

# Chat 시작하기 제품 문서





#### 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** 

#### 🔗 Tencent Cloud

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

#### Service Statement

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

Chat



- 시작하기
  - 시작하기(Android)
  - 시작하기(iOS)
  - 시작하기(Web)
  - 시작하기(Electron)
  - 시작하기(Unity)
  - 시작하기(UE)
  - 시작하기(Flutter)
  - 시작하기 (React Native)
  - 시작하기(기존 애플리케이션에 Flutter 추가)

### Chat

# 시작하기 시작하기(Android)

최종 업데이트 날짜: : 2024-01-31 11:11:30

### 작업 단계

### 1단계: 애플리케이션 생성

1. IM 콘솔에 로그인합니다.

설명 :

이미 애플리케이션이 있는 경우 SDKAppID를 기록하고 키 정보 가져오기를 합니다.

동일한 Tencent Cloud 계정으로 최대 300개의 IM 애플리케이션을 만들 수 있습니다. 이미 300개의 애플리케이션이 있는 경우 신규 애플리케이션 생성 전에 사용하지 않는 애플리케이션을 비활성화 및 삭제합니다. **애플리케이션 삭제 후에는 SDKAppID에 해당하는 모든 데이터 및 서비스를 복구할 수 없으므로 주의하시기 바랍니다.** 

2. 애플리케이션 생성을 클릭하고 애플리케이션 생성 대화 상자에 애플리케이션 이름을 입력한 후 확인을 클릭합니다.

Application	Enter application name	
lag (f)	T Add	

3. 애플리케이션 생성 후, 콘솔 전체보기 페이지에서 새로 생성된 애플리케이션의 상태, 서비스 버전, SDKAppID, 생 성 시간, 태그 및 만료 시간을 확인할 수 있습니다. SDKAppID 정보를 저장하십시오.

Tencent C	Cloud Overvi	ew Products 👻	Security Situation	Awareness Vi	deo on Demand
Overview					
			In use		
	Plan	TRTC Trial (i)		Plan	TRTC Trial
	SDKAppID	) 🖻		SDKAppID	6
	Creation Time	2021-06-29		Creation Time	2021-06-29
	Expiration Time	-		Expiration Time	
	View Upgr	adeable Items		View Upg	gradeable Items

### 2단계: 키 정보 가져오기

1. 대상 애플리케이션 카드를 클릭하여 애플리케이션의 기본 구성 페이지로 이동합니다.

figuration   Current Region: Seoul (i)
Standard Billing Plan
Status In use
Plan Trial
Expiration - Time
Upgrade More 🔻
App Information
SDKAppID
Application Name
Application Game Type
Application - Introduction
Basic Information
Key Hide key
Key information is sensitive. Keep it confidential and do not disclose it.
Creation 2022-01-13 Time
Last Modified 2022-01-13

2. 기본 정보 섹션에서 키 표시를 클릭하고 키 정보를 복사 및 저장합니다.

### 주의:

키 정보가 유출되지 않도록 잘 보관하십시오.

### 3단계: Demo 소스 다운로드 및 설정

1. IM Demo 프로젝트를 다운로드합니다. 다운로드 주소는 SDK 다운로드를 참고하십시오. 설명: 이모티콘 디자인의 저작권 존중을 위해, 다운로드한 Demo 프로젝트에는 주요 이모티콘 요소의 슬라이스 이미지가 포함되어 있지 않습니다. 로컬 이모티콘 패키지를 사용하여 코드를 구성할 수 있습니다. IM Demo에서의 이모티콘 무 단 사용 행위는 디자인 저작권 침해에 해당될 수 있습니다.

2. 단말 디렉터리의 프로젝트를 열고 해당 GenerateTestUserSig 파일을 찾습니다. 경로:

Android/Demo/app/src/main/java/com/tencent/qcloud/tim/demo/signature/GenerateTestUserSig.java

3. GenerateTestUserSig 파일에서 관련 매개변수를 설정합니다.

SDKAPPID: 1단계에서 획득한 실제 애플리케이션 SDKAppID로 설정합니다.

SECRETKEY: 2단계에서 획득한 실제 키 정보로 설정합니다.



#### 주의:

본문의 UserSig 획득 방법은 클라이언트 코드에서 SECRETKEY를 설정하는 것입니다. 해당 방법 중 SECRETKEY는 디컴파일로 크래킹되기 쉬우므로, 키가 유출되면 해커가 사용자의 Tencent Cloud 트래픽을 도용할 수 있습니다. 따라 서 **해당 방법은 로컬 Demo 실행 및 기능 디버깅에만 적합합니다**.

올바른 UserSig 배포 방식은 UserSig 컴퓨팅 코드를 귀하의 서버에 통합하고, App 지향 인터페이스를 제공하는 것입

니다. UserSig가 필요할 때, App은 비즈니스 서버에 동적 UserSig 가져오기 요청을 발송합니다. 자세한 내용은 서버에서 UserSig 생성을 참고하십시오.

### 4단계: 데모 컴파일 및 실행

Android Studio를 사용하여 프로젝트를 가져오고 바로 컴파일 및 실행합니다. 자세한 내용은 3단계에서 복제된 Demo 프로젝트의 해당 디렉터리에 있는 README.md 파일을 참고하십시오. 개발 환경 요구 사항 Android Studio-Chipmunk Gradle-6.7.1 Android Gradle Plugin Version-4.2.0 kotlin-gradle-plugin-1.5.31 주의 :

Demo는 기본적으로 음성/영상 통화 기능과 통합되어 있습니다. 그러나 음성/영상 통화 기능이 의존하는 TRTC SDK 는 현재 시뮬레이터를 지원하지 않습니다. Demo 실행 또는 디버깅을 위해 실제 장치를 사용하십시오.

# 시작하기(iOS)

최종 업데이트 날짜: : 2024-01-31 11:12:31

본문은 Tencent Cloud IM Demo(iOS)를 빠르게 실행하는 방법을 소개합니다.

# 작업 단계

### 1단계: 애플리케이션 생성

1. IM 콘솔에 로그인합니다.

### 설명 :

이미 애플리케이션이 있는 경우 SDKAppID를 기록하고 키 정보 가져오기를 합니다.

동일한 Tencent Cloud 계정으로 최대 300개의 IM 애플리케이션을 만들 수 있습니다. 이미 300개의 애플리케이션이 있는 경우 신규 애플리케이션 생성 전에 사용하지 않는 애플리케이션을 비활성화 및 삭제합니다. **애플리케이션 삭제 후에는 SDKAppID에 해당하는 모든 데이터 및 서비스를 복구할 수 없으므로 주의하시기 바랍니다.** 

애플리케이션 생성을 클릭하고 애플리케이션 생성 대화 상자에 애플리케이션 이름을 입력한 후 확인을 클릭합니
 다.

Application	Enter application name	
Name *		
Tag 🚯	+ Add	

3. 애플리케이션 생성 후, 콘솔 전체보기 페이지에서 새로 생성된 애플리케이션의 상태, 서비스 버전, SDKAppID, 생 성 시간, 태그 및 만료 시간을 확인할 수 있습니다. SDKAppID 정보를 저장하십시오.

Tencent C	Cloud Overvi	ew Products 👻	Security Situation	Awareness Vi	deo on Demand
Overview					
			In use		
	Plan	TRTC Trial (i)		Plan	TRTC Trial
	SDKAppID	) 🖻		SDKAppID	6
	Creation Time	2021-06-29		Creation Time	2021-06-29
	Expiration Time	-		Expiration Time	
	View Upgr	adeable Items		View Upg	gradeable Items

### 2단계: 키 정보 가져오기

1. 대상 애플리케이션 카드를 클릭하여 애플리케이션의 기본 구성 페이지로 이동합니다.

figuration   Current Region: Seoul (i)
Standard Billing Plan
Status In use
Plan Trial
Expiration - Time
Upgrade More 🔻
App Information
SDKAppID
Application Name
Application Game Type
Application - Introduction
Basic Information
Key Hide key
Key information is sensitive. Keep it confidential and do not disclose it.
Creation 2022-01-13 Time
Last Modified 2022-01-13

2. 기본 정보 섹션에서 키 표시를 클릭하고 키 정보를 복사 및 저장합니다.

### 주의:

키 정보가 유출되지 않도록 잘 보관하십시오.

### 3단계: Demo 소스 다운로드 및 설정

1. IM Demo 프로젝트를 다운로드합니다. 다운로드 주소는 SDK 다운로드를 참고하십시오. 설명: Chat

이모티콘 디자인의 저작권 존중을 위해, 다운로드한 Demo 프로젝트에는 주요 이모티콘 요소의 슬라이스 이미지가 포함되어 있지 않습니다. 로컬 이모티콘 패키지를 사용하여 코드를 구성할 수 있습니다. IM Demo에서의 이모티콘 무 단 사용 행위는 디자인 저작권 침해에 해당될 수 있습니다. 2. 단말 디렉터리의 프로젝트를 열고, 해당 'GenerateTestUserSig' 파일을 찾습니다. iOS 경로: iOS/Demo/TUIKitDemo/Private/GenerateTestUserSig.h Mac 경로: Mac/Demo/TUIKitDemo/Debug/GenerateTestUserSig.h

3. GenerateTestUserSig 파일에서 관련 매개변수를 설정합니다.

SDKAPPID: 1단계에서 획득한 실제 애플리케이션 SDKAppID로 설정합니다.

SECRETKEY: 2단계에서 획득한 실제 키 정보로 설정합니다.

/**			
* Tencent Cloud SDKAppID. Set it to the SDKAppID of your account.			
*			
* You can view your SDKAppID after creating an application in the [Tencent Cloud IM			
console](https://console.intl.cloud.tencent.com/im).		App Informa	tion
* SDKAppID uniquely identifies a Tencent Cloud account.			
*/			
static const int public_SDKAPPID = 0;		SDKAppID	306
	-	Application	
/**	1	Vame	
* Signature validity period, which should not be set too short			
*	, A	Application	Game
* Time unit: second	1	ype	
* Default value: 604800 (7 days)	4	Application	
*/		ntroduction	
static const int EXPIRETIME = 604800;		noodoon	
Line in the second s			
* Follow the stars below to obtain the key required for UserSig calculation			
* Toriow the steps below to obtain the key required for Oseroig calculation.			
* Sten 1 Log in to the Fiencent Cloud IM console/https://console.intl.cloud.tencent.com/im), and create an			
application if you don't have one			
<ul> <li>Step 2. Click Application Configuration to go to the basic configuration page and locate Account System Integration.</li> </ul>			
* Step 3. Click View Key to view the encrypted key used for UserSig calculation. Then copy and paste the key to the variable below.			
*			
* Note: this method is for testing only. Before commercial launch, please migrate the UserSig calculation code and key to your			
backend server to prevent key disclosure and traffic stealing.		Basic Inf	ormation
* Reference: https://intl.cloud.tencent.com/document/product/1047/34385			
*/			
static NSString * const public SECRETKEY = @"";		Key	Disp
			Key inform
		Creation	2022-01-1
		Time	
		Last	2022-01-1
		Modified	

### 주의:

본문의 UserSig 획득 방법은 클라이언트 코드에서 SECRETKEY를 설정하는 것입니다. 해당 방법 중 SECRETKEY는 디컴파일로 크래킹되기 쉬우므로, 키가 유출되면 해커가 사용자의 Tencent Cloud 트래픽을 도용할 수 있습니다. 따라 서 **해당 방법은 로컬 Demo 실행 및 기능 디버깅에만 적합합니다**.

올바른 UserSig 배포 방식은 UserSig 컴퓨팅 코드를 귀하의 서버에 통합하고, App 지향 인터페이스를 제공하는 것입 니다. UserSig가 필요할 때, App은 비즈니스 서버에 동적 UserSig 가져오기 요청을 발송합니다. 자세한 내용은 서버 에서 UserSig 생성을 참고하십시오.

### 4단계: 컴파일 실행

3단계에서 복제된 Demo 프로젝트의 해당 디렉터리에 있는 README.md 파일을 참고하십시오. 1. 단말에서 다음 명령어를 실행하여 Pod 버전을 확인합니다.



pod --version

pod가 존재하지 않거나 pod 버전이 1.7.5 이전 버전이라는 메시지가 표시되면 다음 명령을 실행하여 최신 pod를 설치 합니다.





//gem 소스 교체 gem sources --remove https://rubygems.org/ gem sources --add https://gems.ruby-china.com/ //pod 설치 sudo gem install cocoapods -n /usr/local/bin //여러 Xcode가 설치된 경우 다음 명령을 사용하여 Xcode 버전을 선택하십시오(일반적으로 최신 Xcod sudo xcode-select -switch /Applications/Xcode.app/Contents/Developer //로컬 pod 라이브러리 업데이트 pod setup

2. 단말에서 다음 명령을 실행하여 종속 라이브러리를 설치합니다.





//iOS
cd iOS/TUIKitDemo
pod install
//Mac
cd Mac/TUIKitDemo
pod install

설치에 실패하면 다음 명령을 실행하여 로컬 CocoaPods 리포지토리 목록을 업데이트합니다.





pod repo update

3. 컴파일 및 실행

iOS: iOS/TUIKitDemo 폴더로 이동하여TUIKitDemo.xcworkspace를 열어 컴파일 및 실행합니다.Mac: Mac/TUIKitDemo 폴더로 이동하여TUIKitDemo.xcworkspace를 열어 컴파일 및 실행합니다.주의 :

Demo는 기본적으로 음성/영상 통화 기능과 통합되어 있습니다. 그러나 음성/영상 통화 기능이 의존하는 TRTC SDK 는 현재 시뮬레이터를 지원하지 않습니다. Demo 실행 또는 디버깅을 위해 실제 장치를 사용하십시오.

### Chat

# 고급 기능

UI 인터페이스 라이브러리 영상 통화 활성화

# 관련 문서

요금 안내

# 시작하기(Web)

최종 업데이트 날짜: : 2024-01-31 11:13:00

# **Chat UIKit React**

설명:

Chat UIKit은 Tencent Cloud IM SDK를 기반으로 하는 UI 컴포넌트 라이브러리로, 세션, 채팅, 관계 체인, 그룹, 음성/ 영상 통화 등의 기능을 포함한 일부 범용 UI 컴포넌트를 제공합니다.

UI 컴포넌트를 기반으로 블록을 쌓는 것처럼 자신의 비즈니스 로직을 빠르게 구축할 수 있습니다.

Chat UIKit의 컴포넌트는 UI 기능을 구현하는 동시에 IM SDK에 해당하는 인터페이스를 호출하여 IM 관련 로직과 데 이터를 처리하므로 개발자는 Chat UIKit을 사용할 때 자신의 업무에만 집중하거나 필요에 따라 확장할 수 있습니다.

### Example App

당사는 채팅 기능을 보여주는 작동하는 데모를 만들었습니다. 당사 웹사이트에서 이러한 demo와 GitHub의 관련 오 픈 소스 코드를 미리 볼 수 있습니다.

	Andrew			Developer Group
O. Se	Parch	æ		Rosa
	Developer Group Alice: [Image]	2:57 PM	Ŭ	Tencent Cloud Instant Messaging (IM) provides in-app chat A d communication. Based on the IM team's over 20 years of excellent backend performance, IM supports super large chat ro erves over 1 billion monthly active users.
	Rosa	1		Is there any
	You can try sending me text, e	2:57 PM		Linda
			47	https://www.tencentcloud.com/document/product/104
				Jack
				Does IM support on-screen commenting and liking in live stre
			Ø	Alice
				Charles Creat
			• +	- send a message

## 데모 실행

1단계: 소스 코드 다운로드





```
# Run the code in CLI
$ git clone https://github.com/TencentCloud/chat-uikit-react
# Go to the project
$ cd chat-uikit-react
# Install dependencies of the demo
$ npm install && cd examples/sample-chat && npm install
```

### 2단계: demo 설정

### S Tencent Cloud

1. examples/sample-chat 프로젝트를 열고 ./examples/sample-

chat/src/debug/GenerateTestUserSig.js 경로를 통해 GenerateTestUserSig.js 파일을 찾습니다. 2. GenerateTestUserSig.js 파일에 SDKAPPID 와 SECRETKEY 를 설정하고 IM 콘솔에서 해당 값을 얻을 수 있습니다. 대상 애플리케이션 카드를 클릭하여 설정 페이지로 이동합니다.

÷	Basic Configurat	ion	▼ Current Region: Seoul (i)
		Standard Bil	ling Plan
		Status	In use
		Plan	Trial
		Expiration Time	-
		Upgrade	More 💌
		App Informa	ation
		SDKAppID	ā
		Application Name	
		Application Type	Game
		Application Introduction	-
		Basic Inform	ation
		Key	Hide key
			Key information is sensitive. Keep it confidential and do not disclose it.
		Creation Time	2022-01-13
		Last Modified	2022-01-13

3. Basic Information 영역에서 Display key를 클릭하여 키 정보를 복사하고 GenerateTestUserSig 파일에 저장합니 다.

-1.

주의:

본 문서의 UserSig 생성 방법은 클라이언트 코드에서 SECRETKEY를 설정하는 것입니다. 이 방법에서 SECRETKEY 는 디컴파일로 크래킹되기 쉬우므로, 키가 유출되면 해커가 귀하의 Tencent Cloud 트래픽을 도용할 수 있습니다. 따 라서 **해당 방법은 로컬 Demo 실행 및 기능 디버깅용으로만 적합합니다.** 

UserSig의 계산 코드를 서버에 통합하고 App 지향 API를 제공하는 것이 가장 좋습니다. UserSig가 필요할 때 App은 동적 UserSig에 대한 요청을 서버에 보낼 수 있습니다. 자세한 내용은 서버에서 UserSig 생성을 참고하십시오.

### 3단계: 프로젝트 실행



- # Launch the project
- \$ cd examples/sample-chat

\$ npm run start

### 4단계: 첫 번째 메시지 전송

1. 프로젝트가 성공적으로 실행된 후 '+' 아이콘을 클릭하여 세션을 생성합니다.

2. 입력란에 다른 사용자의 userID를 검색합니다.

3. 사용자 프로필 사진을 클릭하여 세션을 시작합니다.

4. 입력란에 메시지를 입력하고 "Enter"키를 누르면 전송됩니다.



#### **Quick Links**

Demo App Client API Free Demos FAQ



GitHub Source Generating UserSig

# 시작하기(Electron)

최종 업데이트 날짜: : 2024-01-31 11:14:29

본문은 IM Demo(Electron)를 빠르게 실행하는 방법과 Electron SDK를 통합하는 방법을 설명합니다.

### 환경 요건

플랫폼	버전
Electron	13.1.5 이상 버전.
Node.js	v14.2.0

### 플랫폼 지원

현재 Macos 및 Windows 2가지 플랫폼을 지원합니다.

### DEMO 체험

통합하기 전에 DEMO를 사용하여 IM Electron SDK의 기능을 빠르게 이해할 수 있습니다.

### 전제 조건

Signing Up된 계정이 있고 Identity Verification이 완료되어야 합니다.

# 작업 단계

### 1단계: 애플리케이션 생성

1. IM 콘솔에 로그인합니다.

### 설명:

이미 애플리케이션이 있는 경우 SDKAppID를 기록하고 키 정보 가져오기를 합니다. 동일한 Tencent Cloud 계정으로 최대 300개의 IM 애플리케이션을 만들 수 있습니다. 이미 300개의 애플리케이션이 있는 경우 신규 애플리케이션 생성 전에 사용하지 않는 애플리케이션을 비활성화 및 삭제합니다. **애플리케이션 삭제** 후에는 SDKAppID에 해당하는 모든 데이터 및 서비스를 복구할 수 없으므로 주의하시기 바랍니다.

2. 애플리케이션 생성을 클릭하고 애플리케이션 생성 대화 상자에 애플리케이션 이름을 입력한 후 확인을 클릭합니다.

Application Name <b>*</b>	Enter application name	
Tag 🚯	+ Add	

3. SDKAppID 정보를 저장하십시오. 콘솔 전체보기 페이지에서 새로 생성된 애플리케이션의 상태, 서비스 버전, SDKAppID, 태그, 생성 시간 및 만료 시간을 확인할 수 있습니다.

Tencent Clo	oud Overvi	ew Products <del>-</del>	Security Situation	Awareness N	Video on Demand
Overview					
			In use		
	Plan	TRTC Trial (i)		Plan	TRTC Trial
	SDKAppID	) 🗅		SDKAppID	6
	Creation Time	2021-06-29		Creation Time	2021-06-29
	Expiration Time	-		Expiration Time	-
	View Upgr	adeable Items		View U	pgradeable Items

4. 생성된 애플리케이션을 클릭한 후, 왼쪽 사이드바의 **보조 툴 > UserSig 툴**을 클릭하여, UserID 및 UserSig를 생성 하고, 로그인에 사용할 UserSig를 복사합니다.

Instant Messaging	- UserSig Generation & Verification
王는 Basic Configuration	Signature (UserSig) Generator
□ Feature ~ Configuration	This tool can quickly generate a UserSig, which can be used to run through demos
晶 Group Management	Username (UserID)
Callback Configuration	Key
🗟 Data Monitor 🛛 👻	
🖬 Auxiliary Tools 🛛 🔿	
<ul> <li>Push Message Tool</li> </ul>	
UserSig Tools	Generate UserSig
	Current Signature (UserSig)
	Copy UserSig
E	

### 2단계: Electron SDK를 통합하기 위한 적절한 방법 선택

Tencent Cloud IM은 두 가지 통합 체계를 제공합니다. 가장 적합한 체계를 선택할 수 있습니다.

통합 방법	적용 가능한 시나리오
DEMO	IM Demo에는 모든 채팅 기능이 포함되어 있으며 오픈 소스 코드를 제공합니다. 채팅 시나리오를 구
사용	현하려면 Demo를 사용하여 2차 개발을 진행할 수 있습니다. Demo에서 체험해 보십시오.

자체	Demo가 UI 요구 사항을 충족하지 않는 경우 이 방법을 사용할 수 있습니다.
구현	

IM SDK의 API 이해를 돕기 위해, API 문서를 제공합니다.

### 3단계: Demo 사용

1. IM Electron Demo 소스 코드를 로컬에 클론합니다.



git clone https://github.com/TencentCloud/tc-chat-demo-electron.git

2. 프로젝트 종속성을 설치합니다.



// 프로젝트의 루트 디렉터리 npm install

// 렌더링 프로세스 디렉터리 cd src/client npm install

### 3. 프로젝트를 실행합니다.



// 프로젝트의 루트 디렉터리 npm start

4. 프로젝트를 패키징합니다.





// mac 패키징 npm run build:mac // windows 패키징 npm run build:windows

### 설명:

demo에서 메인 프로세스 디렉터리 src/app/main.js 와 렌더링 프로세스 디렉터리는 src/client 입니다. 실행 중 문제가 발생하면 먼저 문제 해결을 위한 FAQ를 참고하십시오.

### 4단계: 자체 구현

Electron SDK 설치

다음 명령을 사용하여 최신 버전의 Electron SDK를 설치합니다 다음 명령을 실행합니다.



npm install im\_electron\_sdk

### SDK 초기화

1. TimMain 에 sdkAppID 를 전달합니다.





```
// 메인 프로세스
const TimMain = require('im_electron_sdk/dist/main')
const sdkappid = 0;// IM 콘솔에서 신청할 수 있습니다
const tim = new TimMain({
sdkappid:sdkappid
})
```

2. TIMInit 를 호출하여 SDK를 초기화합니다.





//렌더링 프로세스 const TimRender = require('im\_electron\_sdk/dist/render') const timRender = new TimRender(); // 컴포넌트 초기화 timRender.TIMInit()

3. 테스트 사용자로 로그인합니다.

로그인 확인을 위해 콘솔에서 처음 생성한 테스트 계정으로 로그인합니다. timRender.TIMLogin 메소드를 호출하여 테스트 사용자로 로그인합니다. 반환된 code 가 0이면 로그인에 성공한 것입니다.





```
const TimRender = require('im_electron_sdk/dist/render')
const timRender = new TimRender();
let {code} = await timRender.TIMLogin({
  userID:"userID",
  userSig:"userSig" // userSig 생성 참고
})
```

### 설명:

이 계정은 개발 테스트 전용이며, 애플리케이션이 런칭되기 전에 정확한 UserSig 배포 방법은 UserSig 계산 코드를 서버에 통합하고 APP 지향 API를 제공하는 것입니다. UserSig 가 필요한 경우 APP은 동적 UserSig 에
대한 요청을 비즈니스 서버에 보낼 수 있습니다. 자세한 내용은 Generating UserSig를 참고하십시오.

#### 메시지 보내기

다음 샘플은 문자 메시지를 보내는 방법을 보여줍니다. 반환된 code 가 0이면 메시지가 성공적으로 전송된 것입니다.

코드 예시:



```
conv_type: 1,
params: {
    message_elem_array: [{
        elem_type: 1,
        text_elem_content:'Hello Tencent!',
      }],
      message_sender: "senderID",
    },
    callback: (data) => {}
  }
let {code} = await timRender.TIMMsgSendMessageV2(param);
```

#### 설명 :

메시지 전송에 실패한 경우, sdkAppID가 낯선 사람에게 메시지 전송을 지원하지 않기 때문일 수 있습니다. 이 경우 테스트를 위해 콘솔에서 친구 관계 체인 확인 비활성화할 수 있습니다.

#### 대화 목록 가져오기

이전 단계에서 테스트 메시지를 보내면 다른 테스트 계정에 로그인하여 대화 목록을 가져올 수 있습니다.

일반 응용 시나리오는 다음과 같습니다.

응용 프로그램 시작 시 대화 목록을 가져오고 지속적인 연결을 수신하여 대화 목록을 실시간으로 업데이트합니다.



```
let param:getConvList = {
        userData:userData,
    }
let data:commonResult<convInfo[]> = await timRenderInstance.TIMConvGetConvList(para
```

이 때 이전 단계에서 다른 테스트 계정에서 보낸 메시지를 볼 수 있습니다.

#### 메시지 수신

일반 응용 시나리오는 다음과 같습니다.

1. 인터페이스는 새로운 대화가 시작되면 먼저 일정량의 메시지 기록을 한꺼번에 요청하여 메시지 기록 목록을 보여 줍니다.

2. 긴 링크를 수신하여 실시간으로 새로운 메시지를 수신하여 메시지 기록 목록에 추가합니다.
 일회성 요청 메시지 기록 리스트



```
let param:MsgGetMsgListParams = {
    conv_id: conv_id,
    conv_type: conv_type,
    params: {
        msg_getmsglist_param_last_msg: msg,
        msg_getmsglist_param_count: 20,
        msg_getmsglist_param_is_remble: true,
    },
```

```
user_data: user_data
}
let msgList:commonResult<Json_value_msg[]> = await timRenderInstance.TIMMsgGetM
```

실시간으로 새 메시지 수신 다음은 callback 바인딩에 대한 샘플 코드입니다.



```
let param : TIMRecvNewMsgCallbackParams = {
    callback: (...args)=>{},
    user_data: user_data
}
```

timRenderInstance.TIMAddRecvNewMsgCallback(param);

이제 IM 모듈 개발을 완료했으며, 메시지를 주고받거나 다른 대화에 들어갈 수 있습니다. 그룹, 사용자 프로필, 관계 체인, 오프라인 푸시 및 로컬 검색과 같은 더 많은 기능을 개발할 수 있습니다. 자세한 내용은 API 문서를 참고하십시오.

### FAQ

#### 어떤 플랫폼이 지원됩니까?

현재 Macos 및 Windows 2가지 플랫폼을 지원합니다.

#### 에러 코드를 쿼리하는 방법은 무엇입니까?

IM SDK의 API 레이어 에러 코드는 에러 코드를 참고하십시오.

개발 환경 설치 중에 npm ERR! gyp ERR! stack TypeError: Cannot assign to read only property 'cflags' of object '#<Object>' 오류가 보고되면 어떻게 해야 합니까?

node 버전을 16.18.1로 다운그레이드합니다.

#### 개발 환경 설치 문제 gypgyp ERR!ERR 오류를 해결하는 방법은 무엇입니까?

gypgyp ERR!ERR! 를 참고하십시오.

npm install 실행시 npm ERR! Fix the upstream dependency conflict, or retry 오류가보 고되면 어떻게 해야 하나요?

npmV7 이전 버전에서는 설치 중에 발생하는 종속성 충돌이 자동으로 무시됩니다. 계속 설치를 진행합니다. npmV7 이상 버전에서는 종속성 충돌이 자동으로 무시되지 않으며, 이를 무시하려면 수동으로 명령을 입력해야 합니 다.

종속성 충돌을 무시하는 명령은 다음과 같습니다.





npm install --force

npm run start 실행시 Error: error:0308010C:digital envelope routines::unsupported 오류가 보고되면 어떻게 해야 하나요?

node 버전을 16.18.1로 다운그레이드합니다.

Mac 클라이언트 Demo에서 npm run start 를 실행할 때 화면이 흰색으로 바뀌면 어떻게 해야 하나요?

Mac에서 npm run start 를 실행하면 흰색 화면이 나오는 이유는 아직 렌더링 프로세스의 코드가 build 완료되 지 않았고, 메인 프로세스가 오픈한 포트 3000번이 빈 페이지이기 때문입니다. 렌더링 프로세스의 코드 build 완료 후 창을 새로고침하면 문제가 해결될 수 있습니다. 또는 cd src/client && npm run dev:react , npm run dev:electron 을 실행하여 렌더링 프로세스와 메인 프로세스를 각각 시작합니다.

#### vue-cli-plugin-electron-builder 로 빌드된 프로젝트에서 native modules 을 어떻게 사용합니까?

vue-cli-plugin-electron-builder 로 빌드된 프로젝트에서 native modules 사용과 관련된 문제는 No native build was found for platform = xxx를 참고하십시오.

#### webpack 으로 빌드한 프로젝트에서 native modules 을 어떻게 사용하나요?

webpack으로 빌드한 프로젝트에서 native modules를 사용하는 것과 관련된 문제는 Windows 환경 FAQ를 참고하십 시오.

#### Dynamic Linking Error 오류가 보고되면 어떻게 해야 합니까?

Dynamic Linking Error. electron-builder 구성





```
extraFiles:[
{
    "from": "./node_modules/im_electron_sdk/lib/",
    "to": "./Resources",
    "filter": [
        "**/*"
    ]
  }
]
```

# 시작하기(Unity)

최종 업데이트 날짜: : 2024-01-31 11:17:09

본문은 Unity SDK를 프로젝트에 통합하는 전체 프로세스를 소개합니다.

## 환경 요건

플랫폼	버전
Unity	2019.4.15f1 이후 버전.
Android	Android Studio 3.5 이후 버전. App은 Android 4.1 이후 버전 디바이스가 필요합니다.
iOS	Xcode 11.0 이후 버전. 프로젝트에 유효한 개발자 서명이 설정되어 있는지 확인십시오.

### 플랫폼 지원

당사는 Unity에서 전체 플랫폼을 커버하는 Tencent Cloud IM SDK 개발에 주력하여 하나의 코드 세트로 모든 플랫폼 에서 실행할 수 있도록 지원합니다.

플랫폼	IM SDK
iOS	지원
Android	지원
macOS	지원
Windows	지원
Web	1.8.1+ 버전부터 지원

설명:

Web 빌드에는 간단한 추가 단계가 필요합니다. 파트5를 확인하십시오.

## 전제 조건

- 1. Signing Up된 계정이 있고 Identity Verification이 완료되어야 합니다.
- 2. 애플리케이션 생성 및 업그레이드를 참고하여 애플리케이션을 생성하고 'SDKAppID'를 기록합니다.

## 파트1: 테스트 사용자 생성

IM 콘솔에서 애플리케이션을 선택하고 왼쪽 사이드바에서 **보조 툴 > UserSig Generation & Verification**을 클릭하 여 두 개의 UserID와 해당 UserSig를 생성하고 UserID, 서명(Key), UserSig 를 복사하여 이후 로그인에 사 용합니다.

설명 :

이 계정은 개발 및 테스트 전용입니다. 애플리케이션 런칭 전에 올바른 UserSig 배포 방식은 서버에서 생 성하여, App 지향 인터페이스를 제공하는 것입니다. UserSig 가 필요할 때, App은 비즈니스 서버에 동적 UserSig 가져오기 요청을 발송합니다. 자세한 내용은 Generating UserSig를 참고하십시오.

於 購讯云 总览	云产品 🗸 📄 网站备案 🛛 🕂
即时通信 IM	← UserSig生成&校验 -
書 基本配置	签名(UserSig)牛成丁具 登录鉴权介绍 🖸
品 功能配置 各 群组管理	
	此工具可以快速生成签名(UserSig),用于本地跑通 Demo 以及功能调试。 请输入用户名(UserID)
🕣 安全打击	user1
<table-cell> 数据监控器</table-cell>	密钥 (Key)
@ 辅助工具	
• 离线推送自查	
● UserSig生成&校 验	生成签名 (UserSig)
	当前生成的签名(UserSig)是:
	复制签名 (UserSig)

## 파트2: Tencent Cloud IM SDK를 Unity 프로젝트에 통합

1. Unity를 사용하여 프로젝트를 생성하고 프로젝트 디렉터리를 기록합니다.

•••		使用 Unity 2020.3.23f	1c1创建新项目	
模板			设置	
	©		项目名称 * im_unity_test	
2D	3D	High Definition RP	位置 * /Users/	
	★ 下载	<u>.</u> 	□ 启用 PlasticSCM 并同意 政策条款	
Universal Render Pipeline	2D Platformer Microgame	Karting Microgame		
*	*	± (2)		
			取消	创建

또는 기존 Unity 프로젝트를 엽니다.

2. IDE(예시: Visual Studio Code)를 통해 프로젝트를 엽니다.

ப	资源管理器 ···	() manifest.json M ×
	> 打开的编辑器	Packages > () manifest.json > () dependencies
<b>.</b>	$\sim$ IM-UNITY-UIKIT	20 com.unity.modules.drector: 1.00,
_	✓ Assets	21 Com.unity.modules.imageconversion: "1.0.0",
$\mathcal{Q}$		22 "com.unity.modules.imgui": "1.0.0",
0 -	> Scenes	23 ····································
<b>J</b> 0 2	> Library	24 com.unity.modules.particlesystem": "1.0.0",
	> Logs	25 com.unity.modules.physics": "1.0.0",
¢⁄	✓ Packages	26 "com.unity.modules.physics2d": "1.0.0",
	{} manifest.json M	27   ·   · "com.unity.modules.screencapture": "1.0.0",
-0	> ProjectSettings	28 ····································
нD	> Temp	29 🖂 Com.unity.modules.terrainphysics": "1.0.0",
	> UserSettings	30 com.unity.modules.tilemap": "1.0.0",
	• .gitignore	31 "com.unity.modules.ui": "1.0.0",
	(i) README.md	32 com.unity.modules.uielements": "1.0.0",
$\bigcirc$		33 ·····"com.unity.modules.umbra": "1.0.0",
		34 com.unity.modules.unityanalytics": "1.0.0",
		35 ·····"com.unity.modules.unitywebrequest": "1.0.0",
$\bigcirc$		36 ····································
<u>ک</u> ر		37 ····································
		38 com.unity.modules.unitywebrequesttexture": "1.0.0",
		<pre>39 "com.unity.modules.unitywebrequestwww": "1.0.0",</pre>
		40 "com.unity.modules.vehicles": "1.0.0",
		41 "com.unity.modules.video": "1.0.0",
		42 "com.unity.modules.vr": "1.0.0",
		43 "com.unity.modules.wind": "1.0.0",
		44 "com.unity.modules.xr": "1.0.0",
_		45
		46
0		47 }

3. 디렉터리에 따라 Packages/manifest.json을 찾아 종속성을 다음과 같이 수정합니다.

```
{
   "dependencies":{
   "com.tencent.imsdk.unity":"https://github.com/TencentCloud/chat-sdk-unity.git#u
nity"
}
```

Tencent Cloud IM SDK 내의 모든 API를 더 잘 이해할 수 있도록 개발 초기 단계에서 SDK API를 테스트하고 특정 API 를 호출하기 위한 API Example을 제공합니다.

## 파트3: 로딩 종속성

Unity Editor에서 프로젝트를 열고 종속성 로딩이 완료되기를 기다렸다가 Tencent Cloud IM이 로딩되었는지 확인합 니다.

Chat





### 파트4: 자체 UI와 통합

#### 전제 조건

Unity 프로젝트를 생성하고 Tencent Cloud IM SDK를 로딩했습니다.

#### SDK 초기화

#### 본 섹션의 상세 문서

TencentIMSDK.Init 를 호출하여 SDK를 초기화합니다.

SDKAppID 를 전달합니다.

```
public static void Init() {
int SDKAppID = 0; // IM 콘솔에서 SDKAppID를 가져옵니다.
SdkConfig sdkConfig = new SdkConfig();
```

```
sdkConfig.sdk_config_config_file_path = Application.persistentDataPath + "/TIM-Co
nfig";
```

sdkConfig.sdk\_config\_log\_file\_path = Application.persistentDataPath + "/TIM-Log";



```
TIMResult res = TencentIMSDK.Init(long.Parse(SDKAppID), sdkConfig);
}
```

Init 후에는 주로 네트워크 상태 및 사용자 정보 변경에 대한 리스너를 포함하여 일부 리스너를 IM SDK에 마운트 할 수 있습니다. 자세한 내용은 여기에서 확인하십시오.

#### 테스트 계정으로 로그인

#### 본 섹션의 상세 문서

콘솔에서 처음 생성된 테스트 계정을 사용하여 로그인 인증을 완료할 수 있습니다.

TencentIMSDK.Login 을 호출하여 계정에 로그인하십시오.

반환값 'res.code'가 0이면 로그인이 성공한 것입니다.

```
public static void Login() {
  if (userid == "" || user_sig == "")
  {
    return;
  }
  TIMResult res = TencentIMSDK.Login(userid, user_sig, (int code, string desc, stri
  ng json_param, string user_data)=>{
    // 로그인 콜백 로직 프로세스
  });
  }
```

설명:

이 계정은 개발 테스트 전용입니다. 애플리케이션이 런칭되기 전에 정확한 UserSig 발급 방식은 다음과 같습니다. UserSig 계산 코드를 귀하의 서버에 통합하고 App 방향의 인터페이스를 제공합니다. UserSig 가 필요할 때 귀하의 App이 비즈니스 서버로 동적 UserSig 를 요청합니다. 자세한 내용은 Generating UserSig를 참고하십시오.

#### 메시지 발송

본 섹션의 상세 문서

다음은 문자 메시지를 보내는 예시 코드입니다

코드 예시:

```
public static void MsgSendMessage() {
string conv id = ""; // c2c 메시지의 대화 ID는 userID, 그룹 메시지의 대화 ID는 groupID
```

### S Tencent Cloud

```
Message message = new Message
{
message_conv_id = conv_id,
message_conv_type = TIMConvType.kTIMConv_C2C, // 그룹 메시지는 TIMConvType.kTIMConv
message_elem_array = new List<Elem>
{
new Elem
{
elem type = TIMElemType.kTIMElem Text,
text elem content = "일반 문자 메시지입니다"
}
}
};
StringBuilder messageId = new StringBuilder(128);
TIMResult res = TencentIMSDK.MsgSendMessage(conv_id, TIMConvType.kTIMConv_C2C, me
ssage, messageId, (int code, string desc, string json_param, string user_data)=>{
// 비동기 메시지 전송 결과
});
// 메시지가 전송될 때 반환되는 메시지 ID messageId
}
```

설명:

만약 SDKAppID가 낯선 사람에게 메시지를 보낼 수 없기 때문에 전송에 실패한다면, 테스트용으로 콘솔에서 활성화할 수 있습니다.

이 링크를 클릭하여, 친구 관계망 확인을 비활성화하십시오.

#### 대화 목록 가져오기

#### 본 섹션의 상세 문서

이전 단계에서 테스트 메시지를 보내면 다른 테스트 계정에 로그인하여 대화 목록을 가져올 수 있습니다.

대화 목록을 가져오는 방법은 두 가지입니다.

장기간 연결 콜백을 수신하여 대화 목록을 실시간으로 업데이트합니다.
 API를 요청하여 페이징에 따라 대화 목록을 한번에 가져옵니다.

일반 응용 시나리오는 다음과 같습니다.

응용 프로그램을 실행한 후 바로 대화 목록을 가져온 다음 장기간 연결을 수신하여 대화 목록의 변경 사항을 실시간 으로 업데이트합니다.



#### 일회성 요청 대화 목록

```
TIMResult res = TencentIMSDK.ConvGetConvList((int code, string desc, List<ConvInf
o> info_list, string user_data)=>{
// 비동기 로직 프로세스
});
```

이 때 이전 단계에서 다른 테스트 계정에서 보낸 메시지를 볼 수 있습니다.

#### 긴 링크에서 실시간으로 대화 목록 가져오기

이 단계에서는 SDK에 리스너를 마운트한 다음 콜백 이벤트를 처리하고 UI를 업데이트해야 합니다.

1. 리스너 마운트.

```
TencentIMSDK.SetConvEventCallback((TIMConvEvent conv_event, List<ConvInfo> conv
_list, string user_data)=>{
// 콜백 로직 프로세스
});
```

2. 콜백 이벤트를 처리하고 UI에 최신 대화 목록을 표시합니다.

#### 메시지 수신

#### 본 섹션의 상세 문서

Tencent Cloud IM SDK를 통해 메시지를 수신하는 두 가지 방법:

- 1. 장기간 연결 콜백을 수신하고, 실시간으로 메시지 변경 사항을 가져오고, 렌더링 메시지 기록 목록을 업데이트합 니다.
- 2. API를 요청하여 페이징에 따라 메시지 기록를 한번에 가져옵니다.

일반 응용 시나리오는 다음과 같습니다.

1. 인터페이스는 새로운 대화가 시작되면 먼저 일정량의 메시지 기록을 한꺼번에 요청하여 메시지 기록 목록을 보여 줍니다.

2. 긴 링크를 수신하여 실시간으로 새로운 메시지를 수신하여 메시지 기록 목록에 추가합니다.

#### 일회성 요청 메시지 기록 리스트

페이지당 풀링하는 메시지의 수는 너무 크면 안 됩니다. 그렇지 않으면 풀링 속도에 영향을 줄 수 있습니다. 약 20으로 설정하는 것이 좋습니다.

다음 요청 시 현재 페이지 수를 동적으로 기록해야 합니다.

예시 코드는 다음과 같습니다.

```
// 일대일 메시지 기록 풀링
// 첫 번째 풀링에 대해 msg getmsglist param last msg를 null로 설정
// msg_getmsglist_param_last_msg는 두 번째 풀링에 대해 반환된 메시지 목록의 마지막 메시지
일 수 있습니다
var get_message_list_param = new MsgGetMsgListParam
{
msg_getmsglist_param_last_msg = LastMessage
};
TIMResult res = TencentIMSDK.MsgGetMsgList(conv_id, TIMConvType.kTIMConv_C2C, get
_message_list_param, (int code, string desc, string user_data) => {
// 콜백 로직 프로세스
});
// 그룹 메시지 기록 풀링
// 첫 번째 풀링에 대해 msg_getmsglist_param_last_msg를 null로 설정
// msg_getmsglist_param_last_msg는 두 번째 풀링에 대해 반환된 메시지 목록의 마지막 메시지
일 수 있습니다
var get_message_list_param = new MsgGetMsgListParam
{
msg_getmsglist_param_last_msg = LastMessage
};
TIMResult res = TencentIMSDK.MsgGetMsgList(conv_id, TIMConvType.kTIMConv_Group, q
et_message_list_param, (int code, string desc, string user_data) => {
// 콜백 로직 프로세스
});
```

#### 긴 링크를 통해 실시간으로 새 메시지 가져오기

메시지 기록 목록이 초기화된 후 새 메시지는 긴 링크 Tencent IMSDK.AddRecvNewMsgCallback 에서 나옵니 다.

AddRecvNewMsgCallback 콜백이 트리거된 후 필요에 따라 메시지 기록 목록에 새 메시지를 추가할 수 있습니다.

바인딩 리스너의 예시 코드는 다음과 같습니다.

```
TencentIMSDK.AddRecvNewMsgCallback((List<Message> message, string user_data) => { // 새 메시지 프로세스 });
```

이제 IM 모듈 개발을 완료했으며, 메시지를 주고받거나 다른 대화에 들어갈 수 있습니다.

귀하는 계속해서 그룹, 사용자 프로필, 친구 관리, 로컬 검색 등 관련 기능을 개발할 수 있습니다.

자세한 내용은 통합 솔루션(UI 없음)을 참고하십시오.

## 파트5: #Unity for WebGL

Tencent Cloud IM SDK (Unity 버전)는 1.8.1 버전부터 WebGL용 빌드를 지원합니다.

Web 지원을 활성화하려면 Android 및 iOS 지원을 활성화하는 단계와 비교하여 다음과 같은 추가 단계를 수행해야 합니다.

### JS 가져오기

GitHub에서 다음 3개의 JS 파일을 다운로드하여 프로젝트 빌드 WebGL 제품 폴더에 배치합니다.

- tim-js
- tim-js-friendship.js
- 이 파일의 이름을 tim-upload-plugin.js로 변경

index.html 을 열고 이 세 개의 JS 파일을 가져옵니다. 다음과 같습니다.

```
<script src="./tim-js.js"></script>
<script src="./tim-js-friendship.js"></script>
<script src="./tim-upload-plugin.js"></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></sc
```

### FAQ

#### 어떤 플랫폼이 지원됩니까?

현재 iOS, Android, Windows, Mac 및 WebGL이 지원됩니다.

# Android에서 Build And Run 클릭 후, 사용 가능한 디바이스를 찾을 수 없다는 오류가 보고되면 어떻게 해야 합니까?

디바이스가 다른 리소스에 의해 점유되어 있지 않은지 확인하거나, Build를 클릭하여 apk 패키지를 생성한 다음 시뮬 레이터로 드래그하여 실행합니다.

#### iOS 최초 실행 시 오류가 보고되면 어떻게 합니까?

상기 내용에 따라 설정된 Demo 실행 시 오류가 보고되면, **Product>Clean**을 클릭하여 지우고 다시 Build하거나, Xcode를 닫고 재실행 후 다시 Build합니다.

#### Unity 2019.04 버전이 iOS 플랫폼에서 오류가 보고되면 어떻게 합니까?

Library/PackageCache/com.unity.collab-proxy@1.3.9/Editor/UserInterface/Bootstrap.cs(23,20): error CS0117:

'Collab' does not contain a definition for 'ShowChangesWindow'

Editor 툴바에서 Window>Package Manager를 클릭하여 Unity Collaborate를 1.2.16으로 다운그레이드합니다.

### Unity 2019.04 버전이 iOS 플랫폼에서 오류가 보고되면 어떻게 합니까?

Library/PackageCache/com.unity.textmeshpro@3.0.1/Scripts/Editor/TMP\_PackageUtilities.cs(453,84): error CS0103: The name 'VersionControlSettings' does not exist in the current context 소스 코드를 열고 || VersionControlSettings.mode != "Visible Meta Files" 코드를 삭제합니다.

### 에러 코드를 쿼리하는 방법은 무엇입니까?

• IM SDK의 API 레이어 에러 코드는 에러 코드를 참고하십시오.

# 시작하기(UE)

최종 업데이트 날짜: : 2024-01-31 11:18:04

본문은 Tencent Cloud IM(Instant Messaging) Demo(Unreal Engine)를 빠르게 실행하는 방법을 소개합니다.

#### 설명 :

현재 Windows, macOS, iOS, Android를 지원합니다.

## 환경 요건

Unreal Engine 4.27.1 이상 버전 권장.

개발측	환경
Android	Android Studio 4.0 이상 버전. Visual Studio 2017 15.6 이상 버전. 기기 디버깅만 지원됩니다.
iOS & macOS	Xcode 11.0 이상 버전. OSX 시스템 버전은 10.11 이상 버전이 필요합니다. 프로젝트에 유효한 개발자 서명이 설정되어 있는지 확인하십시오.
Windows	운영 체제: Windows 7 SP1 이상 버전(x86-64 기반 64비트 운영 체제). 디스크 공간: IDE 및 일부 툴 설치 외에 최소 1.64GB의 공간이 있어야 합니다. Visual Studio 2019를 설치합니다.

## 전제 조건

Tencent Cloud 가입된 계정이 있고 실명 인증이 완료되어야 합니다.

## 작업 단계

### 1단계: 신규 애플리케이션 생성

1. Ⅲ 콘솔에 로그인합니다.

설명 :

이미 애플리케이션이 있는 경우 SDKAppID를 기록하고 키 정보 가져오기를 합니다.

동일한 Tencent Cloud 계정으로 최대 300개의 IM 애플리케이션을 만들 수 있습니다. 이미 300개의 애플리케이션이

있는 경우 신규 애플리케이션 생성 전에 사용하지 않는 애플리케이션을 비활성화 및 삭제합니다. **애플리케이션 삭제** 후에는 SDKAppID에 해당하는 모든 데이터 및 서비스를 복구할 수 없으므로 주의하시기 바랍니다.

애플리케이션 생성을 클릭하고 애플리케이션 생성 대화 상자에 애플리케이션 이름을 입력하고 확인을 클릭합니
 다.

Create Appl	cation	~
Create Appl	Cation	~
Application Name *	Enter application name	
Tag 🚯	+ Add	
	Confirm	
Tag	+ Add Confirm	

3. 애플리케이션 생성 후, 콘솔 전체보기 페이지에서 새로 생성된 애플리케이션의 상태, 서비스 버전, SDKAppID, 생 성 시간, 태그 및 만료 시간을 확인할 수 있습니다. SDKAppID 정보를 저장하십시오.

Tencent Cloud	Overview	Products -	Security Situation	Awareness N	video on Demand
Overview					
C1	1 - C		In use		
Pla	in	TRTC Trial (i)		Plan	TRTC Trial
SD	KAppID	) 靣		SDKAppID	6
Cre Tin	eation 2 ne	2021-06-29		Creation Time	2021-06-29
Exp Tin	piration ·	-		Expiration Time	-
	View Upgrade	able Items		View U	pgradeable Items

### 2단계: 키 정보 가져오기

1. 대상 애플리케이션 카드를 클릭하여 애플리케이션의 기본 구성 페이지로 이동합니다.

Instant Messaging	<ul> <li>UserSig Generation &amp; Verification</li> </ul>
∃¦ Basic Configuration	Signature (UserSig) Generator
Feature * Configuration	This tool can quickly generate a UserSig, which can be used to run through demos
晶 Group Management	Username (UserID)
Callback Configuration	Key
🕞 Data Monitor 🛛 👻	
📩 Auxiliary Tools 🛛 🔨	
<ul> <li>Push Message Tool</li> </ul>	
UserSig Tools	Generate UserSig
	Current Signature (UserSig)
<b>=</b>	Copy UserSig

2. 기본 정보 섹션에서 키 표시를 클릭하고 키 정보를 복사 및 저장합니다.

#### 주의:

키 정보가 유출되지 않도록 잘 보관하십시오.

### 3단계: Demo 프로그래밍 파일 설정

1. IM Demo 프로젝트 다운로드 특정 다운로드 주소는 Demo 다운로드를 참고하십시오(문의 사항은 QQ 그룹 번호: 764231117 추가 후 상담).

2. /IM\_Demo/Source/debug/include/DebugDefs.h 파일을 찾아 엽니다.

3. DebugDefs.h 파일에서 관련 매개변수를 설정합니다.

SDKAPPID: 기본값 0, 실제 SDKAppID로 설정하십시오.

SECRETKEY: 기본값 ' ', 실제 키 정보로 설정하십시오.

#### 설명:

본 문서의 UserSig 생성 방법은 클라이언트 코드에서 SECRETKEY를 설정하는 것입니다. 이 방법에서 SECRETKEY 는 디컴파일로 크래킹되기 쉬우므로, 키가 유출되면 해커가 귀하의 Tencent Cloud 트래픽을 도용할 수 있습니다. 따 라서 **해당 방법은 로컬 Demo 실행 및 기능 디버깅용으로만 적합합니다**.

올바른 UserSig 배포 방식은 UserSig 컴퓨팅 코드를 귀하의 서버에 통합하고, App 지향 인터페이스를 제공하는 것입 니다. UserSig가 필요할 때, App은 비즈니스 서버에 동적 UserSig 가져오기 요청을 발송합니다. 자세한 내용은 서버 에서 UserSig 생성을 참고하십시오.

#### 4단계: 컴파일 패키지 실행

1.더블 클릭하여 /IM\_Demo/IM\_Demo.uproject 를 엽니다.

2. 컴파일 실행 디버깅: macOS Windows iOS Android File -> Package Project -> Mac

File->Package Project->Windows->Windows(64-bit)

Choose Files to Save Connect To Source Control Project Wew Project Open Project Wew C++ Class	r Pawn Light r Start	
Package Project	🛉 👘 Android 🛛 🕨	0
Refresh Visual Studio Project	📢 HoloLens	
Open Visual Studio	🔹 iOS	0
Cook Content for Windows	🙏 Linux 🔹 🕨	
Python	斗 Lumin	0
Execute Python Script	tvOS	
🔞 Recent Python Scripts	🖌 🚝 Windows (64-bit)	Build, cook and package your game
DataValidation	Zip Up Project	tvOS is not listed as a target platforn
	Build Configuration	0
Actors Import Into Level	Build Target 🛛 🕨	
Export All	Packaging Settings	0
Export Selected	Supported Platforms	
	e mggel	

프로젝트 패키지

#### File -> Package Project -> iOS

1. 개발 디버깅: 자세한 내용은 Android 시작하기를 참고하십시오.

2. 패키징 프로젝트: 자세한 내용은 Android 프로젝트 패키징을 참고하십시오.

### IM Unreal Engine API 문서

자세한 인터페이스 소개는 API 개요를 참고하십시오.

### FAQ

Android 'Attempt to construct staged filesystem reference from absolute path'''오류 보고

UE4 프로젝트를 비활성화하고 CMD를 열어 다음 명령을 실행합니다.



adb shell

cd sdcard

ls (you should see the UE4Game directory listed)

rm -r UE4Game



프로젝트를 다시 컴파일합니다.

#### Chat

# 시작하기(Flutter)

최종 업데이트 날짜: : 2024-01-31 11:24:44

본문을 읽으면 Flutter SDK를 통합하는 방법을 알 수 있습니다.

### 환경 요건

플랫폼	버전
Flutter	IM SDK에는 최소 Flutter 2.2.0 버전이 필요하고 TUIKit 통합 컴포넌트 라이브러리에는 최소 Flutter 2.10.0 버전이 필요합니다.
Android	Android Studio 3.5 및 그 이후 버전. App은 Android 4.1 및 그 이후 버전 디바이스가 필요합니다.
iOS	Xcode 11.0 및 그 이후 버전. 프로젝트에 유효한 개발자 서명이 설정되어 있는지 확인십시오.

### 플랫폼 지원

당사는 Flutter의 모든 플랫폼을 지원하는 IM SDK 및 TUIKit을 생성하기 위해 최선을 다하고 있으며, 모든 플랫폼에서 하나의 코드 세트를 실행할 수 있도록 지원합니다.

플랫폼	UI 없는 SDK (tencent_cloud_chat_sdk)	UI 및 기본 비즈니스 로직이 포함된 TUIKit (tencent_cloud_chat_uikit)
iOS	지원	지원
Android	지원	지원
Web	v4.1.1+2부터 지원	v0.1.5부터 지원
macOS	v4.1.9부터 지원	곧 출시
Windows	v4.1.9부터 지원	곧 출시
<mark>하이브리드 개발</mark> (기존 네이티브 애플 리케이션에 Flutter SDK 추가)	v5.0.0부터 지원	v1.0.0부터 지원

설명:



Web/macOS/Windows 플랫폼의 경우 통합을 위해 몇 가지 추가 단계가 필요합니다. 자세한 내용은 더 많은 플랫폼으 로 확장을 참고하십시오.

### Demo

통합하기 전에 Tencent Cloud IM Flutter 크로스 플랫폼 SDK 및 TUIKit의 기능을 빠르게 이해하기 위해 DEMO를 사용 해 볼 수 있습니다.

**다음 DEMO는 모두 TUIKit이 도입된 동일한 Flutter 프로젝트로 패키징됩니다.** Flutter용 IM SDK는 이미 Desktop 플 랫폼(macOS/Windows)을 지원하며 해당 DEMO는 곧 출시될 예정입니다.

모바일 APP	WEB - H5
iOS/Android APP, 플랫폼에 따라 자동으로 다운로드됨!	휴대폰으로 QR 코드를 스캔하여 Web DEMO 체험![]
[](https://qcloudimg.tencent-	(https://qcloudimg.tencent-
cloud.cn/raw/ca2aaff551410c74fce48008c771b9f6.png)	cloud.cn/raw/3c79e8bb16dd0eeab35e894a690e0444.

## 준비 작업

1. Signing Up된 계정이 있고 Identity Verification이 완료되어야 합니다.

2. 애플리케이션 생성 및 업그레이드를 참고하여 애플리케이션을 생성하고 'SDKAppID'를 기록합니다.

3. IM 콘솔에서 애플리케이션을 선택하고 왼쪽 사이드바에서 **보조 툴->UserSig Generation & Verification**을 클릭 하여 두 개의 UserID와 해당 UserSig를 생성하고 UserID, 서명(Key), UserSig 를 복사하여 이후 로그인에 사용합니다.

於 勝讯云	云产品 ∨ │ 网站备案 +
即时通信 Ⅲ	← UserSig生成&校验 ▼
弐 基本配置	
品 功能配置	签名(UserSig)生成工具     登录鉴权介绍 I2
晶 群组管理	此工具可以快速生成签名(UserSig),用于本地跑通 Demo 以及功能调试。
③ 回调配置	请输入用户名 (UserID)
🕣 安全打击	user1
数据监控器	密钥 (Key)
@ 辅助工具	
• 离线推送自查	
● UserSig生成&校 验	生成签名 (UserSig)
	当前生成的签名(UserSig)是:
	复制签名 (UserSig)

#### 설명 :

이 계정은 개발 및 테스트 전용입니다. 애플리케이션 런칭 전에 올바른 UserSig 배포 방식은 서버에서 생성하여, App 지향 인터페이스를 제공하는 것입니다. UserSig 가 필요할 때, App은 비즈니스 서버에 동적 UserSig 가 져오기 요청을 발송합니다. 자세한 내용은 Generating UserSig를 참고하십시오.

## Flutter SDK 통합에 적합한 솔루션 선택

IM은 통합하는 세 가지 방법을 제공하며, 통합에 가장 적합한 솔루션을 선택할 수 있습니다.

통합 방법	적용 가능한 시나리오	

<b>DEMO</b> 수정	IM Demo는 완전한 채팅 App이며, 코드는 오픈 소스로, 채팅과 같은 시나리오를 구현해야 하는 경우 Demo를 사용하여 2차 개발을 할 수 있습니다. 바로 Demo 체험할 수 있습니다.
통합 (UI 포 함)	IM의 UI 컴포넌트 라이브러리 TUIKit 은 일반 UI 컴포넌트를 제공하며, 예를 들어 대화 목록, 채팅 인터페이스 및 연락처 목록 등 개발자는 실제 비즈니스 요구에 따라 이 컴포넌트 라이브러리를 통해 사용자 정의 IM 애플리케이션을 빠르게 구축할 수 있습니다. <b>이 방법을 먼저 사용할 것을 권장합니</b> <b>다</b> .
UI 통 합 자 체 구 현 솔 루션	TUIKit이 애플리케이션의 인터페이스 요구 사항을 충족할 수 없거나 더 많은 사용자 정의가 필요한 경우 이 솔루션을 사용할 수 있습니다.

Tencent Cloud IM SDK 내의 모든 API를 더 잘 이해할 수 있도록 개발 초기 단계에서 SDK API를 테스트하고 특정 API 를 호출하기 위한 API Example를 제공합니다.

## 솔루션1: Demo 수정

### Demo 실행

1. Demo 소스 코드를 다운로드하고 종속을 설치합니다.





# Clone the code
git clone https://github.com/TencentCloud/tc-chat-demo-flutter.git

# Install dependencies
flutter pub get

2. Demo 프로젝트를 실행합니다.



#demo 프로젝트를 시작하고 SDK\_APPID 및 KEY의 두 매개변수 교체 flutter run --dart-define=SDK\_APPID={YOUR\_SDKAPPID} --dart-define=ISPRODUCT\_ENV=fal

설명:

```
--dart-define=SDK_APPID={YOUR_SDKAPPID} 중 {YOUR_SDKAPPID} 는 귀하의 애플리케이션
SDKAppID로 대체합니다.
```

--dart-define=ISPRODUCT\_ENV=false 개발 또는 프로덕션 환경을 판단하고 개발 환경이라면 false를 사용 하십시오. --dart-define=KEY={YOUR\_KEY} 중 {YOUR\_KEY} 는 파트1:테스트 사용자 생성 의 키(Key) 정보로 대 체합니다.

#### IDE를 사용하여 실행할 수도 있습니다: (옵션 단계)

Android 플랫폼

iOS 플랫폼

1. Android Studio에 Flutter 및 Dart 플러그인을 설치합니다.

Mac: 플러그인 설정으로 이동하여(v3.6.3.0 이상에서 Preferences > Plugins 선택) => Flutter를 선택하고 설치 클릭 => Dart 플러그인 설치 프롬프트가 표시되면 Yes 클릭 => 다시 시작하라는 메시지가 표시되면 Restart를 클릭합니다. Linux 또는 Windows: 플러그인 설정으로 이동하여(File > Settings > Plugins) = > Marketplace를 선택하고 Flutter plugin을 선택한 다음 Install을 클릭합니다.

	Preferences			
Q	Plugins	Marketplace	Installed <b>①</b>	
<ul> <li>&gt; Appearance &amp; Behavior</li> <li>Keymap</li> <li>&gt; Editor</li> </ul>	Q+ Type / to see op Downloaded (3 of 4	enabled) Update all 🕦	Dart JetBrains	
Plugins 1	Dart 212.5744 Jet	<b>√</b> Brains	Enabled	
<ul> <li>&gt; Version Control</li> <li>&gt; Build, Execution, Deployment</li> <li>&gt; Languages &amp; Frameworks</li> <li>&gt; Tools</li> </ul>	Flutter 67.1.2 → 69.0.1	Update 🗸	Support for Dart Features	
Advanced Settings	Flutter Intl 1.17.3-2020.3		<ul> <li>Smart coding a completion, for refactorings, ar</li> <li>Integration with</li> </ul>	
	Protocol Bu 212.5080.8 Je	atBrains	<ul> <li>The IDE detect on-the-fly and them</li> </ul>	
	Android	Enable all	• Run and debug	
	Smali Supp bundled	ort 🗸	applications rig debugger • Run and debug	
	Build Tools	Enable all	Create new Dar     screen	
	Gradle bundled		Find more information IDE in our docs.	
	Code Coverage	Enable all		
	Code Cover bundled	rage for Java 🗹	Plugin homepage 🛪	
?				

2. 프로젝트를 열고 종속성을 가져옵니다.

Android Studio에서 im-flutter-uikit 디렉터리를 엽니다.
Favorites Recents		im-flutter-uikit	(
<ul> <li>Applicati</li> <li>Desktop</li> <li>Documents</li> <li>Downloads</li> <li>iCloud</li> <li>iCloud Dri</li> <li>Shared</li> </ul>	Yesterday	六月 I flutter_app_badger flutter-samples im-flutter-base im-flutter-uikit-calling im-flutter-uikit-link-preview stream-chat-flutter TPNS-Android-SDK TPNS-Flutter-Plugin	> > > > > > > > > >
Tags ● 红色 ● 橙色 ● 黄色 ● 绿色 ● 蓝色	<ul> <li>■ 720P 三组视频.mov</li> <li>■ 封培视频素材</li> <li>&gt; 明华三班视频</li> <li>&gt; 逆袭阿伟和他的小弟们.mov</li> <li>■ 企业微信截5947a4b3.jpg</li> <li>■ 相片</li> <li>&gt; GitHub</li> </ul>	五月 im_api_example im-flutter-uikit Im-Tlutter-uikit-IDS im-flutter-uikit-push qr_redirect	> > > >
<ul><li>● 紫色</li><li>● 灰色</li></ul>	New Folder		

경로에서 다음 명령을 실행하여 종속성을 설치합니다.





flutter pub get

3. 환경 변수를 구성합니다.

오른쪽 상단 모서리에서 실행 버튼 옆에 있는 main.dart 로 마우스 hover, Edit Configurations 를 선택합 니다.



팝업 창에서 환경 변수(예: SDKAppID)를 입력하여 Additional run args 를 구성합니다. 예시:



```
# SDK_APPID 및 KEY 매개변수 교체
--dart-define=SDK_APPID={YOUR_SDKAPPID} --dart-define=ISPRODUCT_ENV=false --dart-de
```

🕒 🔵 🗧 Run/Debug Configura	
	ntions
+ - 🖻 📭 🐙	
V 🞸 Flutter Name: main.dart	
nain.dart	
Dart entrypoint: /Users/wangr	unlin/Documents/GitHul
The entry-poin	t for the application (e.
Additional run args: art-define	SDK_APPID=.
Additional argu	ments to 'flutter run'.
Additional attach args:	
Additional argu	ments to 'flutter attach
Build flavor:	
An optional bu	ld flavor; either a Gradl
Environment variables:	
Additional envi	ronment variables; only
▼ Before launch	
$+$ $ \checkmark$ $\checkmark$	
Th	ere are no tasks to run l
Show this page 🗸 Activate tool w	indow
Edit configuration templates	
?	

4. Android 시뮬레이터를 생성합니다. 생성한 시뮬레이터를 실행하고 선택합니다.



오른쪽 상단의 Device Manager를 클릭하고 Create devices를 클릭한 다음 시뮬레이터를 생성합니다. Google FCM 푸 시 기능이 필요한 경우 Google Play Store를 지원하는 기기를 설치하는 것이 좋습니다.

							📘 sdk gphone64 arm64 (mobile) 🔻 📔 🎸
• • •			Virtu	al Device Co	onfiguration		Device Manager
							Virtual Physical
🛛 🚺 Sel	ect Hardware						Create device
							Device 🔺
							Pixel 3a API 31 Android 12.0 Google Play   arm64
Choose a dev	vice definition						Pixel XL API 32 ·
						The Pixel 2	Android API 32 Google APIs   arm64
Category		Play Store					
тv	Resizable		6.0"	1080x	420dpi	— 1080ny —	
Phone	Pixel XL		5.5"	1440x	560dpi	Size: large	
Wear OS	Pixel 5		6.0"	1080x	440dpi	Density: 420dpi	
Tablet	Pixel 4a		5.8"	1080x	440dpi	5.0 192000	
Automotive	Pixel 4 XL		6.3"	1440x	560dpi		
	Pixel 4	▶	5.7"	1080x	440dpi		
	Pixel 3a XL		6.0"	1080x	400dpi		
	Pixel 3a		5.6"	1080x	440dpi		
	Divel 2 VI		6.2"	1440×	560dpi		
		<b>N</b>	5.40"	4000	440-4		
		<b>A</b> .	5.46"	1080x	440001		
	Pixel 2 XL		5.99"	1440x	560dpi		
New Hardware	Profile Import Hard	ware Profile:			G	Clone Device	
?						Cancel Previous Next Finish	

5. 프로젝트를 실행합니다.

필요에 따라 왼쪽의 Run 버튼 또는 오른쪽의 Debug 버튼을 클릭하여 프로젝트를 실행합니다.



## 설명 :

UI에 일부 조정 업데이트가 있을 수 있으니 최신 버전을 기준으로 하십시오.

1. Xcode에서 im-flutter-uikit/ios 디렉터리를 엽니다.

Favorites Recents		ios		<b>©</b>
🙏 Applicati	六月	六月		Previous 30 Days
<ul> <li>Desktop</li> <li>Documents</li> <li>Downloads</li> <li>iCloud</li> <li>iCloud Dri</li> <li>Shared</li> </ul>	<ul> <li>1654076866517737.mp4</li> <li>测试视频.mp4</li> <li>闪屏.pptx</li> <li>未命名.mov</li> <li>response</li> <li>五月</li> </ul>	<ul> <li>flutter_app_badger</li> <li>flutter-samples</li> <li>im-flutter-base</li> <li>im-flutter-uikit-calling</li> <li>im-flutter-uikit-link-previev</li> <li>stream-chat-flutter</li> <li>TPNS-Android-SDK</li> <li>TPNS-Flutter-Plugin</li> </ul>	> > > > > > > > > >	六月 README_EN.md README.md 五月 analysis_options.ya android
Tags ● 红色 ● 橙色 ● 黄色 ● 绿色 ● 蓝色	<ul> <li>▼ 720P 三组视频.mov</li> <li>封培视频素材 &gt;</li> <li>明华三班视频 &gt;</li> <li>逆袭阿伟和他的小弟们.mov</li> <li> 企业微信截5947a4b3.jpg</li> <li>相片 &gt;</li> <li>GitHub &gt;</li> </ul>	五月 im_api_example im_flutter-uikit im-flutter-uikit-ibs im-flutter-uikit-push qr_redirect		<ul> <li>assets</li> <li>ios</li> <li>keystore_tuikit.jks</li> <li>lib</li> <li>macos</li> <li>replace_general.js</li> <li>scan_general.js</li> </ul>
● 紫色 ● 灰色	New Folder			

2. iPhone을 연결하고 Build And Run을 클릭한 후 iOS 프로젝트 컴파일이 완료되어 Xcode 프로젝트 창이 팝업될 때 까지 기다립니다.

3. iOS 프로젝트를 열고 기본 Target이 iPhone에서 프로젝트를 실행하도록 Signing & Capabilities(Apple 개발자 계정 필요)를 설정합니다.

4. 프로젝트를 실행하고, 디바이스에서 Demo 디버깅을 진행합니다.



## Demo 코드 구조 개요

## 설명 :

Demo의 UI 및 비즈니스 로직 부분은 Flutter TUIKit을 사용합니다. Demo 레이어 자체는 App을 구축하고, 내비게이션 리디렉션을 처리하고, 인스턴스화된 TUIKit의 개별 컴포넌트 호출에만 사용됩니다.

폴더	소개

lib	프로그램 코어 디렉터리
lib/i18n	국제화 관련 코드. 여기서 국제화에는 TUIKit 자체의 국제화 능력 및 국제화 표제를 포함되지 않 으므로 필요에 따라 가져올 수 있습니다.
lib/src	프로젝트 엔터티 디렉터리
lib/src/pages	이 Demo의 몇 가지 주요 내비게이션 페이지입니다. 프로젝트가 초기화된 후 app.dart 는 로딩 애니메이션을 표시하고 로그인 상태를 판단하여 사용자를 login.dart 또는 home_page.dart 로이동합니다. 사용자가 로그인하면 로그인 정보가 shared_preference 플러그 인을 통해 로컬에 저장됩니다. 이후 애플리케이션을 실행할 때마다 원래 로그인 정보가 로컬에서 발견되면 자동으로 해당 정보를 사용하여 로그인합니다. 이 정보가 없거나 로그인에 실패하면 로그인 페이지로 이동합니다. 자동 로그인 프로세스 동안 사용자는 여전히 app.dart 에 있고 로딩 애니메이션을 볼 수 있습니다. home_page.dart 에는 이 Demo의 네 가지 주요 기능 페이지 전환을 지원하는 하단 Tab이 포함되어 있습니다.
lib/utils	일부 툴 함수 클래스

기본적으로 lib/src 의 각 dart 파일은 TUIKit 컴포넌트를 도입하고 파일에서 컴포넌트를 인스턴스화한 후 페이 지를 렌더링할 수 있습니다.

주요 파일은 다음과 같습니다.

lib/src 주요 파일	파일 소개		
add_friend.dart	친구 추가 신청 페이지, TIMUIKitAddFriend 컴포넌트 사용		
add_group.dart	그룹신청페이지, TIMUIKitAddGroup 컴포넌트사용		
blacklist.dart	차단 리스트 페이지, TIMUIKitBlackList 컴포넌트 사용		
chat.dart	기본 채팅 페이지, 전체 TUIKit 채팅 기능 사용, TIMUIKitChat' 컴포넌트 사용		
chatv2.dart	기본 채팅 페이지, 원자화 기능 사용, TIMUIKitChat 컴포넌트 사용		
contact.dart	연락처 페이지, TIMUIKitContact 컴포넌트 사용		
conversation.dart	대화 목록 인터페이스, TIMUIKitConversation 컴포넌트 사용		
create_group.dart	그룹 채팅 페이지 시작, Demo 단독 구현, 컴포넌트 미사용		
group_application_list.dart	그룹 신청 목록 페이지, TIMUIKitGroupApplicationList 컴포넌트 사용		
group_list.dart	그룹 목록 페이지, TIMUIKitGroup 컴포넌트 사용		
group_profile.dart	그룹 프로필 및 그룹 관리 페이지, TIMUIKitGroupProfile 컴포넌트 사용		

newContact.dart	연락처 친구 신청 페이지, TIMUIKitNewContact 컴포넌트 사용
routes.dart	Demo경로,로그인페이지 login.dart 또는홈페이지 home_page.dart 로이동합니다.
search.dart	전역 검색 및 대화 내 검색 페이지, TIMUIKitSearch (전역 검색) 및 TIMUIKitSearchMsgDetail (대화 내 검색) 컴포넌트 사용
user_profile.dart	사용자 프로필 및 관계망 유지 관리 페이지, TIMUIKitProfile 컴포넌트 사용

대부분의 TUIKit 컴포넌트는 내비게이션 리디렉션 메소드가 필요하므로, Demo 레이어는 'Navigator'를 처리해야 합 니다.

위의 Demo는 2차 개발을 위해 직접 수정하거나 비즈니스 요구 사항을 달성하기 위해 참고할 수 있습니다.

# 솔루션2: UI 라이브러리 및 TUIKit 컴포넌트 라이브러리를 사용하여 반 나절 만에 IM 기능 이식

TUIKit은 Tencent Cloud IM SDK 기반의 UI 컴포넌트 라이브러리로 대화 목록, 채팅 인터페이스, 연락처 목록과 같은 일반적인 UI 컴포넌트를 제공하며 개발자는 실제 비즈니스 요구에 따라 이 컴포넌트 라이브러리를 통해 사용자 정의 IM 애플리케이션을 빠르게 구축할 수 있습니다.TUIKit 그래픽 소개를 참고하십시오.

다음은 TUIKit을 시작하는 데 도움이 되는 간단한 가이드입니다. 자세한 가이드는 TUIKit 기본 기능 통합을 참고하십 시오.



# 전제 조건

Flutter 프로젝트 생성을 완료했거나 구축할 Flutter 프로젝트가 있습니다.

## 액세스 단계

## 권한 설정

TUIKit의 실행으로 인해 촬영/갤러리/녹음/네트워크 등의 권한이 필요하며, Native 파일에 수동으로 선언해야 관련 기 능을 정상적으로 사용할 수 있습니다.

#### Android

android/app/src/main/AndroidManifest.xml 을 열고, <manifest></manifest> 에서 다음 권한을 추가합니다.



```
<uses-permission
   android:name="android.permission.INTERNET"/>
<uses-permission
   android:name="android.permission.RECORD_AUDIO"/>
<uses-permission
   android:name="android.permission.FOREGROUND_SERVICE"/>
<uses-permission
   android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
   android:name="android.permission.VIBRATE"/>
<uses-permission
   android:name="android.permission.ACCESS_BACKGROUND_LOCATION"/>
<uses-permission
   android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission
   android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission
   android:name="android.permission.CAMERA"/>
```

#### iOS

ios/Podfile 을 열고 파일 끝에 다음 권한 코드를 추가합니다.





```
post_install do |installer|
installer.pods_project.targets.each do |target|
flutter_additional_ios_build_settings(target)
target.build_configurations.each do |config|
config.build_settings['GCC_PREPROCESSOR_DEFINITIONS'] ||= [
'$(inherited)',
'PERMISSION_MICROPHONE=1',
'PERMISSION_CAMERA=1',
'PERMISSION_PHOTOS=1',
]
end
```



end end

# 설명 :

푸시 기능을 사용하려면 푸시 관련 권한도 추가해야 하며, 자세한 내용은 Flutter 벤더 메시지 푸시 플러그 인 통합 가 이드를 참고하십시오.

## IM TUIkit 설치

TUlkit에는 이미 IM SDK가 포함되어 있으므로 기본 Chat SDK를 설치할 필요 없이 tencent\_cloud\_chat\_uikit 만 설치하면 됩니다.



```
# 명령 라인에서 실행:
flutter pub add tencent_cloud_chat_uikit
```

프로젝트에 Web 지원이 필요한 경우 후속 단계를 수행하기 전에 Web 호환성 설명 섹션을 참고하여 JS 파일을 가져 오십시오.

## 초기화 API

1. 애플리케이션이 시작되면 TUIKit을 초기화합니다.

2. 먼저 TIMUIKitCore.getInstance() 를 실행한 다음 초기화 함수 init() 를 호출합니다.

sdkAppID 를 전달해야 합니다.

3. API 오류 및 프롬프트를 가져오려면 onTUIKitCallbackListener 리스너를 마운트하는 것이 좋습니다. 자세한 내용은 여기를 참고하십시오.





```
/// main.dart

import 'package:tencent_cloud_chat_uikit/tencent_cloud_chat_uikit.dart';

final CoreServicesImpl _coreInstance = TIMUIKitCore.getInstance();

@override

void initState() {

_coreInstance.init(

_sdkAppID: 0, // Replace 0 with the SDKAppID of your IM application when integr

// language: LanguageEnum.en, // UI 언어. 값을 지정하지 않으면 시스템 언어가 사용됩니

loglevel: LogLevelEnum.V2TIM_LOG_DEBUG,

onTUIKitCallbackListener: (TIMCallback callbackValue){}, // [리스너 구성](https
```

```
listener: V2TimSDKListener());
super.initState();
}
```

## 설명 :

이 단계의 await 초기화가 완료되기 전에 후속 단계를 수행하지 마십시오.

## 테스트 계정으로 로그인

1. 콘솔에서 처음 생성된 테스트 계정을 사용하여 로그인 인증을 완료할 수 있습니다.

2. \_coreInstance.login 메소드를 호출하여 테스트 계정으로 로그인합니다.



```
import 'package:tencent_cloud_chat_uikit/tencent_cloud_chat_uikit.dart';
```

```
final CoreServicesImpl _coreInstance = TIMUIKitCore.getInstance();
_coreInstance.login(userID: userID, userSig: userSig);
```

#### 설명:

이 계정은 개발 및 테스트 전용입니다. 애플리케이션 런칭 전에 올바른 UserSig 배포 방식은 서버에서 생성하여, App 지향 인터페이스를 제공하는 것입니다. UserSig 가 필요할 때, App은 비즈니스 서버에 동적 UserSig 가 져오기 요청을 발송합니다. 자세한 내용은 Generating UserSig를 참고하십시오.



## 구현: 대화 목록 페이지

대화 기록이 있는 모든 사용자와 그룹 채팅 대화를 포함한 대화 목록을 IM 기능의 메인 화면으로 사용할 수 있습니다.



Conversation클래스를 생성하고, body의 TIMUIKitConversation컴포넌트를 사용하여 대화 목록을렌더링합니다.

사용자를 특정 채팅 페이지로 리디렉션하려면 onTapItem 이벤트 처리 함수만 전달하면 됩니다. Chat 클래스 는 다음 단계에서 자세히 설명합니다.





```
import 'package:flutter/material.dart';
import 'package:tencent_cloud_chat_uikit/tencent_cloud_chat_uikit.dart';
class Conversation extends StatelessWidget {
  const Conversation({Key? key}) : super(key: key);
  @override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
    title: const Text(
        "Message",
```

```
style: TextStyle(color: Colors.black),
   ),
 ),
 body: TIMUIKitConversation(
    onTapItem: (selectedConv) {
     Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => Chat(
             selectedConversation: selectedConv,
            ),
          ));
   },
 ),
);
}
}
```

## 구현: 채팅 페이지

페이지는 상단의 엔터티 채팅 기록과 하단의 전송 메시지 모듈로 구성됩니다.

		::!! 중 □)
	My contact	
My co	ntact Call canceled	
	Afternoon 08:19	
	10045	5363
	Unread Chat page	ge
	1004	5363
	10045363: Chat page	
Unread	Here is chat page with TIMUIKitChat	
	Unread	5363
	10045	5363
	https:// intl.cloud.tencent.com/ products/im	
	Instant Messaging   Tencent Chat API for Apps and website. Integrates diversi- chat capabilities such as one-to-one chat, group chat, chat room, and	e
٩		$\bigcirc$ $+$

Chat클래스를 생성하고, body의TIMUIKitChat컴포넌트를 사용하여 채팅 페이지를 렌더링합니다.연락처의 세부 정보 페이지로 리디렉션하려면onTapAvatar이벤트 처리 함수를 전달하는 것이 좋습니다.UserProfile클래스는 다음 단계에서 설명합니다.





```
import 'package:flutter/material.dart';
import 'package:tencent_cloud_chat_uikit/tencent_cloud_chat_uikit.dart';
class Chat extends StatelessWidget {
final V2TimConversation selectedConversation;
const Chat({Key? key, required this.selectedConversation}) : super(key: key);
String? _getConvID() {
return selectedConversation.type == 1
? selectedConversation.userID
: selectedConversation.groupID;
}
```

```
@override
Widget build(BuildContext context) {
return TIMUIKitChat(
   conversationID: _getConvID() ?? '', // groupID or UserID
   conversationType: selectedConversation.type ?? 1, // Conversation type
   conversationShowName: selectedConversation.showName ?? "", // Conversation displa
   onTapAvatar: (_) {
      Navigator.push(
           context,
           MaterialPageRoute(
           builder: (context) => UserProfile(userID: userID),
        ));
   }, // Callback for the clicking of the message sender profile photo. This callbace
);
}
```

## 구현: 사용자 세부 정보 페이지

기본적으로 이 페이지는 사용자가 친구인지 여부를 결정하고 하나의 'userID'만 전달된 후에 사용자 세부 정보 페이지 를 생성합니다.

UserProfile 클래스를 생성하고, body 의 TIMUIKitProfile 컴포넌트를 사용하여 사용자 세부 정보 및 관계 체인 페이지를 렌더링합니다.

#### 설명:

이 페이지를 사용자 정의하려면 먼저 profileWidgetBuilder 를 사용하여 사용자 지정할 profile 컴포넌트를 전 달하고 profileWidgetsOrder 를 사용하여 수직 표시 순서를 결정하는 것을 고려하십시오. 요구 사항을 충족할 수 없는 경우 builder 를 대신 사용하십시오.



11:54		11:58	🗢 🗖 the
	Profile	Profile	
Tencent IM ID: 10045363 Status: Tencent	IM for Flutter	Tencent IM ID: 10045363 Status: Tencent IM for Flutter	
Remarks	Tencent IM Chat official >	Gender	Male
Gender	Male	Date of birth	Unknown
Date of birth	Unknown	Search for chat content	>
Search for chat content	>	Add	
Add to blocklist			
Pin chat to top			
Mute notifications			
Se	end message		
	Voice call		
	Video call		
	Delete		
Add to blocklist Pin chat to top Mute notifications See	end message Voice call Video call Delete	Add	





```
import 'package:flutter/material.dart';
import 'package:tencent_cloud_chat_uikit/tencent_cloud_chat_uikit.dart';
class UserProfile extends StatelessWidget {
    final String userID;
    const UserProfile({required this.userID, Key? key}) : super(key: key);
    @override
    Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
```

```
title: const Text(
    "Message",
    style: TextStyle(color: Colors.black),
    ),
    ),
    body: TIMUIKitProfile(
       userID: widget.userID,
    ),
    );
}
```

이제 애플리케이션은 메시지 송수신을 완료하고, 친구 관계를 관리하고, 사용자 세부 정보를 표시하고, 대화 목록을 표시할 수 있습니다.

## 더 많은 기능

다음 TUIKit 플러그 인을 사용하여 IM 기능을 빠르게 구현할 수 있습니다.

TIMUIKitContact: 연락처 목록 페이지.

TIMUIKitGroupProfile: 사용 모드가 TIMUIKitProfile 과 기본적으로 동일한 그룹 프로필 페이지.

TIMUIKitGroup: 그룹 목록 페이지.

TIMUIKitBlackList: 블록리스트 페이지.

TIMUIKitNewContact: 연락처/친구 요청 목록. 배지를 표시하려면 리스너를 자동으로 마운트할 수 있는

TIMUIKitUnreadCount 배지 컴포넌트를 사용할 수 있습니다.

로컬 검색: TIMUIKitSearch 는 연락처, 그룹 및 채팅 기록의 글로벌 검색을 지원하는 글로벌 검색 컴포넌트입니 다. TIMUIKitSearchMsgDetail 을 사용하여 특정 대화에서 채팅 기록을 검색할 수도 있습니다. 두 가지 검색 모 드 중에서 선택하기 위해 conversation 을 전달할지 여부를 선택할 수 있습니다.

UI 컴포넌트에 대한 전체 보기는 개요 문서 또는 상세 문서를 참고하십시오.

# 솔루션3: 직접 UI 구현

## 전제 조건

Flutter 프로젝트 생성을 완료했거나 구축할 Flutter 프로젝트가 있습니다.

## 액세스 단계

## IM SDK 설치

#### 본 섹션의 상세 문서

다음 명령을 사용하여 최신 버전의 Flutter IM SDK를 설치하십시오. 명령 라인에서 실행:





flutter pub add tencent\_cloud\_chat\_sdk

## 설명 :

프로젝트를 Web 또는 데스크톱(macOS、Windows) 플랫폼에도 적용해야 하는 경우 SDK 통합을 위해 몇 가지 추가 단계를 수행해야 합니다. 자세한 내용은 이 문서의 Web용 Flutter 지원 및 데스크톱용 Flutter 지원을 참고하십시오.

#### SDK 초기화 완료

본 섹션의 상세 문서

'initSDK'를 호출하여 SDK 초기화를 완료하십시오. sdkAppID 를 전달하십시오.



```
import 'package:tencent_cloud_chat_sdk/enum/V2TimSDKListener.dart';
import 'package:tencent_cloud_chat_sdk/enum/log_level_enum.dart';
import 'package:tencent_cloud_chat_sdk/tencent_cloud_chat_sdk.dart';
TencentImSDKPlugin.v2TIMManager.initSDK(
    sdkAppID: 0, // Replace 0 with the SDKAppID of your IM application when integrati
    loglevel: LogLevelEnum.V2TIM_LOG_DEBUG, // Log
    listener: V2TimSDKListener(),
);
```

이 단계에서는 주로 네트워크 상태 및 사용자 정보 변경 등을 포함하여 IM SDK에 대한 일부 리스너를 마운트할 수 있 습니다. 자세한 내용은 V2TimSDKListener 클래스를 참고하십시오.

## 테스트 계정에 로그인

## 본 섹션의 상세 문서

이제 처음에 콘솔에서 생성한 테스트 계정을 사용하여 로그인 인증을 완료할 수 있습니다. 'TencentImSDKPlugin.v2TIMManager.login' 메소드를 호출하여 테스트 계정에 로그인합니다. 반환값 'res.code'가 0이면 로그인이 성공한 것입니다.





```
V2TimCallback res = await TencentImSDKPlugin.v2TIMManager.login(
    userID: userID,
    userSig: userSig,
);
```

#### 설명:

이 계정은 개발 테스트 전용입니다. 애플리케이션이 런칭되기 전에 정확한 UserSig 발급 방식은 다음과 같습니다. UserSig 계산 코드를 귀하의 서버에 통합하고 App 방향의 인터페이스를 제공합니다. UserSig 가 필요할 때 귀하의 App이 비즈니스 서버로 동적 UserSig 를 요청합니다. 자세한 내용은 Generating UserSig를 참고하십시오.

## 메시지 발송

본 섹션의 상세 문서

다음은 문자 메시지 발송 예이며 프로세스는 다음과 같습니다.

1. createTextMessage(String) 를 호출하여 문자 메시지를 생성합니다.

2. 반환 값에 따라 메시지 ID를 가져옵니다.

3. sendMessage() 를 호출하여 해당 ID로 메시지를 발송합니다. receiver 는 이전에 생성한 다른 테스트 계
 정 ID로 입력할 수 있습니다. 하나의 채팅 메시지를 발송할 때 groupID 를 입력할 필요가 없습니다.
 코드 예시:





```
import 'package:tencent_cloud_chat_sdk/tencent_cloud_chat_sdk.dart';
V2TimValueCallback<V2TimMsgCreateInfoResult> createMessage =
    await TencentImSDKPlugin.v2TIMManager
    .getMessageManager()
    .createTextMessage(text: "The text to create");
String id = createMessage.data!.id!; // The message creation ID
V2TimValueCallback<V2TimMessage> res = await TencentImSDKPlugin.v2TIMManager
    .getMessageManager()
```

```
.sendMessage(
    id: id, // Pass in the message creation ID to
    receiver: "The userID of the destination user",
    groupID: "The groupID of the destination group",
    );
```

설명:

만약 sdkAppID가 낯선 사람에게 메시지를 보낼 수 없기 때문에 전송에 실패한다면, 테스트용으로 콘솔에서 활성화할 수 있습니다.

이 링크를 클릭하여, 친구 관계망 확인을 비활성화하십시오.

#### 대화 목록 가져오기

본 섹션의 상세 문서

이전 단계에서 테스트 메시지를 보내면 다른 테스트 계정에 로그인하여 대화 목록을 가져올 수 있습니다.

		::!! 🗢 🗲
	Chats	Ð
	Q Search	
9 (Sec.	0040818 9	Morning 11:57
	Voice Call]: Duration: 00:19	Fri
iii ta	est mute <sup>mage]</sup>	Afternoon 05:00
		Afternoon 04:16
1		Yesterday
F te	Public group elecom number changed Group	Fri notification to
1	0047759	06/24
iii (	ēst Group Name ∌	06/23
<b>i</b>	mage]	06/13
Channels	Chats Contacts	Me

대화 목록을 가져오는 방법은 두 가지입니다.

1. 장기간 연결 콜백을 수신하여 대화 목록을 실시간으로 업데이트합니다.

2. API를 요청하여 페이징에 따라 대화 목록을 한번에 가져옵니다.

일반 응용 시나리오는 다음과 같습니다.

응용 프로그램을 실행한 후 바로 대화 목록을 가져온 다음 장기간 연결을 수신하여 대화 목록의 변경 사항을 실시간 으로 업데이트합니다.

#### 일회성 요청 대화 목록

대화 목록을 가져오려면 현재 위치를 기록하고 'nextSeq'를 유지해야 합니다.





```
import 'package:tencent_cloud_chat_sdk/tencent_cloud_chat_sdk.dart';
String nextSeq = "0";
getConversationList() async {
    V2TimValueCallback<V2TimConversationResult> res = await TencentImSDKPlugin
        .v2TIMManager
        .getConversationManager()
        .getConversationList(nextSeq: nextSeq, count: 10);
    nextSeq = res.data?.nextSeq ?? "0";
```
}

이때 이전 단계에서 다른 테스트 계정에서 보낸 메시지를 볼 수 있습니다.

#### 긴 링크에서 실시간으로 대화 목록 가져오기

이 단계에서는 SDK에 리스너를 마운트한 다음 콜백 이벤트를 처리하고 UI를 업데이트해야 합니다. 1. 리스너 마운트.



await TencentImSDKPlugin.v2TIMManager

- .getConversationManager()
- .setConversationListener(

2. 콜백 이벤트를 처리하고 UI에 최신 대화 목록을 표시합니다.



import 'package:tencent\_cloud\_chat\_sdk/tencent\_cloud\_chat\_sdk.dart';

```
List<V2TimConversation> _conversationList = [];
_onConversationListChanged(List<V2TimConversation> list) {
  for (int element = 0; element < list.length; element++) {
    int index = _conversationList.indexWhere(
        (item) => item!.conversationID == list[element].conversationID);
  if (index > -1) {
    _conversationList.setAll(index, [list[element]]);
  } else {
    _conversationList.add(list[element]);
  }
  }
}
```

### 메시지 수신

### 본 섹션의 상세 문서

Tencent Cloud IM Ffitter SDK를 통해 메시지를 수신하는 두 가지 방법:

1. 장기간 연결 콜백을 수신하고, 실시간으로 메시지 변경 사항을 가져오고, 렌더링 메시지 기록 목록을 업데이트합니 다.

2. API를 요청하여 페이징에 따라 메시지 기록를 한번에 가져옵니다.

일반 응용 시나리오는 다음과 같습니다.

1. 인터페이스는 새로운 대화가 시작되면 먼저 일정량의 메시지 기록을 한꺼번에 요청하여 메시지 기록 목록을 보여 줍니다.

2. 긴 링크를 수신하여 실시간으로 새로운 메시지를 수신하여 메시지 기록 목록에 추가합니다.

### 일회성 요청 메시지 기록 리스트

페이지당 풀링하는 메시지의 수는 너무 크면 안 됩니다. 그렇지 않으면 풀링 속도에 영향을 줄 수 있습니다. 약 20으로 설정하는 것이 좋습니다.

다음 요청 시 현재 페이지 수를 동적으로 기록해야 합니다.

예시 코드는 다음과 같습니다.





```
import 'package:tencent_cloud_chat_sdk/tencent_cloud_chat_sdk.dart';

V2TimValueCallback<List<V2TimMessage>> res = await TencentImSDKPlugin

.v2TIMManager

.getMessageManager()

.getGroupHistoryMessageList(

groupID: "groupID",

count: 20,

lastMsgID: "",

);
```

```
List<V2TimMessage> msgList = res.data ?? [];
// here you can use msgList to render your message list
}
```

### 긴 링크를 통해 실시간으로 새 메시지 가져오기

메시지 기록 리스트가 초기화되면 V2TimAdvancedMsgListener.onRecvNewMessage 라는 긴 링크에서 새 메 시지가 나타납니다.

onRecvNewMessage 콜백이 트리거된 후, 필요에 따라 새 메시지를 메시지 기록 리스트에 추가할 수 있습니다. 바인딩 리스너의 예시 코드는 다음과 같습니다.





```
import 'package:tencent_cloud_chat_sdk/tencent_cloud_chat_sdk.dart';
final adVancesMsgListener = V2TimAdvancedMsgListener(
onRecvNewMessage: (V2TimMessage newMsg) {
  _onReceiveNewMsg(newMsg);
},
/// ... other listeners related to message
);
TencentImSDKPlugin.v2TIMManager
  .getMessageManager()
```

.addAdvancedMsgListener(listener: adVancesMsgListener);

이제 IM 모듈 개발을 완료했으며, 메시지를 주고받거나 다른 대화에 들어갈 수 있습니다. 귀하는 계속해서 그룹, 사용자 프로필, 관계망, 오프라인 푸시, 로컬 검색 등 관련 기능을 개발할 수 있습니다. 자세한 내용은 통합 솔루션(UI 없음)을 참고하십시오.

## 고급 기능 통합

### 더 많은 플러그인으로 Flutter IM 경험 강화

SDK 및 TUIKit 기본 기능 외에도 IM 기능을 강화하는 데 도움이 되는 4가지 선택적 플러그인을 제공합니다. 메시지 푸시 플러그인: 벤더의 네이티브 오프라인 푸시 기능을 지원하고 다른 비즈니스 메시지 푸시를 지원하여 메시 지 도달률을 개선할 수 있습니다.

통합 코드는 Demo를 참고하십시오.

설명:

좋은 아이디어나 제안이 있으시면 언제든지 고객센터로 연락주십시오.

### 더 많은 플랫폼으로 확장

Tencent Cloud IM for Flutter SDK는 기본적으로 Android / iOS 플랫폼을 지원합니다. 더 많은 플랫폼(Web/Desktop) 으로 확장할 수도 있습니다.

### Flutter for Web 지원

당사의 SDK, TUIKit(tencent\_cloud\_chat\_uikit) 0.1.5 이상 및 no-UI SDK(tencent\_cloud\_chat\_sdk) 4.1.1+2 이상은 Web과 완벽하게 호환됩니다.

Web 지원을 활성화하려면 Android 및 iOS 지원을 활성화하는 단계와 비교하여 다음과 같은 추가 단계를 수행해야 합니다.

### Flutter 3.x로 업그레이드

Flutter 3.x는 Web 성능에 크게 최적화되었으며 Flutter Web 프로젝트 개발에 적극 권장됩니다.

### Flutter for Web 플러그인 SDK 가져오기





flutter pub add tencent\_im\_sdk\_plugin\_web

### JS 가져오기

### 설명:

기존 Flutter 프로젝트가 Web을 지원하지 않는 경우 프로젝트의 루트 디렉터리에서 flutter create . 를 실행 하여 Web 지원을 추가하십시오.

프로젝트의 web/ 디렉터리로 이동하고 npm 또는 yarn 을 사용하여 관련 JS 종속성을 설치합니다. 프로젝트 를 초기화하려면 화면의 지시를 따르십시오.





cd web				
npm init				
npm i tim-js-sdk				
npm i tim-upload-plugin				
web/index.html 을 열고, <head> </head> JS 파일을 가져옵니다.아래를 참고하십시오.				





<script src="./node\_modules/tim-upload-plugin/index.js"></script>
<script src="./node\_modules/tim-js-sdk/tim-js-friendship.js"></script></script></script>

		)								im-flutte	r-uikit – index.h
im	flutt	er-uikit 👌 web 👌 <del></del> index.l	ntml							📘 sdk gphone64 a	m64 (mobile) 🔻
ដ	F	Project <del>v</del>	⊕ <u>₹</u>	×.	<b>\$</b> -	el.dart	×	aco	nversation_services.dart 🗙 🚜 open_file.dart 🗙	🛔 pubspec.lock ×	🛃 platform_chan
Proje		k build							name="keywords"		
		🛄 ios							content="Chat SDK, Tencent Cloud	Chat, 腾讯云IM,	Flutter Chat
		🖿 lib							/>		
Imm		🖿 macos									
S		Package_src				30			iOS meta tags & icons		
burce Manager 🛛 🍳		🗖 web				31			<pre><meta content:<="" name="annle-mohile-web-ann-c;&lt;/pre&gt;&lt;/td&gt;&lt;td&gt;anahle" td=""/><td>"ves" /&gt;</td></pre>	"ves" />	
		> icons				32			<pre><meta <="" name="apple mobile-web-apple" pre=""/></pre>	itle" content="1	encent Cloud
						33			<pre><li><li>ink pel="apple-touch-icon" hpef="""</li></li></pre>	" /favicon ico"	/s
		Tim-js-sak				34		Γ	<pre>// /pode modules/tim-up</pre>	load-nlugin/inde	y is">
		IIII-upioad-piu a package-locki				35			<pre><script sic="./node_modules/tim-op&lt;/pre"></script></pre>		

### Flutter for Desktop(PC) 지원

당사의 no-UI SDK(tencent\_cloud\_chat\_sdk) 4.1.9 이상은 macOS 및 Windows 클라이언트와 완벽하게 호환됩니다. macOS 및 Windows에 대한 지원을 활성화하려면 Android 및 iOS에 대한 지원을 활성화하는 단계와 비교하여 다음 추가 단계를 수행해야 합니다.

### Flutter 3.x로 업그레이드

Flutter 3.0 이상에서만 desktop 클라이언트를 지원합니다. 따라서 desktop 클라이언트를 사용해야 하는 경우 Flutter 를 Flutter 3.x로 업그레이드하십시오.

### Flutter for Desktop 플러그인 SDK 가져오기





flutter pub add tencent\_im\_sdk\_plugin\_desktop

### macOS 구성 수정

macos/Runner/DebugProfile.entitlements파일을 엽니다.<dict></dict></dict>에서 다음key-value키 값 쌍을 추가합니다.





<key>com.apple.security.app-sandbox</key><false/>

## FAQ

iOS에서 Pods 종속성 설치에 실패하면 어떻게 해야 하나요?

솔루션1: 구성 후, 오류가 발생하면 Product > Clean Build Folder를 클릭하고 제품을 지운 후 pod install 또 는 flutter run 을 다시 실행하십시오.

<b>É Xcode</b> File Edit View Find	Navigate Editor	Product Debug Source Control V	Vindow
	Runner feat/push_support	Run %R Test %U	3 Pro
► X II Q A ◇ Ø ▷ E	Runner	Profile#  Analyze쇼 # B	list (zh-Ha
> 📰 Flutter < 🛅 Runner 🐼 Runner	PROJECT	Archive Build For > Perform Action >	ng & Capal
> X Main	🛃 Runner	Build & B	
<ul> <li>LaunchScreen</li> <li>LaunchScreen (Base)</li> <li>LaunchScre, Traditional)</li> </ul>	TARGETS	Clean Test Results へて発K Stop 発。	
<ul> <li>LaunchScree, Simplified)</li> <li>Info</li> </ul>		Build Documentation ^ & # D	
<ul> <li>InfoPlist</li> <li>InfoPlist (English)</li> <li>InfoPlist (Ch, Traditional)</li> </ul>		Export Localizations >	
H GeneratedPluginRegistrant     GeneratedPluginRegistrant     AppDelegate		Scheme>Destination>Test Plan>	Dŧ
h Runner-Bridging-Header		Xcode Cloud >	

솔루션2: ios/Pods 폴더와 ios/Podfile.lock 파일을 수동으로 삭제하고 다음 명령을 실행하여 종속성을 다시 설치합니다.

1. M1과 같은 최신 Apple Silicon 칩 시리즈가 탑재된 Mac 장치.









cd ios sudo arch -x86\_64 gem install ffi arch -x86\_64 pod install --repo-update

2. 이전 버전의 Intel 칩이 탑재된 Mac 장치.



```
cd ios
sudo gem install ffi
pod install --repo-update
```

Apple Watch를 착용한 상태에서 실제 iOS 기기에서 디버깅 중 오류가 발생하면 어떻게 해야 하나요?

Ineligible destinations for the "Runner" scheme:
{ platform:iOS, id:00008020-000A61AA2686002E, name:王润霖的 iPhone, error:Device is busy (Preparing the
{    platform:iOS, id:dvtdevice-DVTiPhonePlaceholder-iphoneos:placeholder, name:Any iOS Device }
{    platform:iOS Simulator, id:dvtdevice-DVTiOSDeviceSimulatorPlaceholder-iphonesimulator:placeholder,

Apple Watch를 비행 모드로 조정해 주시고, iPhone의 설정 => 블루투스 로 이동하여 블루투스를 끕니다. Xcode를 다시 시작하고(열린 경우), flutter run 을 다시 실행합니다.

### Flutter 환경 문제

Flutter 환경에 문제가 있는지 알려면 Flutter Doctor를 실행하여 Flutter 환경이 설치되어 있는지 점검하십시오.

Flutter에서 자동으로 생성된 항목을 사용하여 TUIKit을 도입한 후 Android를 실행하면 오류가 보고됨



1. android\\app\\src\\main\\AndroidManifest.xml 을 열고,다음과 같이

xmlns:tools="http://schemas.android.com/tools" /

```
android:label="@string/android_label" 및 tools:replace="android:label" 을 완료합니다.
```





```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="Android 패키지 이름으로 교체"
xmlns:tools="http://schemas.android.com/tools">
<application
android:label="@string/android_label"
tools:replace="android:label"
android:icon="@mipmap/ic_launcher" // icon 경로 지정
android:usesCleartextTraffic="true"
android:requestLegacyExternalStorage="true">
```

2. android\\app\\build.gradle 을열고 defaultConfig 에서 minSdkVersion 및 targetSdkVersion 을완료합니다.



```
defaultConfig {
applicationId "" // Android 패키지 이름으로 교체
minSdkVersion 21
targetSdkVersion 30
}
```

### 에러 코드를 쿼리하는 방법은 무엇입니까?



IM SDK API 에러 코드는 에러 코드를 참고하십시오.

TUIKit 시나리오 코드는 팝업 프롬프트에 사용되며 onTUIKitCallbackListener 를 수신하여 얻을 수 있습니다. 시나리 오 코드 목록은 여기를 참고하십시오.

### Chat

# 시작하기 (React Native)

최종 업데이트 날짜: : 2024-01-31 11:27:48

본문에서는 React Native용 IM(Instant Messaging) Demo를 빠르게 실행하는 방법을 소개합니다.

## 환경 요건

	버전
React Native	0.63.4 이후 버전
Android	Android Studio 3.5 및 그 이후 버전. App은 Android 4.1 및 그 이후 버전 디바이스가 필요합니다.
iOS	Xcode 11.0 이후 버전, 프로젝트에 대한 유효한 개발자 서명이 설정되어 있는지 확인하십시오.

## 전제 조건

Signing Up된 계정이 있고 Identity Verification이 완료되어야 합니다.

## 파트1: 테스트 사용자 생성

IM 콘솔에서 애플리케이션을 선택하고 왼쪽 사이드바에서 **보조 툴->UserSig Generation & Verification**을 클릭하 여 두 개의 UserID와 해당 UserSig를 생성하고 UserID, 서명(Key), UserSig 를 복사하여 이후 로그인에 사 용합니다.

### 설명:

이 계정은 개발 및 테스트 전용입니다. 애플리케이션 런칭 전에 올바른 UserSig 배포 방식은 서버에서 생성하여, App 지향 인터페이스를 제공하는 것입니다. UserSig 가 필요할 때, App은 비즈니스 서버에 동적 UserSig 가 져오기 요청을 발송합니다. 자세한 내용은 서버의 UserSig 생성을 참고하십시오.

Instant Messaging	← UserSig Generation & Verification
금는 Basic Configuration	Signature (UserSig) Generator
Feature * Configuration	This tool can quickly generate a UserSig, which can be used to run through demos
器 Group Management	Username (UserID)
Callback Configuration	Key
🗟 Data Monitor 🛛 👻	
🖬 Auxiliary Tools 🛛 ^	
<ul> <li>Push Message Tool</li> </ul>	
UserSig Tools	Generate UserSig
	Current Signature (UserSig)
	Copy UserSig
Ξ	copy osciolog

## 파트2: React Native SDK 통합

### 전제 조건

React Native 프로젝트를 생성했거나 또는 기반으로 할 수 있는 React Native 프로젝트가 있습니다.

### 액세스 단계



### IM SDK 설치

다음 명령을 사용하여 최신 버전의 React Naitve IM SDK를 설치합니다. 명령 라인에서 실행:



// npm
npm install react-native-tim-js

```
// yarn
yarn add react-native-tim-js
```

SDK 초기화 완료

'initSDK'를 호출하여 SDK 초기화를 완료하십시오. sdkAppID 를 전달합니다.



```
import { TencentImSDKPlugin, LogLevelEnum } from 'react-native-tim-js';
TencentImSDKPlugin.v2TIMManager.initSDK(
    sdkAppID: 0, // Replace 0 with the SDKAppID of your IM application when integra
    loglevel: LogLevelEnum.V2TIM_LOG_DEBUG, // Log
    listener: V2TimSDKListener(),
);
```

이 단계에서는 주로 네트워크 상태 및 사용자 정보 변경 등 IM SDK에 대한 일부 리스너를 마운트할 수 있습니다. 자세 한 내용은 이 문서를 참고하십시오.

### 테스트 계정에 로그인

1. 이제 처음에 콘솔에서 생성된 테스트 계정을 사용하여 로그인 인증을 완료할 수 있습니다.

2. 'TencentImSDKPlugin.v2TIMManager.login' 메소드를 호출하여 테스트 계정에 로그인합니다. 반환값 'res.code'가 0이면 로그인이 성공한 것입니다.



import { TencentImSDKPlugin } from 'react-native-tim-js'; const res = await TencentImSDKPlugin.v2TIMManager.login(

```
userID: userID,
userSig: userSig,
);
```

### 설명 :

이 계정은 개발 테스트 전용입니다. 애플리케이션이 런칭되기 전에 정확한 UserSig 발급 방식은 다음과 같습니다. UserSig 계산 코드를 귀하의 서버에 통합하고 App 방향의 인터페이스를 제공합니다. UserSig 가 필요할 때 귀하의 App이 비즈니스 서버로 동적 UserSig 를 요청합니다. 자세한 내용은 서버의 UserSig 생성을 참고하십시오.

### 메시지 발송

다음은 문자 메시지 발송 예이며 프로세스는 다음과 같습니다.

1. createTextMessage(String) 를 호출하여 문자 메시지를 생성합니다.

2. 반환 값에 따라 메시지 ID를 가져옵니다.

 sendMessage() 를 호출하여 해당 ID로 메시지를 발송합니다. receiver 는 이전에 생성한 다른 테스트 계 정 ID로 입력할 수 있습니다. 하나의 채팅 메시지를 발송할 때 group ID 를 입력할 필요가 없습니다.
 코드 예시:

Chat





```
import { TencentImSDKPlugin } from 'react-native-tim-js';
const createMessage =
    await TencentImSDKPlugin.v2TIMManager
       .getMessageManager()
       .createTextMessage("The text to create");
const id = createMessage.data!.id!; // The message creation ID
const res = await TencentImSDKPlugin.v2TIMManager
       .getMessageManager()
```

```
.sendMessage(
    id: id, // Pass in the message creation ID to
    receiver: "The userID of the destination user",
    groupID: "The groupID of the destination group",
    );
```

설명:

만약 sdkAppID가 낯선 사람에게 메시지를 보낼 수 없기 때문에 전송에 실패한다면, 테스트용으로 콘솔에서 활성화할 수 있습니다.

이 링크를 클릭하여, 친구 관계망 확인을 비활성화하십시오.

### 대화 목록 가져오기

이전 단계에서 테스트 메시지를 보내면 다른 테스트 계정에 로그인하여 대화 목록을 가져올 수 있습니다. 대화 목록을 가져오는 방법은 두 가지입니다.

1. 장기간 연결 콜백을 수신하여 대화 목록을 실시간으로 업데이트합니다.

2. API를 요청하여 페이징에 따라 대화 목록을 한번에 가져옵니다.

일반 응용 시나리오는 다음과 같습니다.

응용 프로그램을 실행한 후 바로 대화 목록을 가져온 다음 장기간 연결을 수신하여 대화 목록의 변경 사항을 실시간 으로 업데이트합니다.

#### 일회성 요청 대화 목록

대화 목록을 가져오려면 현재 위치를 기록하고 'nextSeq'를 유지해야 합니다.





```
import { useState } from "react";
import { TencentImSDKPlugin } from "react-native-tim-js";
const [nextSeq, setNextSeq] = useState<string>("0");
const getConversationList = async () => {
  const count = 10;
  const res = await TencentImSDKPlugin.v2TIMManager
  .getConversationManager()
  .getConversationList(count, nextSeq);
  setNextSeq(res.data?.nextSeq ?? "0");
```

### Chat

### };

이제 이전 단계에서 다른 테스트 계정을 사용하여 보낸 메시지의 대화를 볼 수 있습니다.

### 긴 링크에서 실시간으로 대화 목록 가져오기

이 단계에서는 SDK에 리스너를 마운트한 다음 콜백 이벤트를 처리하고 UI를 업데이트해야 합니다. 1. 리스너 마운트.



import { TencentImSDKPlugin } from "react-native-tim-js";

```
const addConversationListener = () => {
```

```
TencentImSDKPlugin.v2TIMManager
.getConversationManager()
.addConversationListener({
    onNewConversation: (conversationList) => {
        // new conversation created callback
        _onConversationListChanged(conversationList);
    },
    onConversationChanged: (conversationList) => {
        // conversation changed callback
        _onConversationListChanged(conversationList);
    },
    });
};
```

2. 콜백 이벤트를 처리하고 최신 대화 목록을 인터페이스에 표시합니다.





```
const _onConversationListChanged = (list) => {
   // you can use conversation list to update UI
};
```

### 메시지 수신

Tencent Cloud IM React Native SDK를 통해 메시지를 수신하는 방법에는 두 가지가 있습니다:

1. 장기간 연결 콜백을 수신하고, 실시간으로 메시지 변경 사항을 가져오고, 렌더링 메시지 기록 목록을 업데이트합니다.

2. API를 요청하여 페이징에 따라 메시지 기록를 한번에 가져옵니다.



일반 응용 시나리오는 다음과 같습니다.

1. 인터페이스는 새로운 대화가 시작되면 먼저 일정량의 메시지 기록을 한꺼번에 요청하여 메시지 기록 목록을 보여 줍니다.

2. 긴 링크를 수신하여 실시간으로 새로운 메시지를 수신하여 메시지 기록 목록에 추가합니다.

#### 일회성 요청 메시지 기록 리스트

페이지당 풀링하는 메시지의 수는 너무 크면 안 됩니다. 그렇지 않으면 풀링 속도에 영향을 줄 수 있습니다. 약 20으로 설정하는 것이 좋습니다.

다음 요청 시 현재 페이지 수를 동적으로 기록해야 합니다.

예시 코드는 다음과 같습니다.





```
import { TencentImSDKPlugin } from "react-native-tim-js";
const getGroupHistoryMessageList = async () => {
  const groupID = "";
  const count = 20;
  const lastMsgID = "";
  const res = await TencentImSDKPlugin.v2TIMManager
  .getMessageManager()
  .getGroupHistoryMessageList(groupID, count, lastMsgID);
  const msgList = res.data ?? [];
  // here you can use msgList to render your message list
```

### };

### 긴 링크를 통해 실시간으로 새 메시지 가져오기

메시지 기록 리스트가 초기화되면 V2TimAdvancedMsgListener.onRecvNewMessage 라는 긴 링크에서 새 메 시지가 나타납니다.

onRecvNewMessage 콜백이 트리거된 후, 필요에 따라 새 메시지를 메시지 기록 리스트에 추가할 수 있습니다. 바인딩 리스너의 예시 코드는 다음과 같습니다.



import { TencentImSDKPlugin } from "react-native-tim-js";
```
const adVancesMsgListener = {
    onRecvNewMessage: (newMsg) => {
      __onReceiveNewMsg(newMsg);
      /// ... other listeners related to message
    },
  };
const addAdvancedMsgListener = () => {
    TencentImSDKPlugin.v2TIMManager
      .getMessageManager()
      .addAdvancedMsgListener(adVancesMsgListener);
};
```

이제 IM 모듈 개발을 완료했으며, 메시지를 주고받거나 다른 대화에 들어갈 수 있습니다. 그룹, 사용자 프로필, 관계망, 오프라인 푸시, 로컬 검색 등 관련 기능을 계속 개발할 수 있습니다. 자세한 내용은 SDK API 문서를 참고하십시오.

### FAQ

demo 실행 시 Undefined symbols for architecture x86\_64 [duplicate] 를 해결하는 방법은 무엇 입니까?

<mark>문서</mark>를 참고하십시오.

demo 실행 시 Failed to resolve: react-native-0.71.0-rc.0-debug 를 해결하는 방법은 무엇입니 까?

<mark>문서</mark>를 참고하십시오.

## 시작하기(기존 애플리케이션에 Flutter 추가)

최종 업데이트 날짜: : 2024-01-31 11:38:07

이 솔루션을 사용하면 Tencent Cloud IM Flutter SDK를 기존 Android / iOS 애플리케이션에 통합할 수 있습니다. Flutter에서 전체 애플리케이션을 한 번에 다시 작성하는 것이 실용적이지 않은 경우가 있습니다. 이러한 상황에서 Flutter는 라이브러리 또는 모듈로 기존 APP에 단편적으로 통합될 수 있습니다. 그런 다음 해당 모듈을 Android 또는 iOS(현재 지원되는 플랫폼) APP으로 가져와 Flutter에서 APP UI의 일부를 렌더링할 수 있습니다. **기존의 IM 및 통화 모듈을 추가하여 워크로드를 크게 줄일 수 있습니다.** 



### 환경 요건

환경	버전
Flutter	SDK에는 최소 Flutter 2.2.0 버전이 필요하고 TUIKit 통합 컴포넌트 라이브러리에는 최소 Flutter 2.10.0 버전이 필요합니다.

Android	Android Studio 3.5 및 그 이후 버전. App은 Android 4.1 및 그 이후 버전 디바이스가 필요합 니다.
iOS	Xcode 11.0 및 그 이후 버전. 프로젝트에 유효한 개발자 서명이 설정되어 있는지 확인십시 오.
Tencent Cloud IM SDK	tencent_im_sdk_plugin 5.0 이후 버전, tim_ui_kit 0.2 이후 버전.

### 설명 :

상기 Demo 프로젝트의 소스 코드는 GitHub 레지스트리에서 찾아 확인할 수 있습니다.

### 사전 지식

시작하기 전에 Tencent Cloud IM Flutter SDK 및 TUIKit의 사용법에 대해 알고 기존 앱에 Flutter 모듈을 추가하는 것이 좋습니다.

### Tencent Cloud IM

### 시작하기

시작하기 전에 Tencent Cloud IM Flutter의 SDK 및 기본 사용법에 대해 잘 알고 있어야 합니다. non-UI SDK 및 TUIKit의 두 가지 주요 SDK가 포함됩니다. 이 튜토리얼에서는 주로 UI 라이브러리와 기본 비즈니스 로직을 사용하여 TUIKit으로 개발합니다.

시작하기를 통해 Tencent Cloud IM Flutter에 대해 알 수 있습니다.

### 두 개의 모듈

Tencent Cloud IM에는 두 가지 주요 모듈인 Chat과 Call이 포함됩니다. Chat 모듈에는 메시지 송수신, 대화 관리, 관계 관리 등이 포함됩니다. Call 모듈에는 일대일 통화 및 그룹 통화를 포함한 음성 통화 및 영상 통화가 포함됩니다.

### 네이티브 앱에 Flutter 추가

이 솔루션의 핵심은 Flutter module을 서브 프로젝트의 Native 애플리케이션에 포함하는 것입니다. Flutter의 크로스 플랫폼 기능으로 단일 Flutter module을 Android/iOS 프로젝트 모두에 추가할 수 있습니다.

기존 애플리케이션이 Tencent Cloud IM 관련 페이지를 표시해야 하는 경우 Flutter를 호스팅하는 해당 Activity(Android) 또는 ViewController(iOS)를 로딩할 수 있습니다.

Method Channel은 현재 사용자 정보, 오프라인 푸시 및 음성/영상 통화 데이터의 EXT 전송과 같이 필요한 경우 Native APP와 Flutter 모듈 간에 통신하는 데 사용할 수 있습니다. 메소드 채널에서 메소드를 호출하고 미리 설정된 MethodCallHandler로 호출을 수신합니다.

### Android 앱에 추가

### 세부 문서

Gradle에서 Flutter module을 기존 앱의 종속성으로 추가합니다. 이를 구현하는 방법에는 두 가지가 있습니다.

### Android 방법1: Android Archive (AAR)에 따라 다름

AAR 메커니즘은 Flutter module 패키징을 위한 중개자로서 일반 Android AAR을 생성합니다. 자주 빌드하는 경우 빌 드 단계를 추가합니다.

이 방법은 Flutter 라이브러리를 AAR 및 POM 아티팩트로 구성된 일반 로컬 Maven 리포지토리로 패키징합니다. 이 방법을 사용하면 팀에서 Flutter SDK를 설치하지 않고도 호스트 앱을 빌드할 수 있습니다. 그런 다음 로컬 또는 원격 리포지토리에서 아티팩트를 배포할 수 있습니다.

릴리스된 버전에 대해 이 방법을 사용하는 것이 좋습니다.

### 자세한 방법:

Flutter module에서 다음 명령을 실행합니다.





flutter build aar

그런 다음 화면의 지시에 따라 통합합니다.



이제 앱에 Flutter 모듈이 종속 항목으로 포함됩니다.

### Android 방법2: Flutter module의 소스 코드에 따라 다름

소스 코드 서브 프로젝트 메커니즘은 편리한 원스텝 빌드 프로세스이지만 Flutter SDK가 필요합니다. 이것은 Android Studio IDE 플러그인에서 사용하는 메커니즘입니다.

이 방법을 사용하면 Android 프로젝트와 Flutter 프로젝트 모두에 대해 원스텝 빌드가 가능합니다. 이 방법은 두 부분 에서 동시에 작업하고 빠르게 반복할 때 편리하지만 팀에서 호스트 앱을 빌드하려면 Flutter SDK를 설치해야 합니다. 개발 및 디버깅 시 이 방법을 사용하는 것이 좋습니다.

### 자세한 방법:

호스트 앱의 settings.gradle 에 서브 프로젝트로 Flutter module을 추가합니다.



```
// Include the host app project.
include ':app'
setBinding(new Binding([gradle: this]))
evaluate(new File(
```

```
settingsDir.parentFile, // new
'tencent_chat_module/.android/include_flutter.groovy' // new
)) // new
```

애플리케이션에서 app/build.gradle => dependencies 에서 Flutter module의 implementation 을 가져옵니다.



```
dependencies{
    implementation project(':flutter')
}
```

이제 앱에 Flutter 모듈이 종속 항목으로 포함됩니다.

### Flutter 모듈을 iOS 앱에 추가

### 세부문서

기존 애플리케이션에 Flutter를 임베드하는 방법에는 두 가지가 있습니다.

### iOS 방법1: CocoaPods 및 Flutter SDK 임베드

CocoaPods 종속 항목 관리자를 사용하고 Flutter SDK를 설치합니다. 이 방법을 사용하려면 프로젝트에서 작업하는 모든 개발자가 Flutter SDK의 로컬 버전을 설치해야 합니다.

스크립트를 자동으로 실행하여 DART 및 플러그인 코드를 포함하도록 Xcode에서 애플리케이션을 빌드하기만 하면 됩니다. 이를 통해 Xcode 외부에서 추가 명령을 실행하지 않고도 Flutter 모듈의 최신 버전으로 빠르게 반복할 수 있습 니다.

개발 및 디버깅 시 이 방법을 사용하는 것이 좋습니다.

### 자세한 방법:

Podfile에 다음 코드를 추가합니다.





// Flutter Module의 경로 flutter\_chat\_application\_path = '../tencent\_chat\_module'

load File.join(flutter\_chat\_application\_path, '.ios', 'Flutter', 'podhelper.rb')

#### Flutter를 임베드해야 하는 각 Podfile target에 대해

install\_all\_flutter\_pods(flutter\_chat\_application\_path) 을 호출합니다.





```
target 'MyApp' do
    install_all_flutter_pods(flutter_chat_application_path)
end
```

Podfile의post\_install블록에서flutter\_post\_install(installer)를 호출하고 마이크 권한/카메라 권한/앨범 권한을 포함하여Tencent Cloud IM TUIKit에서 요구하는 권한 성명을 완료합니다.





```
post_install do |installer|
flutter_post_install(installer) if defined?(flutter_post_install)
installer.pods_project.targets.each do |target|
flutter_additional_ios_build_settings(target)
target.build_configurations.each do |config|
config.build_settings['GCC_PREPROCESSOR_DEFINITIONS'] ||= [
    '$(inherited)',
    'PERMISSION_MICROPHONE=1',
    'PERMISSION_CAMERA=1',
    'PERMISSION_PHOTOS=1',
]
```



```
end
end
```

'pod install'을 진행합니다.

### 설명 :

tencent\_chat\_module/pubspec.yaml 에서 Flutter 플러그인 종속성을 변경할 때 Flutter Module 디렉터리에 서 flutter pub get 을실행하여 podhelper.rb 스크립트에서 읽은 플러그인 목록을 새로 고칩니다.그다 음 애플리케이션의 루트 디렉터리에서 pod install 을 다시 실행합니다.

Apple Silicon 칩 arm64 아키텍처가 있는 Mac 컴퓨터에서 arch -x86\_64 pod install --repo-update 를 실행해야 할 수도 있습니다.

podhelper.rb 스크립트는 플러그인,/ Flutter.framework / App.framework 를 프로젝트에 임베드합 니다.

### iOS 방법2: Xcode에 frameworks 임베드

Flutter 엔진, 컴파일된 DART 코드 및 모든 Flutter 플러그인용 프레임워크를 생성합니다. 프레임워크를 수동으로 임 베드하고 Xcode에서 기존 앱의 빌드 설정을 업데이트합니다.

또는 기존 Xcode 프로젝트를 수동으로 편집하여 필요한 framework를 생성하고 애플리케이션에 임베드할 수 있습니 다. 팀 구성원이 Flutter SDK 및 CocoaPods를 로컬에 설치할 수 없거나 기존 애플리케이션에서 CocoaPods를 종속 항목 관리자로 사용하고 싶지 않은 경우 이렇게 할 수 있습니다. Flutter 모듈에서 코드를 변경할 때마다 flutter build ios-framework 를 실행해야 합니다.

릴리스된 버전에 대해 이 방법을 사용하는 것이 좋습니다.

### 자세한 방법:

Flutter module에서 다음 명령을 실행합니다. 다음 예시에서는 framework를 some/path/MyApp/Flutter/ 에 생성한다고 가정합니다.





flutter build ios-framework --output=some/path/MyApp/Flutter/

생성된 frameworks를 Xcode의 기존 애플리케이션에 임베드하고 연결합니다. 예를 들어

some/path/MyApp/Flutter/Release/ 디렉터리에서 애플리케이션 target 컴파일 설정의 General >

Frameworks, Libraries, and Embedded Content로 frameworks를 드래그한 다음 Embed 드롭다운 목록에서 'Embed & Sign'을 선택할 수 있습니다.

### 추가 모드

기존 애플리케이션에 Flutter Module을 추가하는 것이 좋습니다.

Native 프로젝트에서 Flutter의 Chat 및 Call 모듈을 수행할 Flutter 엔진을 빌드합니다. 자세한 내용은 여기를 참고하십 시오.

Flutter 엔진의 두 가지 모드(단일 FlutterEngine 또는 FlutterEngineGroup이 있는 두 개의 FlutterEngine)가 제공됩니다.

엔진 모드	소개	장점	단점	Demo 소스 코드 다운로 드
단일 Flutter 엔진	Chat과 Call 모두 하나의 FlutterEngine으 로 통합됩니다.	편리하고 모든 Flutter 코드가 균일 하게 유지됩니다.	Call 플러그인으로 인해 전 화가 걸려오면 통화 상태를 표시하기 위해 Flutter 페이 지로 이동해야 하며 이로 인 해 현재 상태가 중단될 수 있 으며 상대적으로 좋지 않은 경험을 할 수 있습니다.	GitHub
멀티 Flutter 엔진	두 개의 개별 FlutterEngine에 있는 Chat 및 Call 모듈이며, Flutter 엔진 그룹은 두 엔진을 통합 관리 하는 데 사용됩니 다.	Call 플러그인은 Flutter 엔진에 독 립적으로 존재하며 독립적인 페이 지에 의해 제어됩니다. 전화가 오면 통화 페이지는 별도로 존재해야 하 며, 통화가 끝나면 현재 상태를 탐 색하지 않고 자동으로 해제하여 더 나은 경험을 제공합니다.	통화 페이지 최소화는 허용 되지 않습니다.	GitHub

또한 IM Native SDK와 Flutter SDK를 모두 통합하는 솔루션도 제공됩니다. 자세한 내용은 여기를 참고하십시오. Demo는 GitHub에서 확인할 수 있습니다.

### 솔루션1: 멀티 FlutterEngine[권장]

이 솔루션에서 Chat 및 Call 모듈은 서로 다른 Flutter 엔진과 독립적입니다.

멀티 Flutter 엔진을 사용할 때의 이점은 각 인스턴스가 독립적이고 자체 내부 탐색 스택, UI 및 애플리케이션 상태를 유지한다는 것입니다. 이는 상태 유지에 대한 전체 애플리케이션 코드의 책임을 단순화하고 모듈성을 향상시킵니다.



Android 및 iOS에 여러 Flutter 엔진을 추가하는 것은 주로 FlutterEngineGroup 클래스(Android API, iOS API)를 기반 으로 하여 여러 FlutterEngine(Flutter 엔진)을 구성하고 관리합니다.

당사 프로젝트에서는 하나의 FlutterEngineGroup이 Chat 및 Calling 모듈을 포함하여 두 개의 FlutterEngine을 관리하 는 데 사용됩니다.

### Flutter Module 개발

Flutter를 기존 애플리케이션에 임베드하려면 먼저 Flutter 모듈을 생성하십시오. 명령줄에서 다음을 실행합니다.





```
cd some/path/
flutter create --template module tencent_chat_module
```

Flutter 모듈 프로젝트는 some/path/tencent\_chat\_module/에 생성됩니다. 해당 디렉터리에서 flutter run -debug 또는 flutter build ios 와 같은 다른 Flutter 프로젝트에서와 동일한 Flutter 명령을 실행할 수 있습니 다. Flutter 및 Dart 플러그인을 사용하여 Android Studio, IntelliJ 또는 VS Code에서 모듈을 실행할 수도 있습니다. 이 프로젝트에는 기존 애플리케이션에 임베드되기 전에 모듈의 단일 보기 예시 버전이 포함되어 있으며, 이는 코드의 Flutter 전용 부분을 점진적으로 테스트하는 데 유용합니다.

tencent\_chat\_module 모듈 디렉터리 구조는 일반 Flutter 애플리케이션과 유사합니다.





이제 lib/ 내에서 코딩할 수 있습니다.



### Flutter lib의 구조

설명 :

다음 구조와 코드는 데모용이며 실제 요구 사항에 맞게 동적으로 수정할 수 있습니다. 필요에 따라 Tencent Cloud IM Flutter를 가져올 수 있습니다.

이제 lib/ 내에 call, chat, common 을 포함하여 세 개의 디렉터리를 생성해 보겠습니다. 통화 엔진, IM 엔진 및 일반적인 model 클래스에 별도로 사용됩니다.



tencent\_chat\_module/ - lib/ └── call/



| └── chat/ | └── common/

### 일반적인 model 클래스 모듈

새 파일 common/common\_model.dart 에 다음 두 class를 추가합니다.네이티브와 Flutter 간의 통신 프록시를 정 의하는 데 사용됩니다.



```
class ChatInfo {
   String? sdkappid;
   String? userSig;
```

```
String? userID;
 ChatInfo.fromJSON(Map<String, dynamic> json) {
    sdkappid = json["sdkappid"].toString();
   userSig = json["userSig"].toString();
   userID = json["userID"].toString();
  }
 Map<String, String> toMap(){
    final Map<String, String> map = {};
   if(sdkappid != null){
      map["sdkappid"] = sdkappid!;
    }
   if(userSig != null) {
      map["userSig"] = userSig!;
    }
   if(userID != null){
      map["userID"] = userID!;
    }
   return map;
  }
}
class CallInfo{
 String? userID;
 String? groupID;
 CallInfo();
 CallInfo.fromJSON(Map<String, dynamic> json) {
   groupID = json["groupID"].toString();
    userID = json["userID"].toString();
  }
 Map<String, String> toMap() {
    final Map<String, String> map = {};
    if(userID != null){
      map["userID"] = userID!;
    }
    if(groupID != null) {
      map["groupID"] = groupID!;
    }
    return map;
  }
}
```

#### Chat 모듈

먼저 IM 엔진을 작성합니다. 이 모듈의 모든 코드와 파일은 1ib/chat 디렉터리에 있습니다.

1. 글로벌 상태 관리 Model로 사용되는 model.dart 파일을 생성합니다.

이 Model은 Tencent Cloud IM Flutter 모듈, 오프라인 회선 푸시 모듈, 글로벌 상태 관리 및 Native 앱과의 통신을 초기 화하고 유지하는 데 사용됩니다.

Chat 모듈의 핵심입니다.

자세한 구현은 Demo의 소스 코드를 참고할 수 있지만 다음 세 가지 기능에 집중하는 것이 좋습니다.

Future \_handleMessage(MethodCall call): 로그인 정보, 클릭 푸시 이벤트 등 Native 앱에서 호출한 메시지를 수신합 니다.

Future handleClickNotification(Map<String, dynamic> msg): Native 패스스루 전송에서 이벤트를 처리하려면 알림을 클릭하고 Map에서 데이터를 가져오고 특정 세션과 같은 해당 서브 모듈로 리디렉션합니다.

Future initChat(): Tencent Cloud IM SDK 및 오프라인 푸시 플러그인을 초기화하고 로그인하고 Token을 업로드합니 다. 이 방법은 동기화 잠금 메커니즘을 사용하여 동시에 하나만 실행할 수 있도록 하고 초기화가 성공한 후에는 반복 적으로 실행되지 않습니다.

### 설명:

Token을 업로드하기 전에 오프라인 푸시를 구성하고 이 문서를 참고하여 이 기능을 사용하십시오.

2. Chat 모듈의 주 게이트로 사용되는 chat\_main.dart 파일을 생성합니다.

또한 Flutter Chat 모듈의 홈 페이지로 사용됩니다.

Demo에서는 로그인하기 전에 로드 상태를 표시한 다음 대화 목록을 표시합니다.

또한 여기에서 'didChangeAppLifecycleState' 모니터링 및 각 포그라운드/백그라운드 전환 시 애플리케이션의 현재 상태를 Tencent Cloud Chat 백엔드에 리포트해야 합니다. 자세한 내용은 문서를 참고하십시오.

자세한 구현은 Demo의 소스 코드를 참고하십시오.

3. 싱글톤 관리 오프라인 푸시 플러그인을 유지 관리하는 데 사용되는 push.dart 파일을 생성합니다. Token 획 득 및 리포트/푸시 권한 획득 등의 작업에 사용됩니다. 자세한 구현은 Demo의 소스 코드를 참고하십시오.

4. TUIKit의 대화 목록 위젯 TIMUIKitConversation을 구현하는 데 사용되는 conversation.dart 파일을생성합니다. 자세한 구현은 Demo의 소스 코드를 참고하십시오.

5. TUIKit의 기록 메시지 목록을 구현하고 메시지 위젯 TIMUIKitChat 을 보내는 데 사용되는 chat.dart 파일 을 생성합니다.

이 페이지는 Profile 및 Group Profile로 리디렉션할 수도 있습니다.

자세한 구현은 Demo의 소스 코드를 참고하십시오.

6. TUIKit의 사용자 프로필 위젯 TIMUIKitProfile 을 구현하는 데 사용되는 user\_profile.dart 파일을 생성합니다. 자세한 구현은 Demo의 소스 코드를 참고하십시오.

7. TUIKit의 그룹 프로필 위젯TIMUIKitGroupProfile을 구현하는 데 사용되는group\_profile.dart파일을 생성합니다. 자세한 구현은 Demo의 소스 코드를 참고하십시오.

이제 다음과 같은 구조로 Chat 모듈이 개발되었습니다.





tence	nt_chat_module/
11	ib/
_	— call/
	└── chat.dart
	└── model.dart
	└── chat_main.dart
	└── push.dart
	└── conversation.dart
	└── user_profile.dart
	└── group_profile.dart
_	— chat/

common/

#### Call 모듈

음성/영상 통화 플러그인에서 제공하는 음성 통화 및 영상 통화에 사용되는 모듈입니다.

이 모듈의 핵심 기능은 걸려오는 전화를 수신할 때 Native에 이 페이지를 보여달라고 요청하는 메소드를 호출하는 것 입니다. 또는 Native를 통해 Chat 모듈에서 요청을 수신하면 다른 사람에게 전화를 걸 수 있습니다.

\*\* 먼저 IM 엔진을 작성하십시오. 다음 코드 및 파일은 1ib/call 디렉터리에 있습니다.\*\*

1. 글로벌 상태 관리 Model로 사용되는 model.dart 파일을 생성합니다.

이 Model은 음성/영상 통화 플러그인의 인스턴스 초기화 및 관리, 전역 상태 관리, Native 앱과의 통신에 사용됩니다. Call 모듈의 핵심입니다.

자세한 구현은 Demo의 소스 코드를 참고할 수 있지만 다음 세 가지 기능에 집중하는 것이 좋습니다.

\_onRtcListener = TUICallingListener(...): 통화 이벤트의 리스너, Method Channel을 통해 Native 레이어에 알리고, Call 모듈이 속한 ViewController(iOS)/Activity(Android)의 프런트 엔드 표시 여부를 동적으로 제어합니다.

Future \_handleMessage(MethodCall call): Native를 통해 Call 모듈에서 요청을 수신할 때 주로 다른 사람에게 통화를 시작하는 데 사용되는 메소드 채널 호출 이벤트의 리스너입니다.

2. Call 모듈의 주 게이트로 사용되는 call\_main.dart 파일을 생성합니다.

시작 호출 페이지에 사용되며 음성/영상 통화 플러그인에 바인딩해야 하는 navigatorKey는 여기에 추가되어야 합니 다.

자세한 구현은 Demo의 소스 코드를 참고하십시오.

### 각 Flutter 엔진의 게이트 구성

상기 세가지 모듈을 개발한 후 이제 FlutterEngine의 게이트로 사용되는 각 모듈의 게이트를 구성할 수 있습니다. 1. 기본 게이트

lib/main.dart 를 열고 빈 MaterialApp을 반환하도록 기본 main() 기능을 수정합니다.

이 메소드는 Flutter Module의 기본 게이트로 사용되며, FlutterEngineGroup을 사용하는 Flutter 멀티 엔진 관리에서 서 브 Flutter Engine이 없고 entry point가 설정되지 않은 경우 이 메소드는 사용되지 않습니다.

예를 들어 이 시나리오에서는 기본 main() 메소드가 사용되지 않습니다.





```
void main() {
  WidgetsFlutterBinding.ensureInitialized();
  runApp(MaterialApp(
    title: 'Flutter Demo',
    theme: ThemeData(
       primarySwatch: Colors.blue,
    ),
    home: Container(),
  ));
}
```

### 2. Chat 모듈 게이트

메소드를 entry-point 로표시하려면 @pragma('vm:entry-point') 주석을 사용합니다. 메소드 이름 chatMain 은 항목의 이름입니다. Native에서 이 이름은 해당 FlutterEngine을 생성하는 데에도 사용됩니다. 전역 ChangeNotifierProvider 상태 관리를 사용하여 ChatInfoModel 데이터 및 비즈니스 로직을 유지 합니다.



```
@pragma('vm:entry-point')
void chatMain() {
    // This call ensures the Flutter binding has been set up before creating the
    // MethodChannel-based model.
```

```
WidgetsFlutterBinding.ensureInitialized();
final model = ChatInfoModel();
runApp(
    ChangeNotifierProvider.value(
      value: model,
      child: const ChatAPP(),
    ),
  );
}
```

3. Call 모듈 게이트

이게이트의 이름은 callMain 으로 지정됩니다.

```
전역 ChangeNotifierProvider 상태관리를 사용하여 CallInfoModel 데이터 및 비즈니스 로직을 유지
합니다.
```





```
@pragma('vm:entry-point')
void callMain() {
    // This call ensures the Flutter binding has been set up before creating the
    // MethodChannel-based model.
    WidgetsFlutterBinding.ensureInitialized();
    final model = CallInfoModel();
    runApp(
        ChangeNotifierProvider.value(
        value: model,
    }
}
```

```
child: const CallAPP(),
    ),
  );
}
```

여기까지 Flutter Module용 Dart 코드가 작성 완료되었습니다. 이제 기존 앱의 Native 통합을 살펴보겠습니다.

### iOS Native 개발

여기에서는 Swift를 예로 듭니다.

설명 :

다음 구조와 코드는 데모용이며 실제 요구 사항에 맞게 효율적으로 구성할 수 있습니다.

XCode 내에서 iOS 프로젝트를 엽니다.

기존 애플리케이션(MyApp)에 아직 Podfile이 없는 경우 CocoaPods 시작하기 가이드에 따라 프로젝트에 Podfile 을 추가하십시오.

### Flutter Module 가져오기

기존 iOS 앱에 Flutter module을 추가하는 부분은 이 부분을 참고하십시오.

iOS 프로젝트에서 FlutterEngine 관리

# Tencent Cloud Chat - Add Flutter To Native AP



FlutterEngineGroup 을 생성하여 여러 엔진 인스턴스를 통합 관리합니다.

AppDelegate.swift 에 다음을 추가합니다.





```
@UIApplicationMain
class AppDelegate: FlutterAppDelegate {
    lazy var flutterEngines = FlutterEngineGroup(name: "chat.flutter.tencent", projec
    ...
}
```

### FlutterEngine을 관리할 싱글톤 객체를 생성합니다.

이 Swift 싱글톤 객체는 한 곳에서 Flutter 인스턴스를 관리하는 데 사용되며 메소드를 호출하기 위해 전체 프로젝트에 메소드를 제공합니다.

Demo의 기본 구현 로직은 Chat의 ViewController에 대한 새 네비게이터를 사용하는 동안 자동으로 Call의 ViewController를 present 및 dismiss하는 것입니다.

새 파일 FlutterUtils.swift 를 생성하고 코딩합니다. 자세한 코드는 Demo 소스 코드를 참고하십시오. 주로 다음에 초점을 맞춥니다:

private override init(): 각 Flutter 엔진 인스턴스를 초기화하고 Method Channel 이벤트를 등록합니다.

func reportChatInfo(): Tencent Cloud IM SDK 초기화 및 로그인을 위해 현재 사용자 로그인 정보 및 SDKAPPID를 Flutter Module에 리포트합니다.

func launchCallFunc(): Call을 시작하는 데 사용되는 Flutter 페이지는 통화 초대를 받는 Call 모듈이나 적극적으로 통 화를 시작하는 Chat 모듈에 의해 트리거될 수 있습니다.

func triggerNotification(msg: String): iOS Native 레이어에서 받은 오프라인 푸시 메시지 클릭 이벤트와 그 안에 포함 된 ext 정보를 Flutter 레이어에 바인딩된 수신 처리 이벤트에 JSON String 형태로 패스스루합니다. 예를 들어 해당 세 션으로의 오프라인 푸시 클릭 리디렉션을 처리하는 데 사용됩니다.

### 오프라인 푸시 클릭 이벤트 수신 및 포워딩

오프라인 푸시 초기화/Token 업로드/클릭 이벤트에 따른 세션 리디렉션 처리 등은 Flutter Chat 모듈에서 모두 이루어 졌기 때문에 Native 영역에서는 클릭 알림 이벤트의 ext만 전달하면 됩니다.

이렇게 해야 하는 이유는 클릭 알람 이벤트가 Native에서 소비되었기 때문에 Flutter 레이어에서 직접 가져올 수 없으며 Native를 통해 포워딩해야 하기 때문입니다.

AppDelegate.swift 파일에 다음 코드를 추가합니다. 특정 코드는 Demo 소스 코드를 참고하십시오.

```
20
  21 @UIApplicationMain
22 class AppDelegate: FlutterAppDelegate {
  23
        lazy var flutterEngines = FlutterEngineGroup(name: "chat.flutter.tencent", project: nil)
  24
        override func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplicat
  25
  26
            if #available(iOS 10.0, *) {
28
              UNUserNotificationCenter.current().delegate = self
            }
  29
  30
  31
            if let remoteNotification = launchOptions?[UIApplication.LaunchOptionsKey.remoteNotification]{
  32
                let notificationExt: [AnyHashable:Any] = remoteNotification as! [AnyHashable:Any]
                let remoteNotificationString: String = notificationExt.jsonStringRepresentaiton ?? "{}"
   34
                FlutterUtils.shared.triggerNotification(msg: remoteNotificationString)
  35
  36
  37
          return super.application(application, didFinishLaunchingWithOptions: launchOptions);
  38
        3
  39
          override func userNotificationCenter(_ center: UNUserNotificationCenter, didReceive response: UNNotificationRe
  40
              @escaping () -> Void) {
 41
              let notificationExt: Dictionary = response.notification.request.content.userInfo
  42
              let notificationExtString: String = notificationExt.jsonStringRepresentaiton ?? "{}"
              FlutterUtils.shared.triggerNotification(msg: notificationExtString)
   45
   46 }
```

이제 iOS Native 레이어 작성을 완료했습니다.

### Android Native 개발

여기서는 Kotlin을 예로 듭니다.

### 설명 :

다음 구조와 코드는 데모용이며 실제 요구 사항에 맞게 효율적으로 구성할 수 있습니다.

### Flutter Module 가져오기

기존 Android 앱에 Flutter module을 추가하는 부분은 이 부분을 참고하십시오. 방법2를 권장합니다.

### Android 프로젝트에서 FlutterEngine 관리

### FlutterEngine을 관리할 싱글톤 객체를 생성합니다.

이 Kotlin 싱글톤 객체는 한 곳에서 Flutter 인스턴스를 관리하는 데 사용되며 메소드를 호출하기 위해 전체 프로젝트 에 메소드를 제공합니다.

새파일 FlutterUtils.kt 를생성하고 정적 객체 FlutterUtils 를 정의합니다.





```
@SuppressLint("StaticFieldLeak")
object FlutterUtils {}
```

### FlutterEngineGroup 을 생성하여 여러 엔진 인스턴스를 통합 관리합니다.

FlutterUtils.kt 파일에서 FlutterEngineGroup 을 정의하고, 각 FlutterEngine 인스턴스와 MethodChannel을 지원하며 초기화 시 초기화합니다.





```
lateinit var context : Context
lateinit var flutterEngines: FlutterEngineGroup
private lateinit var chatFlutterEngine:FlutterEngine
private lateinit var callFlutterEngine:FlutterEngine
```

lateinit var chatMethodChannel: MethodChannel
lateinit var callMethodChannel: MethodChannel

```
// 초기화
flutterEngines = FlutterEngineGroup(context)
```
#### Flutter 엔진을 관리하는 싱글톤 객체를 계속 완료합니다.

Demo의 기본 구현 로직은 Chat과 Call 모두의 Activity에 대한 새 네비게이터를 사용하는 것입니다.

Chat의 Activity는 사용자가 들어가고 나가는 반면, Call의 Activity는 자동으로 들어가고 나가며, 리스너에 의해 트리거 되거나 수동으로 전화를 겁니다.

주로 다음에 초점을 맞춥니다:

fun init(): 각 Flutter 엔진 인스턴스를 초기화하고 Method Channel을 등록하고 이벤트를 수신합니다.

fun reportChatInfo(): Flutter 레이어를 초기화하고 Tencent Cloud IM에 로그인할 수 있도록 사용자 로그인 정보 및 SDKAPPID를 Flutter Module에 패스스루합니다.

fun launchCallFunc(): Call을 시작하는 데 사용되는 Flutter 페이지는 통화 초대를 받는 Call 모듈이나 통화를 시작하는 Chat 모듈에 의해 트리거될 수 있습니다.

fun triggerNotification(msg: String): iOS Native 레이어에서 받은 오프라인 푸시 메시지 클릭 이벤트와 그 안에 포함된 ext 정보를 Flutter 레이어에 바인딩된 수신 처리 이벤트에 JSON String 형태로 패스스루합니다. 예를 들어 해당 세션 으로의 오프라인 푸시 클릭 리디렉션을 처리하는 데 사용됩니다.

이 싱글톤 object에 대한 Demo 소스 코드를 참고하십시오.

#### 주 게이트 MyApplication 에서 위의 객체를 초기화합니다.

MyApplication.kt 파일에서 전역 context를 싱글톤 객체로 전송하고 초기화합니다.





```
class MyApplication : MultiDexApplication() {
    override fun onCreate() {
        super.onCreate()
        FlutterUtils.context = this // new
        FlutterUtils.init() // new
    }
}
```

# 오프라인 푸시 클릭 이벤트 수신 및 포워딩

오프라인 푸시 초기화/Token 업로드/클릭 이벤트에 따른 세션 리디렉션 처리 등은 Flutter Chat 모듈에서 모두 이루어 졌기 때문에 Native 영역에서는 클릭 알림 이벤트의 ext만 전달하면 됩니다.

이렇게 해야 하는 이유는 클릭 알람 이벤트가 Native에서 소비되었기 때문에 Flutter 레이어에서 직접 가져올 수 없으며 Native를 통해 포워딩해야 하기 때문입니다.

#### 주의:

다양한 벤더 간의 오프라인 푸시 액세스 단계는 일치하지 않습니다. 본문에서는 OPPO를 예로 들어 설명하였습니다. 전체 벤더의 액세스 솔루션은 이 문서를 참고하십시오.

Tencent Cloud IM 콘솔에 OPPO의 푸시 인증서를 추가하고 '후속 작업 클릭', 앱에서 지정된 페이지 열기 를 선택하면인앱 페이지는 오프라인 푸시 정보를 처리하기 위한 페이지를Activity로 구성하며, 앱의 홈 페이지로설정하는 것이 좋습니다. 예를 들어 Demo 구성은com.tencent.chat.android.MainActivity입니다.

	Push Platform	Mi Huawei Google	🔵 Meizu 🔷 Vivo 🔇	
	АррКеу *	2423525	How to generate an OPP	
	AppID *	2352323		
s sen	MasterSecret *	asdxgasdgasdgad		
	ChannellD	Enter a channel ID		
	Response after Click	Open application Open we	ebpage Open specifie	
	Specified In-app	activity <ul> <li>com.tencent.chat.android.MainAc</li> <li>Redirect to specified in-app page through push component</li> <li>Component</li> <li>Component</li></ul>		
	Page *			
3?		C	onfirm	

상기 콘솔에 대해 설정된 오프라인 푸시를 위한 Activity 파일에 다음 코드를 추가합니다.

이 코드의 기능은 벤더가 해당 Activity를 가져올 때 Bundle에서 HashMap 형식의 ext 정보를 꺼내고 싱글톤 객체의 메 소드를 트리거하고 이 정보를 Flutter에 수동으로 포워딩할 수 있습니다. 이 기능에 대한 Demo 소스 코드를 참고하십 시오. 4

}



이제 Android Native 레이어 작성을 완료했습니다.

if(!keyMap.isEmpty()){

# 솔루션2: 단일 FlutterEngine

이 솔루션에서 Chat 모듈과 Call 모듈은 단일 Flutter 엔진 인스턴스에 포함됩니다.

FlutterUtils.triggerNotification(gson.toJson(keyMap))



결과적으로 이러한 모듈은 동시에 표시하거나 숨길 수만 있으며 하나의 Flutter 엔진만 점검하면 됩니다.

## Flutter Module 개발

Flutter를 기존 애플리케이션에 임베드하려면 먼저 Flutter 모듈을 생성하십시오. 명령줄에서 다음을 실행합니다.





```
cd some/path/
flutter create --template module tencent_chat_module
```

Flutter 모듈 프로젝트는 some/path/tencent\_chat\_module/에 생성됩니다. 해당 디렉터리에서 flutter run -debug 또는 flutter build ios 와 같은 다른 Flutter 프로젝트에서와 동일한 Flutter 명령을 실행할 수 있습니 다. Flutter 및 Dart 플러그인을 사용하여 Android Studio, IntelliJ 또는 VS Code에서 모듈을 실행할 수도 있습니다. 이 프로젝트에는 기존 애플리케이션에 임베드되기 전에 모듈의 단일 보기 예제 버전이 포함되어 있으며, 이는 코드의 Flutter 전용 부분을 점진적으로 테스트하는 데 유용합니다.

tencent\_chat\_module 모듈 디렉터리 구조는 일반 Flutter 애플리케이션과 유사합니다.





이제 lib/ 내에서 코딩할 수 있습니다.

#### main.dart

main.dart 를 수정하여 TUIKi, 오프라인 푸시 플러그인 및 통화 플러그인을 통합합니다.

글로벌 상태인 IM SDK 및 Method Channel과 Native 통신 상태는 Chat InfoModel 에서 관리됩니다.

Native로부터 로그인 사용자 정보 및 SDKAPPID를 받은 후 \_coreInstance.init(),

\_coreInstance.login() 을 호출하여 Tencent Cloud IM SDK를 초기화하고 로그인합니다. 또한 오디오/비디 오 푸시 플러그인과 오프라인 푸시 플러그인도 초기화하고 푸시 Token 리포트를 완료합니다.

#### 설명 :

Token을 업로드하기 전에 벤더의 오프라인 푸시를 구성하고 이 오프라인 푸시 엑세스 가이드를 참고하여 이 기능을 사용하십시오.

음성/영상 통화 플러그인에 대한 팁:

통화 초대를 수신하고 Native 메소드를 호출하여 사용자가 현재 이 Flutter 모듈 페이지에 있는지 여부를 Native가 점 검하도록 합니다. 현재 위치에 있지 않은 경우 수신 전화 페이지를 표시하려면 프런트 엔드 페이지를 이 모듈로 강제 조정해야 합니다.

오프라인 푸시 플러그인에 대한 팁:

알림을 클릭하여 Native 패스스루 전송에서 이벤트를 처리하고, Map에서 데이터를 가져오고, 특정 세션과 같은 해당 서브 모듈로 이동합니다.

홈페이지 제작을 완료하고 로그인하기 전에 로딩 상태를 표시하고, 로그인 후 대화 목록 페이지를 표시합니다.

또한 여기에서 didChangeAppLifecycleState 수신 및 각 포그라운드/백그라운드 전환 시 애플리케이션의 현 재 상태를 리포트해야 합니다. 자세한 내용은 오프라인 푸시 플러그인 문서 5단계를 참고하십시오.

자세한 구현은 Demo의 소스 코드를 참고하십시오.

#### TUIKit의 다른 모듈 가져오기

1. 싱글톤 관리 오프라인 푸시 플러그인을 유지 관리하는 데 사용되는 push.dart 파일을 생성합니다. Token 획 득 및 리포트/푸시 권한 획득 등의 작업에 사용됩니다. 자세한 구현은 Demo의 소스 코드를 참고하십시오.

2. TUIKit의 대화 목록 위젯TIMUIKitConversation을 구현하는 데 사용되는conversation.dart파일을생성합니다. 자세한 구현은 Demo의 소스 코드를 참고하십시오.

3. TUIKit의 기록 메시지 목록을 구현하고 메시지 위젯 TIMUIKitChat 을 보내는 데 사용되는 chat.dart 파일을 생성합니다.

이 페이지는 Profile 및 Group Profile로 리디렉션할 수도 있습니다.

자세한 구현은 Demo의 소스 코드를 참고하십시오.

4. TUIKit의 사용자 프로필 위젯 TIMUIKitProfile을 구현하는 데 사용되는 user\_profile.dart파일을생성합니다. 자세한 구현은 Demo의 소스 코드를 참고하십시오.

5. TUIKit의 그룹 프로필 위젯 TIMUIKitGroupProfile 을 구현하는 데 사용되는 group\_profile.dart 파 일을 생성합니다. 자세한 구현은 Demo의 소스 코드를 참고하십시오.

이제 Flutter Module이 개발되었습니다.

## iOS Native 개발

©2013-2022 Tencent Cloud. All rights reserved.

여기에서는 Swift를 예로 들어 Objective-C도 사용할 수 있습니다.

#### 설명 :

다음 구조와 코드는 데모용이며 실제 요구 사항에 맞게 효율적으로 구성할 수 있습니다.

XCode 내에서 iOS 프로젝트를 엽니다.

기존 애플리케이션(MyApp)에 아직 Podfile이 없는 경우 CocoaPods 시작하기 가이드에 따라 프로젝트에 Podfile 을 추가하십시오.

## Flutter Module 가져오기

기존 iOS 앱에 Flutter module을 추가하는 부분은 이 부분을 참고하십시오.

## iOS 프로젝트에서 FlutterEngine 관리

#### FlutterEngine을 생성합니다.

FlutterEngine을 생성하는 적절한 위치는 호스트 앱에 따라 다릅니다. 예를 들어 AppDelegate 에서 app 시작 시 속성으로 노출되는 FlutterEngine 생성을 시연합니다.





```
import UIKit
import Flutter
import FlutterPluginRegistrant
@UIApplicationMain
class AppDelegate: FlutterAppDelegate { // More on the FlutterAppDelegate.
lazy var flutterEngine = FlutterEngine(name: "tencent cloud chat")
override func application(_ application: UIApplication, didFinishLaunchingWithOpt
    // Runs the default Dart entrypoint with a default Flutter route.
    flutterEngine.run();
```

```
GeneratedPluginRegistrant.register(with: self.flutterEngine);
return super.application(application, didFinishLaunchingWithOptions: launchOpti
}
```

# FlutterEngine을 관리할 싱글톤 객체를 생성합니다.

이 Swift 싱글톤 객체는 한 곳에서 Flutter Method Channel을 관리하는 데 사용되며 Flutter Module 통신 메소드를 호출 하기 위해 전체 프로젝트에 메소드를 제공합니다.

주로 다음에 초점을 맞춥니다:

private override init(): 각 Flutter 인스턴스를 초기화하고 Method Channel 이벤트를 등록합니다.

func reportChatInfo(): Tencent Cloud IM SDK 초기화 및 로그인을 위해 현재 사용자 로그인 정보 및 SDKAPPID를 Flutter Module에 리포트합니다.

func launchChatFunc(): Flutter Module용 ViewController를 표시합니다.

func triggerNotification(msg: String): iOS Native 레이어에서 받은 오프라인 푸시 메시지 클릭 이벤트와 그 안에 포함 된 ext 정보를 Flutter 레이어에 바인딩된 수신 처리 이벤트에 JSON String 형태로 패스스루합니다. 예를 들어 해당 세 션으로의 오프라인 푸시 클릭 리디렉션을 처리하는 데 사용됩니다.

자세한 구현은 Demo의 소스 코드를 참고하십시오.

# 오프라인 푸시 클릭 이벤트 수신 및 포워딩

오프라인 푸시 초기화/Token 업로드/클릭 이벤트에 따른 세션 리디렉션 처리 등은 Flutter Chat 모듈에서 모두 이루어 졌기 때문에 Native 영역에서는 클릭 알림 이벤트의 ext만 전달하면 됩니다.

이렇게 해야 하는 이유는 클릭 알람 이벤트가 Native에서 소비되었기 때문에 Flutter 레이어에서 직접 가져올 수 없으 며 Native를 통해 포워딩해야 하기 때문입니다.

AppDelegate.swift 파일에 다음 코드를 추가합니다. 특정 코드는 Demo 소스 코드를 참고하십시오.



이제 iOS Native 레이어 작성을 완료했습니다.

## Android Native 개발

여기서는 Kotlin을 예로 들지만 Java도 사용할 수 있습니다.

#### 설명 :

다음 구조와 코드는 데모용이며 실제 요구 사항에 맞게 효율적으로 구성할 수 있습니다.

#### Flutter Module 가져오기

기존 Android 앱에 Flutter module을 추가하는 부분은 이 부분을 참고하십시오. 방법2를 권장합니다.

#### Android 프로젝트에서 FlutterEngine 관리

#### FlutterEngine을 관리할 싱글톤 객체를 생성합니다.

이 Kotlin 싱글톤 객체는 한 곳에서 Flutter Method Channel을 관리하는 데 사용되며 Flutter Module 통신 메소드를 호 출하기 위해 전체 프로젝트에 메소드를 제공합니다.

새파일 FlutterUtils.kt 를 생성하고 정적 객체 FlutterUtils 를 정의합니다.





```
@SuppressLint("StaticFieldLeak")
object FlutterUtils {}
```

# FlutterEngine (Flutter 엔진)을 생성합니다.

FlutterUtils.kt 파일에서 FlutterEngine 을정의하고 초기화 시초기화합니다.





lateinit var context : Context
private lateinit var flutterEngine:FlutterEngine

#### // 초기화

flutterEngine = FlutterEngine(context)

#### Flutter 엔진을 관리하는 싱글톤 객체를 위해 추가로 개발되었습니다.

Demo의 기본 구현 로직은 Chat과 Call 모두의 Activity에 대한 새 네비게이터를 사용하는 것입니다. Chat의 Activity는 사용자가 들어가고 나가는 반면, Call의 Activity는 자동으로 들어가고 나가며, 리스너에 의해 트리거 되거나 수동으로 전화를 겁니다. 주로 다음에 초점을 맞춥니다:

fun init(): 각 Method Channel을 초기화하고 메소드 채널 이벤트를 등록합니다.

fun reportChatInfo(): Flutter 레이어를 초기화하고 Tencent Cloud IM에 로그인할 수 있도록 사용자 로그인 정보 및 SDKAPPID를 Flutter Module에 패스스루합니다.

fun launchChatFunc(): Flutter Module용 ViewController를 표시합니다.

fun triggerNotification(msg: String): iOS Native 레이어에서 받은 오프라인 푸시 메시지 클릭 이벤트와 그 안에 포함된 ext 정보를 Flutter 레이어에 바인딩된 수신 처리 이벤트에 JSON String 형태로 패스스루합니다. 예를 들어 해당 세션 으로의 오프라인 푸시 클릭 리디렉션을 처리하는 데 사용됩니다.

이 싱글톤 object에 대한 Demo 소스 코드를 참고하십시오.

#### 주 게이트 MyApplication 에서 상기 객체를 초기화합니다.

MyApplication.kt 파일에서 전역 context를 싱글톤 객체로 전송하고 초기화합니다.





```
class MyApplication : MultiDexApplication() {
    override fun onCreate() {
        super.onCreate()
        FlutterUtils.context = this // new
        FlutterUtils.init() // new
    }
}
```

오프라인 푸시 클릭 이벤트 수신 및 포워딩

오프라인 푸시 초기화/Token 업로드/클릭 이벤트에 따른 세션 리디렉션 처리 등은 Flutter Chat 모듈에서 모두 이루어 졌기 때문에 Native 영역에서는 클릭 알림 이벤트의 ext만 전달하면 됩니다.

이렇게 해야 하는 이유는 클릭 알람 이벤트가 Native에서 소비되었기 때문에 Flutter 레이어에서 직접 가져올 수 없으며 Native를 통해 포워딩해야 하기 때문입니다.

#### 주의:

다양한 벤더 간의 오프라인 푸시 액세스 단계가 일치하지 않기 때문에, 본문에서는 OPPO를 예로 들 뿐입니다. 전체 벤더의 액세스 솔루션은 이 문서를 참고하십시오.

Tencent Cloud IM 콘솔에 OPPO의 푸시 인증서를 추가하고 '후속 작업 클릭', 앱에서 지정된 페이지 열기 를 선택하면인앱 페이지는 오프라인 푸시 정보를 처리하기 위한 페이지를Activity로 구성하며, 앱의 홈 페이지로설정하는 것이 좋습니다. 예를 들어 Demo 구성은com.tencent.chat.android.MainActivity입니다.

	Push Platform	Mi Huawei Google	🔵 Meizu 🔷 Vivo 🔇	
	АррКеу *	2423525	How to generate an OPP	
	AppID *	2352323		
s sen	MasterSecret *	asdxgasdgasdgad		
	ChannellD	Enter a channel ID		
	Response after Click	Open application Open we	ebpage Open specifie	
	Specified In-app	activity <ul> <li>com.tencent.chat.android.MainAc</li> <li>Redirect to specified in-app page through push component</li> <li>Component</li> <li>Component</li></ul>		
	Page *			
3?		C	onfirm	

상기 콘솔에 대해 설정된 오프라인 푸시를 위한 Activity 파일에 다음 코드를 추가합니다.

이 코드의 기능은 벤더가 해당 Activity를 가져올 때 Bundle에서 HashMap 형식의 ext 정보를 꺼내고 싱글톤 객체의 메 소드를 트리거하고 이 정보를 Flutter에 수동으로 포워딩할 수 있습니다. 이 기능에 대한 Demo 소스 코드를 참고하십 시오.



이제 Android Native 레이어 작성을 완료했습니다.

# 추가 솔루션: Native 레이어에서 Tencent Cloud IM 초기화 및 로그인

경우에 따라 복잡한 Chat 및 Call 페이지 없이 기존 UI에 Chat 및 Call 모듈을 통합하는 것을 선호할 수 있습니다. 예를 들어 게임이 있고 전체 화면 채팅 페이지로 이동하지 않고도 플레이어가 경기 중에 서로 채팅할 수 있기를 바랍 니다.

즉, 사용자가 채팅 페이지로 전환하기 전에 복잡한 Flutter Module을 시작하고 싶지 않을 수 있지만 사용자가 여전히 작은 Chat 모듈에서 직접 채팅할 수 있기를 바랄 수 있습니다. 이 경우 Native SDK로 Tencent Cloud IM을 초기화 및 로그인 메소드를 호출할 수 있으며, Tencent Cloud IM Native SDK를 직접 사용하여 필요한 빈도가 높고 간단한 시나리오에서 In-App Chat 기능을 구축할 수 있습니다. 설명:

그러나 필요에 따라 Flutter 내에서 초기화하고 로그인하도록 선택할 수도 있습니다. 이 프로세스는 어디서 실행하든 한 번만 실행해야 합니다.

Flutter SDK를 사용하면 Native SDK를 통합할 수 있으므로 수동으로 Native SDK를 가져올 필요가 없습니다.

# Native 초기화 및 로그인

iOS Swift 코드를 예로 들어 Native 레이어에서 초기화하고 로그인하는 방법을 보여줍니다.





```
userSig: chatInfo.userSig,
succ: {
    self.isLoginSuccess = true
    self.reportChatInfo()
    },
    fail: onLoginFailed()
    }
}
```

그 다음 Native SDK에서 제공하는 API를 사용하여 채팅 모듈을 기존 UI 페이지에 수동으로 구현할 수 있습니다. 자세 한 내용은 iOS 시작하기 또는 Android 시작하기를 참고하십시오.

# Flutter TUIKit 초기화

현재 사용자 정보는 \_coreInstance.setDataFromNative() 를 호출하여 Native 레이어에서 초기화 및 로그 인한 후 Flutter TUIKit에 제공되어야 합니다.





final CoreServicesImpl \_coreInstance = TIMUIKitCore.getInstance();
\_coreInstance.setDataFromNative(userId: chatInfo?.userID ?? "");

이 모듈의 Demo 소스 코드를 참고하십시오.



# Get source code from

# You can refer to our Demo source code, to implem

-----

이제 Tencent Cloud IM Flutter를 기존 애플리케이션에 추가하는 방법에 대한 소개가 끝났습니다.

본문에서 제공된 솔루션을 기반으로 Flutter SDK와 동일한 Flutter 코드 세트를 사용하여 기존 네이티브 Android/iOS APP 개발에 Chat 및 Call 모듈 기능을 빠르게 추가할 수 있습니다.

여전히 궁금한 점이 있으시면 언제든지 문의해 주십시오.

Telegram Group WhatsApp Group

# Reference

- 1. Integrate a Flutter module into your Android project.
- 2. Integrate a Flutter module into your iOS project.
- 3. Adding a Flutter screen to an iOS app.
- 4. Multiple Flutter screens or views.