be deleted at a time.

#### Sample Request URL





https://xxxxxx/v4/timpush/remove\_tag?usersig=xxx&identifier=admin&sdkappid=888888888

#### **Request Parameters**

Parameter	Description		
https	The request protocol is HTTPS and the request method is POST.		
ХХХХХХ	The dedicated domain name corresponding to the country/region of the SDKAppII China: console.tim.qq.com		

	Singapore: adminapisgp.im.qcloud.com		
	Seoul: adminapikr.im.qcloud.com		
	Silicon Valley: adminapiusa.im.gcloud.com		
	Jakarta: adminapiidn.im.qcloud.com		
v4/timpush/remove_tag	Request API.		
usersig	The signature generated by the app admin account. For details, see Generating UserSig.		
identifier	The app admin account.		
sdkappid	The SDKAppID assigned by the Chat console when an application is created.		
random	A random 32-bit unsigned integer.		
contenttype	The value is always json .		

### **Maximum Call Frequency**

100 times/second

### Sample Request Packets





```
{
    "UserTags": [
        {
            "To_Account": "xiaojun012",
            "Tags": ["a", "b"]
        },
        {
            "To_Account": "xiaojun013",
            "Tags": ["a", "b"]
        }
    ]
```



#### }

### **Request Fields**

Field	Туре	Required	Description
To_Account	String	Yes	Target user account.
Tags	Array	Yes	Tag set.

### Sample Response Packets





```
{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0
}
```

### **Response Fields**

Field	Туре	Description



ActionStatus	String	Processing result. OK : succeeded. FAIL : failed.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error information.

# **Error Codes**

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information. For common error codes (60000 to 79999), see Error Codes.

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the request packets meet JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error. Try again.

# API Debugging Tool

Use the RESTful API online debugging tool to debug this API.

# References

Pushing to All/Tagged Users Setting Application Attribute Names Obtaining Application Attribute Names Setting User Attributes Deleting User Attributes Obtaining User Attributes Obtaining User Tags Adding User Tags Deleting User Tags



Deleting All User Tags Recalling Push

# **Deleting All User Tags**

Last updated : 2024-07-16 11:40:03

# Feature Overview

This API is used by the administrator to delete all tags for users. Up to 100 users' tags can be deleted at a time.

#### Sample Request URL





https://xxxxxx/v4/timpush/clear\_all\_tags?usersig=xxx&identifier=admin&sdkappid=8888

#### **Request Parameters**

Parameter	Description
https	The request protocol is HTTPS and the request method is POST.
XXXXXX	The dedicated domain name corresponding to the country/region of the SDKAppID.

	China: console.tim.qq.com		
	Singapore: adminapisgp.im.qcloud.com		
	Seoul: adminapikr.im.qcloud.com		
	Frankfurt: adminapiger.im.qcloud.com		
	Silicon Valley: adminapiusa.im.qcloud.com		
	Jakarta: adminapiidn.im.qcloud.com		
v4/timpush/clear_all_tags	Request API.		
usersig	The signature generated by the app admin account. For details, see Generating UserSig.		
identifier	The app admin account.		
sdkappid	The SDKAppID assigned by the Chat console when an application is created.		
random	A random 32-bit unsigned integer.		
contenttype	The value is always json .		

### Maximum Call Frequency

100 times/second

### Sample Request Packets





### **Request Fields**

Field	Туре	Required	Description

### Sample Response Packets

🔗 Tencent Cloud



### **Response Fields**

Field	Туре	Description
ActionStatus	String	Processing result. OK : succeeded. FAIL : failed.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error information.

# Error Codes

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information. For common error codes (60000 to 79999), see Error Codes.

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the request packets meet JSON specifications.
90009	The request requires app admin permissions.
90018	The number of requested accounts exceeds the limit.
91000	Internal service error. Try again.

# API Debugging Tool

Use the RESTful API online debugging tool to debug this API.

## References

Pushing to All/Tagged Users Setting Application Attribute Names Obtaining Application Attribute Names Setting User Attributes Deleting User Attributes Obtaining User Attributes Obtaining User Tags Adding User Tags Deleting User Tags Deleting All User Tags Recalling Push

# **Recalling Push**

Last updated : 2024-07-16 11:40:03

If there is an error in the content pushed to all/tagged users, it will have a negative impact on the product when end users view or click the content. In this case, it is necessary to handle it in a timely manner. You can choose to recall the push.

# Feature Overview

Termination: The delivery of push tasks takes some time, and the delivery to accounts for which the tasks have not been delivered will be terminated.

Recall: For accounts for which the tasks have been delivered, unread and roaming messages can be recalled.

Overwrite: If an account has received an offline push, that push can be overwritten.

This API supports the the termination, recall, and overwrite of push tasks for all or tagged users. In the following, termination/recall/overwrite will be referred to as recall by default.

The validity period for recall is 24 hours, calculated from the time of task initiation. Push tasks that exceed 24 hours cannot be recalled.

# **API Call Description**

### Sample Request URL





https://xxxxxx/v4/timpush/revoke?usersig=xxx&identifier=admin&sdkappid=88888888&kran

#### **Request Parameters**

Parameter	Description	
https	The request protocol is HTTPS and the request method is POST.	
XXXXXX	The dedicated domain name corresponding to the country/region of the SDKAppID. China: console.tim.qq.com	

	Singapore: adminapisgp.im.qcloud.com Seoul: adminapikr.im.qcloud.com Frankfurt: adminapiger.im.qcloud.com Silicon Valley: adminapiusa.im.qcloud.com Jakarta: adminapiidn.im.qcloud.com
v4/timpush/revoke	Request API.
usersig	The signature generated by the app admin account. For details, see Generating UserSig.
identifier	The app admin account.
sdkappid	The SDKAppID assigned by the Chat console when an application is created.
random	A random 32-bit unsigned integer.
contenttype	The value is always json .

### **Maximum Call Frequency**

1 time/second

### Sample Request Packets



Note:

1. Vendors that support offline push overwrite: APNS/Google FCM/Huawei/Honor. Offline push from other vendors does not support overwrite. (Google FCM's notification mode supports overwrite, while the data mode currently does not support.)

2. During recall, if the recipient is in the foreground and the offline push (notification message) is read by default, the offline push will not be overwritten.

#### **Request Fields**

Field	Туре	Required	Description
Taskld	String	Yes	Push task ID.
OfflinePushInfo	Object	No	The information to be pushed offline. For more information, see Message Formats. Note: If OfflinePushinfo.PushFlag=1 or OfflinePushInfo is not set, offline push will not be overwritten.

#### Sample Response Packets





```
{
    "ActionStatus": "OK",
    "ErrorInfo": "",
    "ErrorCode": 0
}
```

### **Response Fields**

Field	Туре	Description



ActionStatus	String	Processing result. OK : succeeded. FAIL : failed.
ErrorCode	Integer	Error code.
ErrorInfo	String	Error information.

# **Error Codes**

Unless a network error (such as error 502) occurs, the HTTP return code for this API is always 200. ErrorCode and ErrorInfo in the response packets represent the actual error code and error information. For common error codes (60000 to 79999), see Error Codes.

The following table describes the error codes specific to this API:

Error Code	Description
90001	Failed to parse the JSON format. Check whether the request packets meet JSON specifications.
90009	The request requires app admin permissions.
90049	Recall TaskId is illegal and there is no push record. The returned TaskId can only be used for recall when the task is pushed through the timpush/push API.
90050	Repeated recall. Push tasks that have already been recalled cannot be called repeatedly.
90051	Recall is too frequent. The recall frequency limit is 1 time per second.
90052	The recall validity period has been exceeded. The recall must be within 24 hours, and push tasks that exceed 24 hours cannot be recalled.
90053	Invalid recall. Push with OnlineOnlyFlag=0, but the recall does not include OfflinePushInfo.
91000	Internal service error. Try again.

# **API Debugging Tool**

Use the RESTful API online debugging tool to debug this API.

# References

### Pushing to All/Tagged Users

Setting Application Attribute Names Obtaining Application Attribute Names Setting User Attributes Deleting User Attributes Obtaining User Tags Adding User Tags Deleting User Tags Deleting All User Tags

**Recalling Push** 

# Advanced Features Custom Definition Badge

Last updated : 2024-08-01 11:33:30

# Android

#### **Supported Vendors**

Huawei.

#### **Configuration Method**

To configure the Huawei badge parameters in the console, set them to the application's startup class, for example, "com.tencent.qcloud.tim.demo.SplashActivity". The component will automatically parse and update the badge; otherwise, it will not update the badge.

Add Huawei cer	tificate	
Package Name *	Enter Package Name	How to generate a Huawei certificate?
AppID *	Enter AppID	
Category	Enter category	
AppSecret *	Enter AppSecret	
ChannellD	Enter a channel ID	
Badge Parameter	Please enter the badge paramete	
	*Note: It is available only for Chat SD	K 4.8 or later.
Response after Click	Open application Open we	bpage Open specified in-app page
Specified In-app	intent://com.tencent.qcloud.unia	
Page *	The click notification bar event will be broadcast or callback. The App can h	e called back here and given to the applic nandle opening the application in the call
	Co	nfirm

## iOS

By default, when the App goes into the background, the IMSDK will set the total number of unread IM messages as the badge. If the App is integrated with offline push, when a new offline push notification is received, the App badge will increment by 1 based on the baseline badge (default is the total number of unread IM messages, or the custom-defined badge if one has been set).

### **Configuration Method**

If you want to customize the badge, follow these steps:

1. The App calls the - (void)setAPNSListener:(id<V2TIMAPNSListener>)apnsListener interface to set the listener.

2. The App implements the - (uint32\_t)onSetAPPUnreadCount interface and returns the custom-defined badge number.



```
// 1. Set the listener
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSD
    // Listen for push notifications
    [V2TIMManager.sharedInstance setAPNSListener:self];
    // Listen for unread conversation counts
    [V2TIMManager sharedInstance] setConversationListener:self];
```

```
Chat
   return YES;
}
// 2. Save the unread count after it changes
- (void) on Total Unread Message Count Changed: (UInt 64) total Unread Count {
    self.unreadNumber = totalUnreadCount;
}
// 3. Report custom-defined unread count after the app is pushed to the background
/** After the application enters the background, customize the app's unread count.
 *
   *
 *
    - (uint32_t)onSetAPPUnreadCount {
 *
         return 100; // Custom-defined unread count
 *
    }
 *
   */
- (uint32_t) onSetAPPUnreadCount {
    // 1. Get the custom-defined badge
    uint32_t customBadgeNumber = ...
    // 2. Add the IM message unread count
    customBadgeNumber += self.unreadNumber;
    // 3. Report to the IM server via IMSDK
   return customBadgeNumber;
}
```

# Flutter

For configuration, please refer to the iOS and Android sections below. The methods called are also available in the Flutter version of the IM SDK, with the same names.

# **Custom Definition Ringtone**

Last updated : 2024-08-01 11:36:20

# Android

### Pre-8.0 systems

1. Custom ringtone resource files for Android should be added to the project's raw directory; for iOS, link them into the Xcode project.

2. Please use the message invocation interfaces setAndroidSound() and setIOSSound().




```
V2TIMOfflinePushInfo v2TIMOfflinePushInfo = new V2TIMOfflinePushInfo();
v2TIMOfflinePushInfo.setAndroidSound("Ringtone Name");
v2TIMOfflinePushInfo.setIOSSound("Ringtone Name.mp3");
```

```
String msgID = V2TIMManager.getMessageManager().sendMessage(v2TIMMessage, isGroup ?
V2TIMMessage.V2TIM_PRIORITY_DEFAULT, false, v2TIMOfflinePushInfo, new V2TIMSendCal
@Override
public void onProgress(int progress) {
```

```
}
```

```
@Override
public void onError(int code, String desc) {
    }
    @Override
    public void onSuccess(V2TIMMessage v2TIMMessage) {
    }
});
```

#### Note:

Supported in IMSDK v6.1.2155 or above.

The interface supports Huawei, Xiaomi, FCM, and APNS.

#### System version 8.0 and later

#### Huawei and APNs

Huawei, APNS use setAndroidSound() and setIOSSound() to set the offline push ringtone prompt.

#### OPPO

1. Locally create a custom notification channel





```
©2013-2022 Tencent Cloud. All rights reserved.
```

```
notificationChannel.setSound(Uri.parse("sound"), null);
nm.createNotificationChannel(notificationChannel);
```

2. Use the created channel

#### Create Private Message Channel - .

Push messages carry the channel ID field. For details, see setAndroidOPPOChannelID. For console settings, see Certificate Editing ChannelID field. Setting one of the two is sufficient.

#### Mi

}

1. log in to the Manufacturer Console to create a channel and configure, where the ringtone file needs to be added to the raw directory of your local Android Studio project.

通知类别名称	通知美则的名称			
	通知类别名称不允许出现数字,长	度不超过20个字符		
通知类别描述	描述该通知类别推送的内容,方便用户理解			
	字符长度在40个字符以内			
消息分类	新闻资讯		~	
自定义铃声 (sound_url) 💿 自定义铃声 💦 🔿 使用系统铃声				
	请输入用于通知铃声的声音文件存放的url			
url需满足android.resource://your packagename/XXX/XXX, 且创建后不允许修改url				
消息内容使用场景	消息内容使用场景			
消息内容示例	消息标题	消息内容	○ 请输入内容示例	
	+ 添加			
		т ужин		
联系邮箱	请输入工作邮箱,用于按收审核	- 7 / / / / / / / / / / / / / / / / / /		

2. Send a message specifying the channel ID of the custom ringtone; for details, please refer to setAndroidXiaoMiChannelID.





```
V2TIMOfflinePushInfo v2TIMOfflinePushInfo = new V2TIMOfflinePushInfo();
v2TIMOfflinePushInfo.setAndroidXiaoMiChannelID("Channel ID Applied by Manufacturer"
String msgID = V2TIMManager.getMessageManager().sendMessage(v2TIMMessage, isGroup ?
V2TIMMessage.V2TIM_PRIORITY_DEFAULT, false, v2TIMOfflinePushInfo, new V2TIMSendCal
@Override
public void onProgress(int progress) {
   TUIChatUtils.callbackOnProgress(callBack, progress);
  }
@Override
```

Chat

```
public void onError(int code, String desc) {
   TUIChatUtils.callbackOnError(callBack, TAG, code, desc);
}
@Override
public void onSuccess(V2TIMMessage v2TIMMessage) {
});
```

#### FCM

1. Custom ringtone resource files should be added to the project's raw directory. To configure FCM's custom ringtone parameters, call before registering for push services. For details, see configFCMPrivateRing.

2. Send a message specifying the channel ID for the custom ringtone; for details, please refer to

setAndroidFCMChannelID.





```
V2TIMOfflinePushInfo v2TIMOfflinePushInfo = new V2TIMOfflinePushInfo();
v2TIMOfflinePushInfo.setAndroidFCMChannelID(PrivateConstants.fcmPushChannelId);
String msgID = V2TIMManager.getMessageManager().sendMessage(v2TIMMessage, isGroup ?
V2TIMMessage.V2TIM_PRIORITY_DEFAULT, false, v2TIMOfflinePushInfo, new V2TIMSendCal
@Override
public void onProgress(int progress) {
   TUIChatUtils.callbackOnProgress(callBack, progress);
  }
@Override
```

```
©2013-2022 Tencent Cloud. All rights reserved.
```

```
public void onError(int code, String desc) {
   TUIChatUtils.callbackOnError(callBack, TAG, code, desc);
}
@Override
public void onSuccess(V2TIMMessage v2TIMMessage) {
});
```

#### Note:

Supported in IMSDK v7.0.3754 or above.

FCM custom ringtones or setting channel ID is supported only in Certificate Mode.

### iOS

1. When calling sendMessage to send a message, set the V2TIMOfflinePushInfo's iOSSound field. Pass in the voice file name to iOSSound .

#### Note:

Offline push notification sound settings (effective only for iOS), when `iOSSound = kIOSOfflinePushNoSound`, it indicates no sound will be played upon receiving.

When `iOSSound = kIOSOfflinePushDefaultSound`, it indicates the system sound will be played upon receiving. To customize `iOSSound`, you first need to link the audio file into the Xcode project, and then set the audio filename (including the extension) to `iOSSound`.





```
V2TIMOfflinePushInfo *pushInfo = [[V2TIMOfflinePushInfo alloc] init];
pushInfo.title = @"push title";
pushInfo.iOSSound = @"phone_ringing.mp3"; // your voice file's name
[[V2TIMManager sharedInstance] sendMessage:msg receiver:receiver groupID:groupID pr
} fail:^(int code, NSString *msg) {
}];
```

2. When calling sendMessage to send a message, set the V2TIMOfflinePushInfo's AndroidSound field; pass in the voice file name to AndroidSound .

#### Note:

Offline push notification sound settings (effective only for Android, supported only in imsdk 6.1 and above) are supported only on Huawei and Google phones for setting ringtone prompts.

For Xiaomi ringtone settings, please refer to: Server-side Java SDK documentation.

To customize `AndroidSound`, you first need to place the audio file in the raw directory of the Android project, and then set the `AndroidSound` with the audio filename (without the extension).



```
V2TIMOfflinePushInfo *pushInfo = [[V2TIMOfflinePushInfo alloc] init];
```

### 🔗 Tencent Cloud

```
pushInfo.title = @"push title";
pushInfo.AndroidSound = @"phone_ringing"; // your voice file's name
[[V2TIMManager sharedInstance] sendMessage:msg receiver:receiver groupID:groupID pr
} fail:^(int code, NSString *msg) {
}];
```

### uniapp

#### Note:

The receiver needs to integrate TencentCloud-TIMPush. For the sender, @tencentcloud/chat ≥ 3.3.2. Supports Huawei, Xiaomi, FCM, and APNS.

#### Receiver

Android

iOS

Custom ringtone resource files should be added to the project **nativeResources/android/res/raw** directory, as shown:



#### Note:

For iOS custom ringtones, the uniapp must be the official package.

Custom ringtone resource files should be added to the project **nativeResources**/ios/Resources directory, as shown:



#### Sender

1. Upgrade @tencentcloud/chat to the latest version.





npm install @tencentcloud/chat@latest

web

uniapp

Check the TencentCloudChat.VERSION in the browser console to confirm @tencentcloud/chat ≥ 3.3.2 as shown:



Check the TencentCloudChat.VERSION in the HBuilder logs to confirm @tencentcloud/chat ≥ 3.3.2 as shown:



#### 2. Send a message, set related parameters for the custom ringtone in offlinePushInfo.

#### Note:

On Xiaomi phones with Android 8.0 and above, setting androidInfo.XiaoMiChannelID is mandatory, please refer to:Xiaomi Custom Ringtone.

Google Phone FCM Push on Android 8.0 and above systems requires setting androidInfo.FCMChannelID for sound notifications.

Integration (UI Included)

Integration Without UI

Note:



When apnsInfo.sound = TUIChatEngine.TYPES.IOS\_OFFLINE\_PUSH\_NO\_SOUND, it means no sound will be played upon receiving.

When apnsInfo.sound = TUIChatEngine.TYPES.IOS\_OFFLINE\_PUSH\_DEFAULT\_SOUND, it means the system sound will be played upon receiving.

When using TUIChatService in UIKit to send messages, set related offlinePushInfo parameters. For example, to send a regular text message, the code is as follows:







```
// Send Plain Text Message
let promise = TUIChatService.sendTextMessage(
{
    payload: { text: 'Hello world!' }
},
{
    // If the recipient is offline, the message will be stored for roaming and an off
    offlinePushInfo: {
        androidInfo: { // Android Push Configuration
            sound: 'private_ring.mp3', // Custom ringtone for Android
        XiaoMiChannelID: '', // On Xiaomi Mobile Phone with Android 8.0 and above,
```

```
FCMChannelID: '', // Google Phone FCM on Android 8.0 and above systems requ
},
apnsInfo: { // APNs Push Configuration
// apnsInfo.sound = TUIChatEngine.TYPES.IOS_OFFLINE_PUSH_NO_SOUND, indicates n
// apnsInfo.sound = TUIChatEngine.TYPES.IOS_OFFLINE_PUSH_DEFAULT_SOUND, indica
sound: 'private_ring.mp3', // Custom ringtone for iOS
}

promise.catch((error) => {
// Business side can handle errors by catching exceptions through promise.catch
});
```

Reference Documentation:

uikit - TUIChatService documentation related to sending messages

#### Note:

When apnsInfo.sound = TencentCloudChat.TYPES.IOS\_OFFLINE\_PUSH\_NO\_SOUND, it means no sound will be played upon receiving.

When apnsInfo.sound = TencentCloudChat.TYPES.IOS\_OFFLINE\_PUSH\_DEFAULT\_SOUND, it means the system sound will be played upon receiving.

When sending a message in chat, set the related fields for offlinePushInfo as follows:







```
// Message Sending Options
chat.sendMessage(message, {
    // If the recipient is offline, the message will be stored for roaming and an off
    offlinePushInfo: {
        androidInfo: { // Android Push Configuration
            sound: 'private_ring.mp3', // Custom ringtone for Android
        XiaoMiChannelID: '', // On Xiaomi Mobile Phone with Android 8.0 and above,
        FCMChannelID: '', // Google Phone FCM on Android 8.0 and above systems requ
        },
        apnsInfo: { // APNs Push Configuration
        // apnsInfo.sound = TencentCloudChat.TYPES.IOS_OFFLINE_PUSH_NO_SOUND, indicate
```

```
// apnsInfo.sound = TencentCloudChat.TYPES.IOS_OFFLINE_PUSH_DEFAULT_SOUND, ind
    sound: 'private_ring.mp3', // Custom ringtone for iOS
    }
});
```

Reference document: Chat SDK - sendMessage documentation

### Flutter

#### Note:

The interface supports Huawei, Xiaomi, FCM, and APNS.

Custom ringtone resource files for Android should be added to the project's raw directory; for iOS, link them into the Xcode project.

When calling sendMessage to send a message, set the fields offlinePushInfo, iOSSound , and androidSound . For specific manufacturer configurations, please refer to the content of the Android and iOS modules below. The methods called are all available under the same name in the Flutter version of the IM SDK.

# **Customized Icon**

Last updated : 2024-06-13 10:21:45

## Android

#### **Supported Vendors**

Huawei and Google FCM support Customized Icon; other manufacturers do not support Customized Icon, and by default, use the App Icon.

#### **Configuration Method**

#### Method 1

The component will automatically parse and update the Customized Icon related to Application Project Configuration; otherwise, it will use the default App Icon of the application. **Huawei**:huawei\_private\_icon.png **FCM**:fcm\_private\_icon.png



#### Method 2

Effective when configured in the main project's Manifest File:

Huawei

Google FCM





<meta-data android:name="com.huawei.messaging.default\_notification\_icon" android:resource="@drawable/Icon Resource Name" />





```
<!-- [START fcm_default_icon] -->
<!-- Set custom default icon. This is used when no icon is set for incoming notific
    See README(https://goo.gl/l4GJaQ) for more. -->
<meta-data
    android:name="com.google.firebase.messaging.default_notification_icon"
    android:resource="@drawable/Icon Resource Name" />
<!-- Set color used with incoming notification messages. This is used when no color
    notification message. See README(https://goo.gl/6BKBk7) for more. -->
<meta-data
    android:name="com.google.firebase.messaging.default_notification_color"</pre>
```

```
android:resource="@android:color/white" />
<!-- [END fcm_default_icon] -->
```

#### Note:

FCM Icon Requirements:

Small icon must be a PNG image with Alpha transparency channel.

Background must be transparent.

Avoid leaving excessive padding around the icon.

It is recommended to use 46 x 46px uniformly. Smaller images will be fuzzy, while larger ones will be automatically scaled down by the system.

### iOS

Custom definitions are not supported; the App Icon is used by default.

# **Custom Definition Click Redirect**

Last updated : 2024-06-13 10:21:45

### Flutter

#### **Step 1: Manufacturer Configuration**

Refer to Manufacturer Configuration > Flutter, complete the manufacturer configuration. Please note, for subsequent actions after clicking, use the default configuration.

Package Name *	Enter Package Name	How to generate a Huawei certificate?	
AppID *	Enter AppID		
Category	Enter category		
AppSecret *	Enter AppSecret		
ChannelID	Enter a channel ID		
Badge Parameter	Please enter the badge paramete		
	*Note: It is available only for Chat SD	K 4.8 or later.	
Response after Click	Open application Open we	opage Open specified in-app page	
Specified In-app	intent://com.tencent.qcloud.uniar		
Page *	The click notification bar event will be called back here and given to the application through broadcast or callback. The App can handle opening the application in the callback by itself		

#### **Step 2: Client Code Configuration**

Refer to Client Code Configuration, complete the configuration.

#### Step 3: Handle Message Click Callback and Parse Parameters

Please define a function for receiving push message click callback events.

**Define the function with the parameter format** {required String ext, String? userID, String? groupID} .

Here, the ext field contains the full ext information carried by the message, specified by the sender. If not specified, there is a default value. You can use this to navigate to the corresponding page based on parsing this field. The userID and groupID fields, for this plugin, attempt to automatically parse the ext Json String and obtain the individual chat partner userID and group chat groupID information carried within. If you haven't defined the ext field yourself, and the ext field is specified by default by the SDK or UIKit, you can use the default parsing here. If the parsing attempt fails, they will be null.

You can define a function to receive this callback and use it to navigate to the corresponding Session Page or your Business Page.

Example below:



```
void _onNotificationClicked({required String ext, String? userID, String? groupID})
print("_onNotificationClicked: $ext, userID: $userID, groupID: $groupID");
if (userID != null || groupID != null) {
    // Navigate to the corresponding Message Page based on userID or groupID.
    } else {
        // Use your own parsing method based on the ext field to navigate to the corres
    }
}
```

#### Step 4: Mount the Listen for Callbacks

After completing the IM Login, and before using other plugins (such as CallKit), immediately register the Push Plugin.

Invoke the TencentCloudChatPush().registerPush method, pass in the Click Callback Function defined in the background.

Additionally, you can optionally pass in the <code>apnsCertificateID</code> for the iOS Push Certificate ID and

androidPushOEMConfig for Android Push Vendor Configuration. These configurations have been specified in previous steps. If no changes are necessary, they need not be passed in again.



TencentCloudChatPush().registerPush(onNotificationClicked: \_onNotificationClicked);

#### Note:

If your application needs to use the **push plugin for business message notifications**, and it does not immediately start and log in to the IM module after launching<3>, or you need to handle business navigation through message click callbacks before <5>logging in to the IM module</5>, it is recommended that you call the

<7>TencentCloudChatPush().registerOnNotificationClickedEvent</7> method as soon as possible to manually mount the message click callback, in order to timely retrieve message parameters.

In such scenarios, you can execute this function before calling TencentCloudChatPush().registerPush and place it as early as possible in your code.



TencentCloudChatPush().registerOnNotificationClickedEvent(onNotificationClicked: \_o

## Android

After receiving an offline push, the notification bar will display the push message as shown. Clicking on the notification bar can custom open the application interface.

1. For console configuration, click on 'Subsequent Actions' and configure as follows, choose **Open the specified interface within the app**, and the plugin users will by default fill in the jump parameters.

Package Name *	Enter Package Name	How to generate a Huawei certificate?	
AppID *	Enter AppID		
Category	Enter category		
AppSecret *	Enter AppSecret		
ChannellD	Enter a channel ID		
Badge Parameter	Please enter the badge paramete		
	*Note: It is available only for Chat SD	K 4.8 or later.	
Response after Click	Open application Open we	bpage 🛛 Open specified in-app page	
Specified In-app	intent://com.tencent.qcloud.uniar		
Page *	The click notification bar event will be called back here and given to the application th broadcast or callback. The App can handle opening the application in the callback by		

2. For registration and callback handling of click events, it's recommended to place the registration in the application's Application oncreate() function. The component will notify the application via callback or broadcast, and the application can configure its custom jump page in the callback.

The callback method is as follows:



TUICore.registerEvent(TUIConstants.TIMPush.EVENT\_NOTIFY, TUIConstants.TIMPush.EVENT @Override public void onNotifyEvent(String key, String subKey, Map<String, Object> pa Log.d(TAG, "onNotifyEvent key = " + key + "subKey = " + subKey); if (TUIConstants.TIMPush.EVENT\_NOTIFY.equals(key)) { if (TUIConstants.TIMPush.EVENT\_NOTIFY\_NOTIFICATION.equals(subKey))



The broadcast method is as follows:



// Dynamic Broadcast Registration

```
IntentFilter intentFilter = new IntentFilter();
intentFilter.addAction(TUIConstants.TIMPush.NOTIFICATION_BROADCAST_ACTION);
LocalBroadcastManager.getInstance(context).registerReceiver(localReceiver, intentFi
// Broadcast Receiver
public class OfflinePushLocalReceiver extends BroadcastReceiver {
    public static final String TAG = OfflinePushLocalReceiver.class.getSimpleName()
    QOverride
    public void onReceive(Context context, Intent intent) {
        DemoLog.d(TAG, "BROADCAST_PUSH_RECEIVER intent = " + intent);
        if (intent != null) {
            String ext = intent.getStringExtra(TUIConstants.TIMPush.NOTIFICATION_EX
            TUIUtils.handleOfflinePush(ext, null);
        } else {
            DemoLog.e(TAG, "onReceive ext is null");
        }
    }
}
```

### iOS

Please set the V2TIMOfflinePushInfo's ext field when calling sendMessage to send a message. When the user receives an offline push and starts the App, they can get the ext field in the AppDelegate -> didReceiveRemoteNotification system callback, and then jump to the specified UI interface according to the content of the ext field.

Taking the scenario of **Denny sending a message to Vinson** as an example.

The sender, Denny, needs to set the push extension field ext when sending the message:





```
// Denny sets offlinePushInfo and specifies the ext field when sending the message
V2TIMMessage *msg = [[V2TIMManager sharedInstance] createTextMessage:@"Text Message
V2TIMOfflinePushInfo *info = [[V2TIMOfflinePushInfo alloc] init];
info.ext = @"jump to denny";
[[V2TIMManager sharedInstance] sendMessage:msg receiver:@"vinson" groupID:nil prior
onlineUserOnly:NO offlinePushInfo:info progress:^(uint32_t progress) {
} succ:^{
} fail:^(int code, NSString *msg) {
}];
```
Recipient: Vinson's App may be offline, but can still receive APNS offline push. When Vinson clicks on the push message, the App will launch:





# **Push Message Categorization**

Last updated : 2024-06-13 10:21:45

Every vendor has a message categorization mechanism, and different types have different push strategies. If the push requirement is of the IM type and timely delivery is desired, it is necessary to set your application as the corresponding push type according to vendor rules. It will be categorized as a system message type or an important message type with high priority. Conversely, offline push will not be pushed to the device in a timely manner.

#### Note:

Only IM type messages need to be configured as system message types or important message types for timely delivery. Some marketing or advertising type pushes, which do not require immediate delivery, can arrive at the device within a certain period, and do not need to be configured as a system message type.

Vendors have restrictions on the daily push quantity and push frequency for applications, which can be viewed in the vendor console regarding the daily limit on push quantity and restrictions.

Do not arbitrarily configure message types. Configuration that does not meet the standard may result in the vendor freezing your account.

### Huawei

Starting from EMUI 10.0, Huawei Push intelligently categorizes notification messages into two levels: **Service and Communication** and **Information Marketing**. Versions earlier than EMUI 10.0 did not categorize notifications, having only one level where all messages were displayed through the **Default Notifications** channel, equivalent to Service and Communication on EMUI 10.0. The daily push quantity for Information Marketing messages will be subject to an upper limit management based on the application type from January 5, 2023, while Service and Communication messages will not be limited in daily push quantity.

#### **Customized Self-Categorization Push Method**

Apply for Self-Categorization Benefits.

Push messages carry the category field, for details see setAndroidHuaWeiCategory. For console settings, see the Category field in Certificate Editing. Setting either one is sufficient.

For more information, see Message Categorization Standards or Push Quantity Management Rules.

# HONOR

Honor phone push is related to the system version.

Currently, the Honor channel only supports Honor devices in China with Magic UI 4.0 or above and those overseas with Magic UI 4.2 or above.

Honor devices below the versions mentioned can use the Huawei Manufacturer for push access. For more information, see Product Description.

### vivo

Push messages are classified into system messages and operational messages, with different push effectiveness and policies. System messages also undergo a secondary correction through the vendor's intelligent classification; if it's incorrectly identified as a non-system message, it will automatically be corrected as an operational message. False judgments can be corrected by email application feedback. Moreover, message pushing is also subject to a daily limit, determined by the app's subscription statistics with the manufacturer.

#### **Customized Self-Categorization Push Method**

Push messages carry the category field. For details, see setAndroidVIVOCategory. For console settings, refer to Certificate Editing Category field. Setting one of the two is sufficient. For more details, see Push Message Classification Description or Push Message Restriction Description.

# OPPO

Push messages are classified into private messages and public messages with different push effectiveness and policies. Private messages target users who pay a certain level of attention and wish to receive information promptly. Application for private message channel benefits requires an email application. The number of pushes via the public trust channel is limited.

#### **Customized Self-Categorization Push Method**

#### Create Private Message Channel.

Push messages carry the channel ID field. For details, see setAndroidOPPOChannelID. For console settings, see Certificate Editing ChannelID field. Setting one of the two is sufficient. For specific details, see Message Classification Description or Push Service Restriction Explanation.

### Mi

Push messages are divided into "Private Message" and "Public Trust Message" categories, with the default channel being Public Trust Message. The daily push quantity of Public Trust Messages will be subject to upper limit management. Public Trust Messages are suitable for pushing Hot News, New Product Promotions, Platform Announcements, Community Topics, Prize-Winning Activities, etc., mostly content of universal interest to users. Private Messages are suitable for pushing Chat Messages, Personal Order Changes, Courier Notifications, Transaction Reminders, IoT System Notifications, etc., relating to private notifications, with no restrictions on the number of notification messages pushed. Implementation of Message Classification Management requires channel application and access in the Manufacturer Console.

#### Custom Method for Self-Classified Push:

#### Channel Application and Access.

Push messages carry the channel ID field. For details, see setAndroidXiaoMiChannelID. For console settings, see Certificate Editing ChannelID field. Setting one of the two is sufficient.

For details, see Push Message Classification New Rules or Push Message Restriction Description.

# Meizu

The number of push messages is limited. For details, see Push Access Guide.

# FCM

The frequency of push uplink messages is limited.

For details, see Message Frequency Limit.

#### Note:

The setting of push message Channel ID and the category field has two methods: API interface and console certificate settings, each with its scope of effect, and the API setting has a higher priority than the console setting.

# **Release Notes**

Last updated : 2024-08-01 12:04:20

#### TIMPush 8.0.6897 @2024.07.15

Added intelligent detection available channel strategy. Added push registration timeout protection mechanism.

Resolved small icon settings issue.

Fixed the issue where configuring the jump option to the homepage couldn't launch the application. Optimized model recogni

#### TIMPush 7.9.5668 @2024.04.09

Added OfflinePushExtInfo to support Transparent Push Feature. Upgraded Vivo push package version. Resolved occasional crash issues on OPPO and Vivo models. Fixed occasional mismatches of FCM Data mode for multiple push notifications and transparent data. Optimized WorkManager compilation issues. UniApp supports Reach Reporting, supports Process Termination Callback for click events.

#### TIMPush - Android 7.8.5484 @2024.02.02

Released TIMPush-UniApp. FCM PUSH supports Pass-through Message. Optimization of Push Registration and Reporting Logic.

#### TIMPush - iOS 7.8.5483 @2024.02.02

iOS support for Minimum version updated to 10.0 and above.

#### TIMPush 7.7.5283 @2023.12.28

Add custom Definition push configuration file feature.

Upgrade Honor Push Package.

Optimize Vivo placeholder configuration.

Fix issue where some functions did not callback after iOS hosted UNUserNotificationCenterDelegate.

#### TIMPush 7.7.5282 @2023.12.18

Upgrade Xiaomi and Meizu Push Packages.

Optimize Context occasionally empty issue.

Optimize registerPush manual invocation of API logic.

#### TIMPush 7.6.5011 @2023.11.13

Release Push Plugin.

#### Chat

# FAQS

Last updated : 2024-06-13 10:21:45

# Android

#### How do I troubleshoot if I cannot receive offline push messages?

#### **Special Case Investigation**

#### **OPPO devices**

There are several common reasons why OPPO devices do not receive push notifications:

According to the requirements of the official OPPO Push website, ChannelID must be configured on OPPO devices running Android 8.0 or above, otherwise, push notifications cannot be displayed. For the method of configuration, refer to setAndroidOPPOChannelID.

The notification bar display feature is disabled by default for applications installed on the OPPO device. If this is the case, check the switch status.

#### Google FCM

If push notifications are not received, verify whether the certificates have been correctly uploaded to the IM console. For the troubleshooting path, see the document Configuring in the IM console > Google FCM, and check against the diagram to see if it has been added correctly.

#### Xiaomi and vivo

Xiaomi and vivo: The app needs to be listed in the App Marketplace before it can be delivered through the Manufacturer Channel. It is usually indicated that the app is in the blocklist, prohibiting message sending. Or, the App has closed the push channel.

#### Sending messages as Custom Definition Messages

The offline push of Custom Definition Messages is different from ordinary messages. We cannot parse the content of Custom Definition Messages, and the push content cannot be determined. Therefore, by default, they are not pushed offline. If you have offline push requirements, you need to set the desc field of offlinePushInfo during sendMessage, and the desc information will be displayed by default during push.

#### Effects of Device Notification Bar Settings

The immediate effect of offline push is the Notification Bar Prompt, so like other notifications, it's affected by the device notification settings. Taking Huawei as an example:

Phone Settings > Notifications > Lock Screen Notifications > Hide or Do Not Display Notifications, will affect the display of offline push notifications when the screen is locked.

Phone Settings > Notifications > More Notification Settings > Status Bar Notification Icons, will affect the display of offline push notification icons in the status bar.

Phone Settings > Notifications > App Notification Management > Allow Notifications, toggling on or off will directly affect the display of offline push notifications.

Phone Settings > Notifications > App Notification Management > Notification Ringtone and Phone Settings > Notifications > App Notification Management > Silent Notifications, will affect the sound of offline push notification.

#### Message Type

Vendor push services all have message categorization mechanisms, and different types will have different push strategies. For details, please see Push Message Categorization.

#### Vendor Features

Offline push depends on the vendor's capabilities, and some simple characters may be filtered by the vendor and not be pushed through transparently. It is recommended to use meaningful content for pushing.

#### Self-service Push Troubleshooting Tool

#### Whether device push plugin integration is working properly

Test whether messages can be properly pushed offline by using the Offline Testing Tool in the IM Console.

#### Push Link Troubleshooting

Use the Troubleshooting Tool to view the details of the full push link.

#### How to troubleshoot if the jump interface is unsuccessful?

#### **Click to perform a Configuration Check**

#### See Custom Click-through Redirect for details, check:

Configure the subsequent actions after clicking in the console as follows, select **Open the specified interface within the app**, plugin users will by default fill in the jump parameters, check if the parameters have been modified. Registration and callback processing for click events. It is recommended to register during the application Application's oncreate() function, which needs to be placed early in the application lifecycle. Check if the click callback correctly handles the redirect logic.

#### **Device System Permission Restrictions**

If the application process does not exist and clicking the notification bar to jump to the application interface, it is necessary to pull the application from the background to the foreground. Some manufacturers' systems will check whether the App has enabled **background self-start** or **floating window** permissions. If not enabled, the system side will intercept and handle it, leading to failure.

Different manufacturers, and even different Android versions from the same manufacturer, offer different permissions and permission names for applications. Through testing, only the background pop-up interface permission needs to be enabled for Mi 6, while Redmi requires both the background pop-up interface and floating window display permissions, necessitating different treatments for different manufacturers.

#### **Push Link Troubleshooting**

Use the Troubleshooting Tool to view the details of the full push link.

#### Others

#### Huawei

Huawei Push Certificate ID <= 11344, using Huawei Push v2 Version Interface, does not support reach and click acknowledgment. If you need statistical data feature, please regenerate and update the Certificate ID. AndroidInfo.ExtAsHuaweiIntentParam, passing "1" means treating the pass-through content Ext as an Intent parameter, "0" means treating the pass-through content Ext as an Action parameter. The restapi using "1" currently does not support click event statistics.

#### **Statistical Data Feature**

Only the push data details of the last device logged in are recorded; multi-platform login scenarios are not supported.

The Debug version of the App feature works normally, but anomalies occur with the Release version of the App feature

This issue is very likely caused by obfuscation. You can add the following obfuscation rules:





```
-keep class com.tencent.qcloud.** { *; }
-keep class com.tencent.timpush.** { *; }
```

#### Resolving conflicts between integrating TIMPush and other competitors

The reason is that the application itself or the third-party push client it integrates or depends on conflicts with the thirdparty client in TIMPush. It is necessary to keep only one in use. For specific methods, please refer to: TIMPush Integration Conflict Resolution.

# iOS

#### Why doesn't offline push work for common messages?

First, check that the app runtime environment is the same as the certificate environment. Otherwise, offline push messages will not be received.

Next, check if the app and the certificate environment are set to production. If they are in a development environment, applying for a deviceToken from Apple might fail. This issue has not been found in the production environment, so please switch to production for testing.

#### Why doesn't offline push work for custom messages?

The offline push for custom messages differs from that for regular messages. We cannot parse the content of custom messages, making it impossible to determine the push content. Therefore, custom messages are not pushed by default. If you require push, you need to set the offlinePushInfo desc field during sendMessage. The desc information will by default be displayed during push.

#### How do I disable the receiving of offline push messages?

To disable receiving offline push messages, you can set the config parameter of the setAPNS API to nil. This feature has been supported starting from version 5.6.1200.

#### Push notifications are not received, and the server reports a 'bad devicetoken'.

Apple's deviceToken depends on the current build environment. If the certificate ID and token used when logging into IMSDK and uploading deviceToken to Tencent Cloud do not match, an error will occur.

If you are using the Release environment compilation, the

application:didRegisterForRemoteNotificationsWithDeviceToken: callback returns the production environment's token. At this time, set the `businessID` to the production environment's Certificate ID. If you are using the Debug environment compilation, the \_\_\_\_\_\_

application:didRegisterForRemoteNotificationsWithDeviceToken: callback returns the

development environment's token. At this time, set the `businessID` to the development environment's Certificate ID.





```
V2TIMAPNSConfig *confg = [[V2TIMAPNSConfig alloc] init];
/* You need to register a developer certificate with Apple, download and generate t
// Push certificate ID
confg.businessID = sdkBusiId;
confg.token = self.deviceToken;
[[V2TIMManager sharedInstance] setAPNS:confg succ:^{
    NSLog(@"%s, succ, %@", __func__, supportTPNS ? @"TPNS": @"APNS");
} fail:^(int code, NSString *msg) {
    NSLog(@"%s, fail, %d, %@", __func__, code, msg);
}];
```

# Under the iOS development environment, is the registration occasionally not returning a deviceToken or indicating a failure in requesting the token from APNs?

This problem is caused by instability of APNs. You can fix it in the following ways:

- 1. Insert a SIM card into the phone and use the 4G network.
- 2. Uninstall and reinstall the application, restart the application, or shut down and restart the phone.
- 3. Use a package for the production environment.
- 4. Use another iPhone.

#### iOS did not receive the delivery receipt

1. To report push delivery data, enable the console switch to support iOS 10's Extension feature. For more details, please refer to Statistics Push Arrival Rate.

2. To ensure reported data is not lost in abnormal conditions, iOS push data reporting will only occur upon the next log in, which may result in a delay. Please try to log in again.

#### **iOS** Certificate Expiration

In the IM console, click editing on the corresponding certificate and update the p12 certificate. It's important to note: do not delete the certificate and recreate it, as recreating the certificate will change the certificate ID, and an update will be required to support it.

#### **iOS Statistical Reporting Feature**

To report push reach data, this switch must be enabled to support iOS 10's Extension feature. For more details, please refer to Statistics Push Arrival Rate.

### Others

#### Cannot Receive Push Notifications, Plugin Subscription Expired

Once the push plugin's **trial or purchase expires**, **the push service will automatically stop (including standard message offline push, broadcast/Tag push, etc.)**. To avoid affecting the normal use of your business, please purchase/renew in advance.

# Exchange and Feedback

Welcome to join the Tencent Cloud Chat Community for technical communication and feedback.