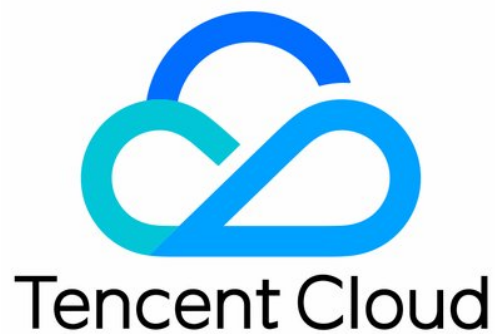


Tencent Real-Time Communication Video Calling (Including UI)

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



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.

목록:

- Video Calling (Including UI)
 - 개요(TUICallKit)
 - 통합(TUICallKit)
 - Android
 - iOS
 - Web
 - UI 사용자 정의(TUICallKit)
 - Android
 - iOS
 - Web
 - 오프라인 푸시(TUICallKit)
 - Android
 - 온클라우드 녹화(TUICallKit)
 - Client APIs (TUICallKit)
 - Android
 - API 개요
 - TUICallKit
 - TUICallEngine
 - TUICallObserver
 - iOS
 - API 개요
 - TUICallKit
 - TUICallEngine
 - TUICallObserver
 - Web
 - API 개요
 - TUICallKit
 - TUICallEngine
 - TUICallEvent
 - 릴리스 노트(TUICallKit)
 - Web
 - Android & iOS

Video Calling (Including UI) 개요(TUICallKit)

최종 업데이트 날짜: : 2024-06-20 17:24:42

컴포넌트 개요

TUICallKit은 음성/영상 통화를 위한 UI 컴포넌트입니다. 이를 통합하고 몇 줄의 간단한 코드를 작성하면 App이 오디오/비디오 통화 및 오프라인 푸시를 지원할 수 있으므로 애플리케이션이 오프라인일 때도 사용자가 전화를 받을 수 있습니다. Android, iOS 및 Web 플랫폼을 지원하며 다음과 같은 기본 기능을 제공합니다.



주의사항 :

2022년 8월 TUICalling은 개발자들의 피드백과 시장 조사를 바탕으로 TUICallKit으로 정식 업그레이드 되었습니다. 새로운 컴포넌트 버전에는 그룹 통화 및 오프라인 푸시와 같은 기능이 업그레이드 되었으며, 더 낮은 비용으로 음성/영상 통화 SDK 플랜을 제공하고, 60일 무료 평가판도 제공합니다.

강점

손쉬운 통합: TUICallKit은 개발 시간의 90%를 절약하고 음성/영상 통화 응용 프로그램을 빠르게 출시할 수 있도록 UI와 함께 오픈 소스 컴포넌트를 제공합니다.

플랫폼 상호 연결: 다양한 플랫폼을 위한 TUICallKit 컴포넌트는 서로 전화를 걸고, 받고, 끊고, 연결할 수 있습니다.

그룹 통화: TUICallKit은 일대일 영상 통화 외에도 여러 그룹 구성원과의 그룹 영상 통화를 지원합니다.

오프라인 푸시: TUICallKit은 Android&iOS에서 오프라인 푸시를 지원하므로 수신자는 App이 오프라인일 때도 새로운 수신 전화 알림을 받을 수 있습니다.

적용 시나리오

TUICallKit은 게임 내 커뮤니케이션, 온라인 고객 서비스, 온라인 의료 상담, 보험 상담 등 다양한 시나리오에서 2명 이상의 음성/영상 통화에 이상적입니다.

TUICallKit 다운로드

오픈 소스 구축

업그레이드 이전의 TUICalling 오픈 소스 프로젝트는 [여기](#)에서 찾을 수 있습니다.

통합(TUICallKit)

Android

최종 업데이트 날짜: : 2022-11-23 15:48:06

본문에서는 TUICallKit 컴포넌트의 액세스를 최단 시간에 완료하는 방법에 대해 설명합니다. 본 문서를 따르면, 다음 몇 가지 주요 단계를 한 시간 내에 완료하고 최종적으로 UI 인터페이스로 영상 통화 기능을 얻을 수 있습니다.

환경 준비

Android 4.1(SDK API Level 16)까지 호환됩니다. Android 5.0(SDK API Level 21) 이후 버전 사용을 권장합니다.

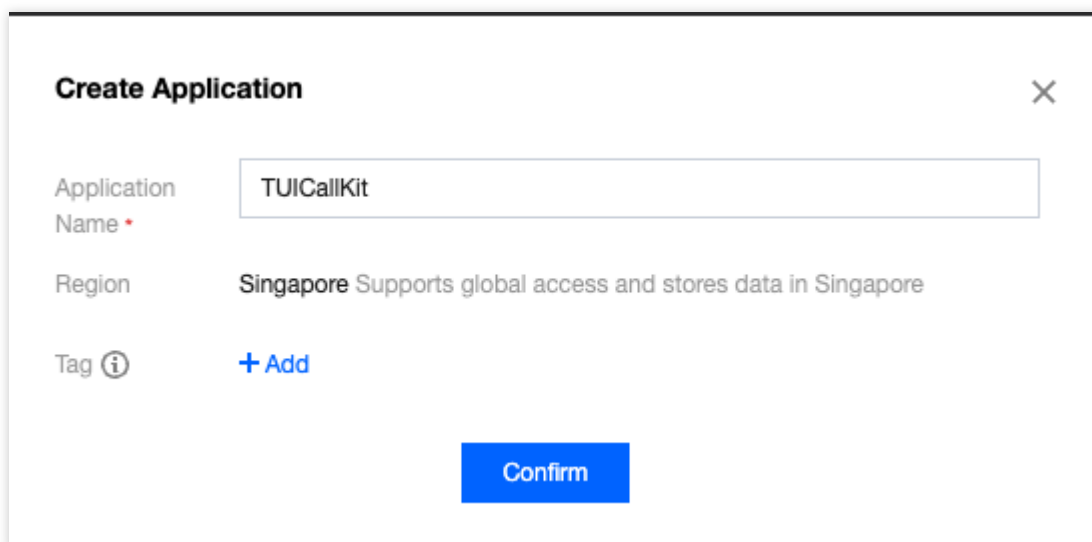
Android Studio 3.5 이상 버전(Gradle 3.5.4 이상 버전).

Android 4.1 이상이 설치된 기기.

1단계: 서비스 활성화

TUICallKit은 Tencent Cloud의 두 가지 유료 PaaS 서비스 [Instant Messaging\(IM\)](#) 및 [Tencent Real-Time Communication\(TRTC\)](#)을 기반으로 하는 오디오/비디오 통신 컴포넌트입니다. 아래 단계에 따라 관련 서비스를 활성화하고 60일 무료 베타 서비스를 체험할 수 있습니다.

1. [IM 콘솔](#)에 로그인하고 **애플리케이션 생성**을 클릭하고 팝업 대화 상자에 애플리케이션 이름을 입력하고 **확인**을 클릭합니다.



Create Application [X]

Application Name

Region

Tag ⓘ

2. 방금 생성한 애플리케이션을 클릭하여 기본 구성 페이지로 이동하고 TUICallKit의 7일 무료 평가판을 보려면 페이지 오른쪽 하단 모서리에 있는 TRTC 활성화에서 무료 평가판을 클릭하십시오. 애플리케이션을 공식 릴리스하려면 구매를 클릭하여 구매 페이지로 이동합니다.

Tencent Real-Time Communication [View application](#)

1. To facilitate TRTC integration in the current IM application, we have automatically created a TRTC application with the same SDKAppID as the current IM application in the [TRTC Console](#). Accounts and authentication for both can be reused.
2. Tencent Real-Time Communication (TRTC) is an independent Tencent Cloud service. For billing details, please see [Price](#).

TRTC enables you to implement audio/video call, group audio chat, video conference and other audio/video features in the current IM application.

Audio/video call: Not activated [Try now](#)

주의 :

다양한 비즈니스 요구 사항에 따라 IM 음성/영상 통화 기능의 다양한 유료 버전을 사용할 수 있습니다. IM 구매 페이지에서 사용 가능한 기능에 대해 자세히 알아보고 원하는 버전을 구매할 수 있습니다.

3. 같은 페이지에서 **SDKAppID**와 ****키(SecretKey)****를 찾아 기록해 둡니다. **4단계: TUI 컴포넌트 로그인**에서 사용될 예정입니다.

Standard Billing Plan

Status **In use**


Plan **Trial**

Expiration time -

[Upgrade](#) [More](#) ▼

App Information

[Edit](#)


SDKAppID 20[REDACTED] 

Application Name TUICallKit

Application Type Video

Application Introduction -

Basic info

Key 8fca0d57d09[REDACTED] 

[Hide key](#)

Key information is sensitive. Keep it confidential and do not disclose it.

Creation time 2022-11-02

Last Modified 2022-11-02

설명 :

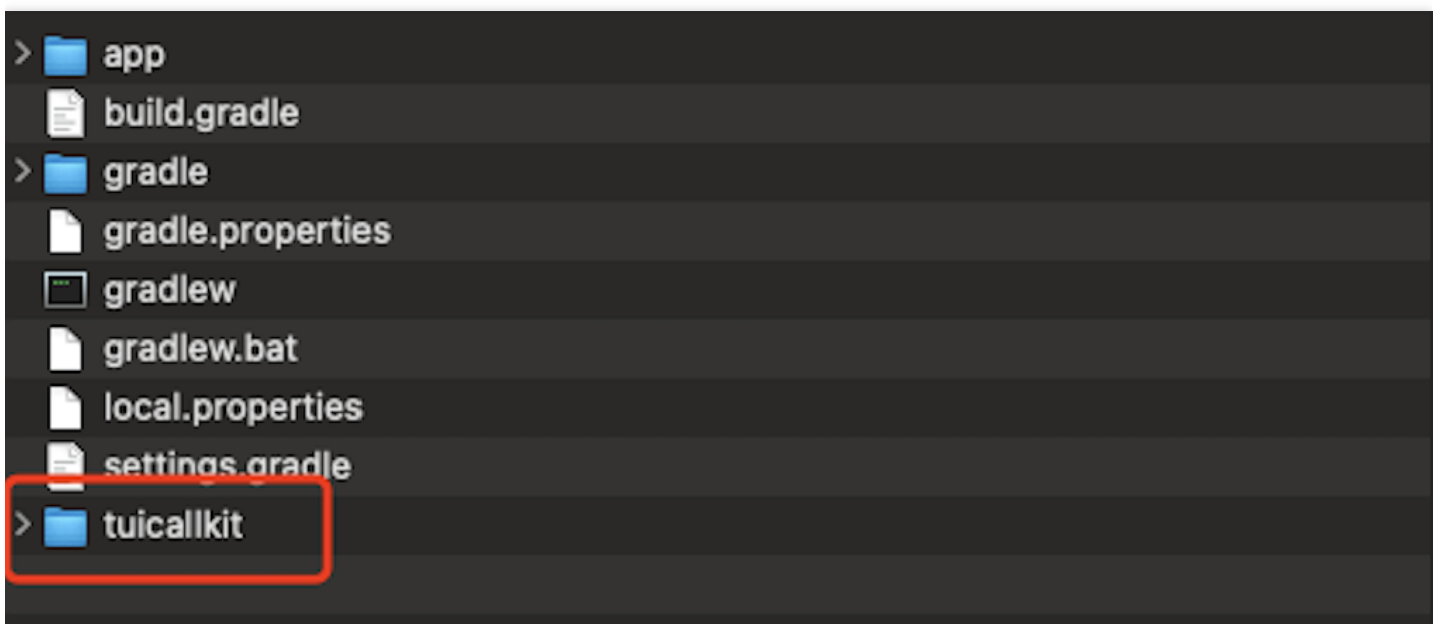
참고: 무료 체험을 클릭하면 이전에 **TRTC** 서비스를 사용한 적이 있는 일부 사용자에게 다음 메시지가 표시됩니다.

```
[-100013]:TRTC service is suspended. Please check if the package balance is 0 or the Tencent Cloud account is in arrears
```


새로운 IM 음성 및 영상 통화 기능은 **TRTC** 및 **IM**의 두 가지 기본 PaaS 서비스가 통합되어 있으므로, **TRTC**의 무료 할당량(10000분)이 만료되거나 소진되면 이 서비스를 활성화할 수 없습니다. 예시 이미지와 같이 **TRTC 콘솔**을 클릭하여 SDKAppID에 해당하는 애플리케이션 관리 페이지에서 후불 기능을 활성화한 후 다시 **애플리케이션을 활성화**하면 음성 및 영상 통화 기능을 정상적으로 체험할 수 있습니다.

2단계: 컴포넌트 다운로드 및 가져오기

[Github](#)에서 코드를 복제/다운로드한 다음 아래와 같이 Android 디렉터리 아래의 tuicallkit 서브 디렉터리를 현재 프로젝트의 app과 동일한 수준 디렉터리로 복사합니다.



3단계: 프로젝트 구성 완료

1. 프로젝트의 루트 디렉터리에서 'setting.gradle' 파일을 찾고 다음 코드를 추가합니다. 이 파일은 **2단계**에서 다운로드한 tuicallkit 컴포넌트를 현재 프로젝트로 가져올 수 있습니다.

```
include ':tuicallkit'
```

2. app 디렉터리에서 'build.gradle' 파일을 찾아 다음 코드를 추가합니다. 이 파일은 현재 새로 추가된 tuicallkit 컴포넌트에 대한 현재 app의 종속을 선언합니다.

```
api project(':tuicallkit')
```

설명 :

TUICallKit 프로젝트에는 기본 종속성인 TRTC SDK , IM SDK , tuicallengine 및 공용 라이브러리 tuicore 가 있으며 개발자가 별도로 구성할 필요가 없습니다. 버전을 업그레이드하려면 tuicallkit/build.gradle 파일을 수정하기만 하면 됩니다.

3. SDK 내에서 Java의 리플렉션 특성을 사용하기 때문에 SDK의 일부 클래스를 난독화 방지 목록에 추가해야 하므로 proguard-rules.pro 파일에 다음 코드를 추가해야 합니다.

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

주의 :

TUICallKit은 내부적으로 카메라, 마이크, 읽기 저장 권한 등을 동적으로 신청할 수 있도록 도와드립니다. 업무 문제로 인해 삭제해야 하는 경우 tuicallkit/src/main/AndroidManifest.xml 을 수정합니다.

4단계: TUI 컴포넌트 로그인

프로젝트에 다음 코드를 추가하는 것은 TUICore에서 관련 인터페이스를 호출하여 TUI 컴포넌트의 로그인을 완료하는 것입니다. TUICallKit의 다양한 기능은 로그인이 성공한 후에야 정상적으로 사용할 수 있기 때문에 이 단계는 매우 중요합니다. 따라서 관련 매개변수가 올바르게 구성되었는지 확인하십시오.

```
//로그인 결과에 대한 리스너 설정
private final TUILoginListener mLoginListener = new TUILoginListener() {
    @Override
    public void onKickedOffline() {
        super.onKickedOffline();
        Log.i(TAG, "You have been kicked off the line. Please login again!");
        //logout();
    }
    @Override
    public void onUserSigExpired() {
        super.onUserSigExpired();
        Log.i(TAG, "Your user signature information has expired");
        //logout();
    }
};
TUILogin.addLoginListener(mLoginListener);
```

```
//로그인
TUILogin.login(context,
1400000001, // 1단계에서 얻은 SDKAppID로 교체하십시오
"denny", // UserID로 바꾸십시오
"xxxxxxxxxxx", // 콘솔에서 UserSig를 계산하고 이 위치에 입력할 수 있습니다
new TUICallback() {
@Override
public void onSuccess() {
Log.i(TAG, "login success");
}
@Override
public void onError(int errorCode, String errorMessage) {
Log.e(TAG, "login failed, errorCode: " + errorCode + " msg:" + errorMessage);
}
});
```

매개변수 설명: 여기에서는 login 함수에 사용되는 몇 가지 주요 매개변수에 대해 자세히 설명합니다.

- **SDKAppID:** 1단계의 마지막 순서에서 이미 얻었으므로 여기서 더 이상 반복하지 않습니다.
- **UserID:** 현재 사용자 ID입니다. 영어 알파벳(a-z, A-Z), 숫자(0~9), 하이픈(-), 언더바(_)의 문자열로 구성합니다.
- **UserSig:** 1단계의 3번째 순서에서 획득한 SecretKey를 사용하여 SDKAppID, UserID 등의 정보를 암호화하면, Tencent Cloud가 현재 사용자의 TRTC 서비스 사용 가능 여부를 식별하는 인증용 티켓인 UserSig를 얻을 수 있습니다. 콘솔의 [보조 툴](#)을 통해 일시적으로 사용 가능한 UserSig를 생성할 수 있습니다.

자세한 내용은 [UserSig 계산 방법](#)을 참고하십시오.

설명 :

이 단계는 지금까지 개발자들로부터 가장 많은 피드백을 받은 단계이기도 합니다. 자주 발생하는 문제는 다음과 같습니다.

- SDKAppID 설정 오류. 중국 사이트의 SDKAppID는 일반적으로 140으로 시작하는 10자리 정수입니다.
- UserSig가 암호화 키(SecretKey)와 일치하지 않는 경우 SecretKey를 UserSig로 직접 구성하지 않고 SDKAppID, UserID, 만료 시간 등의 정보를 SecretKey로 암호화하여 UserSig를 얻습니다.
- UserID는 '1', '123', '111' 등의 간단한 문자열로 설정됩니다. **TRTC는 동일한 UserID의 다중 단말 로그인을 지원하지 않기 때문에**, 여러 사람이 공동으로 개발할 경우, '1', '123', '111'과 같은 UserID는 동료에 의해 쉽게 점유되어 로그인 실패의 원인이 될 수 있으므로 디버깅할 때 인식도가 높은 UserID를 설정하는 것이 좋습니다.

Github의 예시 코드는 genTestUserSig 함수를 사용하여 현재 액세스 프로세스를 더 빠르게 실행할 수 있도록 로컬에서 UserSig를 계산하지만 이 솔루션은 App 코드에 SecretKey를 노출하므로 SecretKey를 업그레이드하고 보호하려면 UserSig의 계산 로직을 서버에 두는 것을 강력히 권장합니다. app은 TUICallKit 컴포넌트가 사용될 때마다 서버에서 실시간으로 계산된 UserSig를 요청합니다.

5단계: 전화 걸기

1:1 영상 통화

TUICallKit의 call 함수를 호출하고 통화 유형 및 수신자의 userId를 지정하여 음성 또는 영상 통화를 시작할 수 있습니다.

```
// 1:1 영상 통화 시작(UserID가 mike라고 가정)
TUICallKit.createInstance(context).call("mike", TUICallDefine.MediaType.Video);
```

매개변수	유형	의미
userId	String	대상 사용자의 ID: "mike"
callMediaType	TUICallDefine.MediaType	통화의 미디어 유형. 예시: TUICallDefine.MediaType.Video

그룹 영상 통화

TUICallKit의 groupCall 함수를 호출하고 통화 유형 및 수신자의 UserID 목록을 지정하여 그룹 내에서 음성 또는 영상 통화를 시작할 수 있습니다.

```
TUICallKit.createInstance(context).groupCall("12345678", Arrays.asList("jane", "mike", "tommy"), TUICallDefine.MediaType.Video);
```

매개변수	유형	의미
groupId	String	그룹 Id. 예시: "12345678"
userIdList	List	대상 사용자의 UserID 값 목록. 예시: {"jane", "mike", "tommy"}
callMediaType	TUICallDefine.MediaType	통화의 미디어 유형. 예시: TUICallDefine.MediaType.Video

설명 :

Android, iOS, macOS의 안내에 따라 그룹을 생성할 수 있습니다. IM TUIKit을 사용하여 채팅 및 통화 시나리오를 원스톱으로 통합할 수도 있습니다.

TUICallKit은 현재 그룹에 속하지 않은 사용자 간의 다중 사용자 영상 통화를 지원하지 않습니다. 그러한 필요가 있는 경우 colleenyu@tencent.com으로 문의하십시오.

6단계: 전화 수신

TUICallKit 컴포넌트는 전화 요청을 받으면 자동으로 알람을 울리는 수신 인터페이스입니다. 그러나 Android 시스템 권한 때문에 다음과 같은 경우로 나뉩니다.

- App이 포그라운드에 있을 때 초대가 수신되면 호출 인터페이스가 자동으로 팝업되고 벨소리가 재생됩니다.
- App이 백그라운드에 있지만 '플로팅 창 권한' 또는 '백그라운드 팝업 애플리케이션'과 같은 권한을 부여한 경우에도 호출 인터페이스가 자동으로 팝업되고 벨소리가 재생됩니다.
- App이 백그라운드에 있고 '플로팅 창 권한' 또는 '백그라운드 팝업 애플리케이션'을 허용하지 않은 경우 TUICallKit은 벨소리를 재생하여 사용자에게 응답하거나 끊으라는 메시지를 표시합니다.
- App 프로세스가 종료된 경우, [오프라인 푸시\(Android\)](#)에 설명된 대로 오프라인 푸시 기능을 사용하여 상태 표시줄 알람을 통해 사용자에게 전화를 받거나 거절하도록 요청할 수 있습니다.

7단계: 더 많은 특성

1. 닉네임 및 프로필 사진 설정

닉네임이나 프로필 사진을 사용자 지정하려면 다음 인터페이스를 사용하여 업데이트할 수 있습니다.

```
TUICallKit.createInstance(context).setSelfInfo("jack", "https://****/user_avatar.png", callback);
```

주의 :

사용자 개인정보 제한으로 인해 친구가 아닌 사용자와의 통화 시 수신자의 닉네임 및 프로필 사진 업데이트가 딜레이될 수 있으며, 통화 성공 후 원활하게 업데이트 됩니다.

2. 오프라인 푸시

상기 단계를 완료한 후 음성 또는 영상 통화를 걸거나 받을 수 있습니다. 그러나 응용 프로그램이 백그라운드에 있거나 응용 프로그램이 닫힌 후에도 사용자가 전화 초대를 받을 수 있도록 하려면 오프라인 푸시 기능도 구현해야 합니다. 자세한 내용은 [오프라인 푸시\(Android\)](#)를 참고하십시오.

3. 플로팅 창 기능

플로팅 창 기능을 활성화하려면 TUICallKit 컴포넌트 초기화 시 다음 인터페이스를 호출하여 이 기능을 활성화할 수 있습니다.

```
TUICallKit.createInstance(context).enableFloatWindow(true);
```

4. 통화 상태 수신

만약 귀하의 업무에서 통화 시작, 종료, 통화 중 네트워크 품질 등 **통화 상태를 수신**하려면 다음 이벤트를 수신할 수 있습니다.

```
TUICallEngine.createInstance(context).addObserver(new TUICallObserver() {
    @Override
    public void onCallBegin(TUICommonDefine.RoomId roomId, TUICallDefine.MediaType callMediaType, TUICallDefine.Role callRole) {
    }
    public void onCallEnd(TUICommonDefine.RoomId roomId, TUICallDefine.MediaType callMediaType, TUICallDefine.Role callRole, long totalTime) {
    }
    public void onUserNetworkQualityChanged(List<TUICommonDefine.NetworkQualityInfo> networkQualityList) {
    }
});
```

5. 사용자 지정 벨소리

수신 전화의 벨소리를 사용자 지정하려면 다음 인터페이스를 통해 설정할 수 있습니다.

```
TUICallKit.createInstance(context).setCallingBell(filePath);
```

FAQ

1. 'The package you purchased does not support this ability'라는 오류 메시지가 표시되면 어떻게 해야 합니까?

오류 메시지는 애플리케이션의 오디오/비디오 통화 기능 패키지가 만료되었거나 활성화되지 않았음을 나타냅니다. TUICallKit을 계속 사용하려면 [1단계](#)의 지침에 따라 음성/영상 통화 기능을 요청하거나 활성화할 수 있습니다.

2. TUICallKit이 크래쉬되고 "No implementation found for xxxx" 로그가 출력되면 어떻게 해야 합니까?

다음은 스택 정보입니다.

```
java.lang.UnsatisfiedLinkError: No implementation found for void com.tencent.liteav.base.Log.nativeWriteLogToNative(int, java.lang.String, java.lang.String) (tried Java_com_tencent_liteav_base_Log_nativeWriteLogToNative and Java_com_tencent_liteav_base_Log_nativeWriteLogToNative__ILjava_lang_String_2Ljava_lang_String_2)
at com.tencent.liteav.base.Log.nativeWriteLogToNative(Native Method)
at com.tencent.liteav.base.Log.i(SourceFile:177)
```

```
at com.tencent.liteav.basic.log.TXCLog.i (SourceFile:36)
at com.tencent.liteav.trtccalling.model.impl.base.TRTCLogger.i (TRTCLogger.java:15)
)
at com.tencent.liteav.trtccalling.model.impl.ServiceInitializer.init (ServiceInitializer.java:36)
at com.tencent.liteav.trtccalling.model.impl.ServiceInitializer.onCreate (ServiceInitializer.java:101)
at android.content.ContentProvider.attachInfo (ContentProvider.java:2097)
at android.content.ContentProvider.attachInfo (ContentProvider.java:2070)
at android.app.ActivityThread.installProvider (ActivityThread.java:8168)
at android.app.ActivityThread.installContentProviders (ActivityThread.java:7709)
at android.app.ActivityThread.handleBindApplication (ActivityThread.java:7573)
at android.app.ActivityThread.access$2600 (ActivityThread.java:260)
at android.app.ActivityThread$H.handleMessage (ActivityThread.java:2435)
at android.os.Handler.dispatchMessage (Handler.java:110)
at android.os.Looper.loop (Looper.java:219)
at android.app.ActivityThread.main (ActivityThread.java:8668)
at java.lang.reflect.Method.invoke (Native Method)
at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run (RuntimeInit.java:513)
at com.android.internal.os.ZygoteInit.main (ZygoteInit.java:1109)
```

TUICallKit은 가상 컴퓨터에서 지원되지 않으며 실제 디바이스에서 실행해야 합니다. 상기 예외가 실제 디바이스에서 발생하는 경우 TUICallKit에 의존하는 TRTC SDK와 같은 SDK의 일부 API는 JNI를 통해 구현되지만 Android Studio는 일부 조건에서 APK를 컴파일할 때 Native .so 라이브러리를 패키징하지 않을 수 있습니다. 이 경우 프로젝트를 다시 Clean하면 됩니다.

설명 :

자세한 내용은 [FAQ\(Android\)](#)를 참고하십시오.

교류 및 피드백

요구 사항이나 피드백이 있는 경우 colleenyu@tencent.com으로 문의하십시오.

iOS

최종 업데이트 날짜: : 2022-11-23 15:50:04

본문에서는 TUICallKit 컴포넌트의 액세스를 최단 시간에 완료하는 방법에 대해 설명합니다. 본 문서를 따르면, 다음 몇 가지 주요 단계를 한 시간 내에 완료하고 최종적으로 UI 인터페이스로 영상 통화 기능을 얻을 수 있습니다.

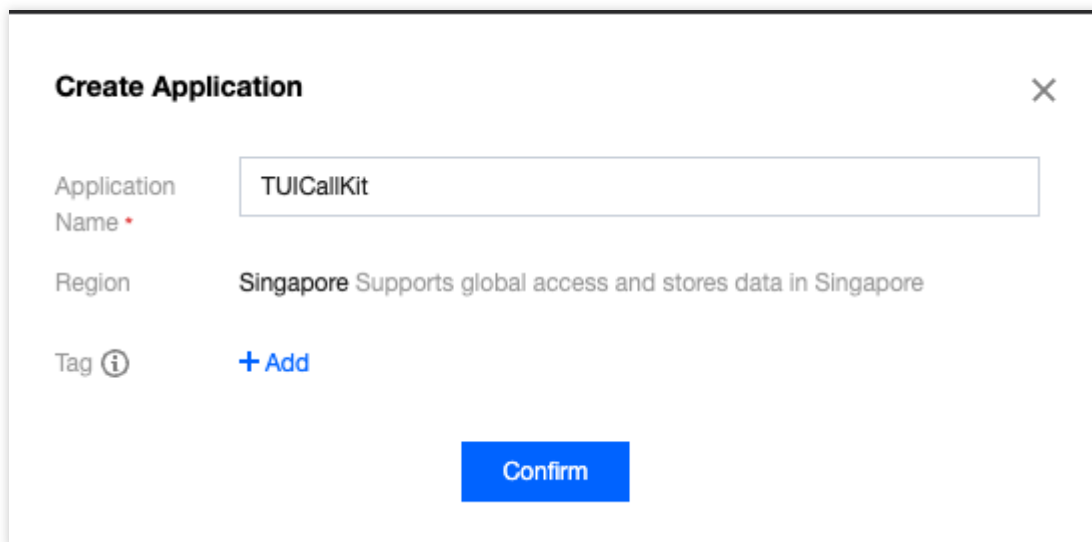
환경 준비

iOS 9.0 (API level 16) 이상.

1단계: 서비스 활성화

TUICallKit은 Tencent Cloud의 두 가지 유료 PaaS 서비스 [Instant Messaging\(IM\)](#) 및 [Tencent Real-Time Communication\(TRTC\)](#)을 기반으로 하는 오디오/비디오 통신 컴포넌트입니다. 아래 단계에 따라 관련 서비스를 활성화하고 60일 무료 베타 서비스를 체험할 수 있습니다.

1. [IM 콘솔](#)에 로그인하고 [애플리케이션 생성](#)을 클릭하고 팝업 대화 상자에 애플리케이션 이름을 입력하고 [확인](#)을 클릭합니다.



Create Application ✕

Application Name

Region **Singapore Supports global access and stores data in Singapore**

Tag ⓘ [+ Add](#)

[Confirm](#)

2. 방금 만든 애플리케이션을 클릭하여 기본 구성 페이지로 들어갑니다. TUICallKit의 7일 무료 평가판을 보려면 페이지 오른쪽 하단 모서리에 있는 TRTC 활성화에서 무료 평가판을 클릭하십시오. 애플리케이션을 공식 릴리스하려

면 [문의](#) 하십시오.

Tencent Real-Time Communication

[View application](#) 

1. To facilitate TRTC integration in the current IM application, we have automatically created a TRTC application with the same SDKAppID as the current IM application in the [TRTC Console](#). Accounts and authentication for both can be reused.
2. Tencent Real-Time Communication (TRTC) is an independent Tencent Cloud service. For billing details, please see [Price](#).

TRTC enables you to implement audio/video call, group audio chat, video conference and other audio/video features in the current IM application.

Audio/video call:Not activated [Try now](#)

주의 :

다양한 비즈니스 요구 사항에 따라 IM 음성/영상 통화 기능의 다양한 유료 버전을 사용할 수 있습니다. 사용 가능한 기능에 대해 자세히 알아보고 적합한 버전을 구입하려면 [문의](#)하십시오.

3. 같은 페이지에서 **SDKAppID**와 ****키(SecretKey)****를 찾아 기록해 둡니다. **4단계: TUI 컴포넌트 로그인**에서 사용될 예정입니다.

Standard Billing Plan

Status **In use**


Plan **Trial**

Expiration time -

[Upgrade](#) [More](#) ▼

App Information

[Edit](#)


SDKAppID 20[REDACTED] 

Application Name TUICallKit

Application Type Video

Application Introduction -

Basic info

Key 8fca0d57d09[REDACTED] 

[Hide key](#)

Key information is sensitive. Keep it confidential and do not disclose it.

Creation time 2022-11-02

Last Modified 2022-11-02

설명 :

참고: 무료 평가판을 클릭하면 이전에 TRTC 서비스를 사용한 적이 있는 일부 사용자에게 다음 메시지가 표시됩니다.

```
[-100013]:TRTC service is suspended. Please check if the package balance is 0 or the Tencent Cloud account is in arrears
```

새로운 IM 음성 및 영상 통화 기능은 TRTC 및 IM의 두 가지 기본 PaaS 서비스가 통합되어 있으므로, TRTC의 무료 할당량(10000분)이 만료되거나 소진되면 이 서비스를 활성화할 수 없습니다. 예시 이미지와 같이 TRTC 콘솔을 클릭하여 SDKAppID에 해당하는 애플리케이션 관리 페이지에서 후불 기능을 활성화한 후 다시 애플리케이션을 활성화하면 음성 및 영상 통화 기능을 정상적으로 체험할 수 있습니다.

2단계: 컴포넌트 가져오기

CocoaPods를 사용하여 다음과 같이 컴포넌트를 가져옵니다.

1. Podfile 에 다음 종속성을 추가합니다.

```
pod 'TUICallKit'
```

2. 다음 명령을 실행하여 컴포넌트를 설치합니다.

```
pod install
```

최신 버전의 TUICallKit을 설치할 수 없는 경우 다음 명령을 실행하여 로컬 CocoaPods 저장소 목록을 업데이트하십시오.

```
pod repo update
```

3단계: 프로젝트 구성 완료

오디오/비디오 기능을 사용하려면 마이크 및 카메라 권한을 부여해야 합니다. App의 Info.plist에 아래 두 항목을 추가합니다. 해당 콘텐츠는 사용자가 마이크 및 카메라 액세스 팝업 창에서 보는 것입니다.

```
<key>NSCameraUsageDescription</key>
<string>CallingApp이 이미지가 포함된 동영상을 촬영하려면 카메라에 액세스 필요</string>
<key>NSMicrophoneUsageDescription</key>
<string>CallingApp이 오디오가 포함된 동영상을 촬영하려면 마이크에 액세스 필요</string>
```

The screenshot shows the 'Info' tab of Xcode project settings for 'TUICallKitApp'. Under 'Custom iOS Target Properties', a table lists various keys and their values. Two rows are highlighted with red boxes:

Key	Type	Value
Application supports indirect input events	Boolean	YES
Bundle identifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
Bundle name	String	\$(PRODUCT_NAME)
InfoDictionary version	String	6.0
Bundle version	String	1
Required background modes	Array	(1 item)
Executable file	String	\$(EXECUTABLE_NAME)
Application requires iPhone environment	Boolean	YES
Application Scene Manifest	Dictionary	(2 items)
Privacy - Microphone Usage Description	String	TUICallKitApp needs access to your microphone, and it can be use
Launch screen interface file base name	String	LaunchScreen
Main storyboard file base name	String	Main
Required device capabilities	Array	(1 item)
Supported interface orientations	Array	(3 items)
Bundle OS Type code	String	\$(PRODUCT_BUNDLE_PACKAGE_TYPE)
Privacy - Camera Usage Description	String	TUICallKitApp needs access to your camera, and it can be used for
Development localization	String	\$(DEVELOPMENT_LANGUAGE)
Supported interface orientations (iPad)	Array	(4 items)
Bundle version string (short)	String	1.0

4단계: TUI 컴포넌트 로그인

프로젝트에 다음 코드를 추가하는 것은 TUICore에서 관련 인터페이스를 호출하여 TUI 컴포넌트의 로그인을 완료하는 것입니다. TUICallKit의 다양한 기능은 로그인이 성공한 후에야 정상적으로 사용할 수 있기 때문에 이 단계는 매우 중요합니다. 따라서 관련 매개변수가 올바르게 구성되었는지 확인하십시오.

- [Objective-C Objectivec](#)
- [Swift](#)

```
// 컴포넌트에 로그인
[TUILogin login:1400000001 // // 1단계에서 얻은 SDKAppID로 교체하십시오
userID:@"denny" // UserID로 교체하십시오
userSig:@"xxxxxxxxxxxx" // 콘솔에서 UserSig를 계산하고 여기에 입력할 수 있습니다
succ:^(
    NSLog(@"login success");
} fail:^(int code, NSString *msg) {
    NSLog(@"login failed, code: %d, error: %@", code, msg);
}
```

매개변수 설명:

여기에서는 login 함수에 사용되는 몇 가지 주요 매개변수에 대해 자세히 설명합니다.

- **sdkAppId**: 1단계의 마지막 순서에서 이미 얻었으므로 여기서 더 이상 반복하지 않습니다.
- **userId**: 현재 사용자 ID입니다. 영어 알파벳(a-z, A-Z), 숫자(0~9), 하이픈(-), 언더바(_)의 문자열로 구성합니다.
- **userSig**: 현재 사용자가 TRTC 서비스를 사용할 수 있는지 여부를 확인하기 위해 Tencent Cloud에서 사용하는 인증 자격 증명입니다. 3단계에서 획득한 SecretKey를 사용하여 SDKAppID, UserID 등의 정보를 암호화하여 얻을

수 있습니다. 콘솔에서 [UserSig 생성](#) 버튼을 클릭하여 임시 UserSig를 생성할 수 있습니다. 자세한 내용은 [UserSig 계산 방법](#)을 참고하십시오.

주의 :

- 이 단계는 지금까지 개발자들로부터 가장 많은 피드백을 받은 단계이기도 합니다. 자주 발생하는 문제는 다음과 같습니다:
 - sdkAppId 설정 오류. 중국 사이트의 SDKAppID는 일반적으로 140으로 시작하는 10자리 정수입니다.
 - userSig가 암호화 키(Secretkey)와 일치하지 않는 경우 SecretKey를 userSig로 직접 구성하지 않고 sdkAppId, userId, 만료 시간 등의 정보를 Secretkey로 암호화하여 userSig를 얻습니다.
 - userId는 '1', '123', '111' 등의 간단한 문자열로 설정됩니다. **TRTC는 동일한 UserID의 다중 단말 로그인을 지원하지 않기 때문에**, 여러 사람이 공동으로 개발할 경우, '1', '123', '111'과 같은 userId는 동료에 의해 쉽게 점유되어 로그인 실패의 원인이 될 수 있으므로 디버깅할 때 인식도가 높은 userId를 설정하는 것이 좋습니다.
- Github의 예시 코드는 genTestUserSig 함수를 사용하여 현재 액세스 프로세스를 더 빠르게 실행할 수 있도록 로컬에서 userSig를 계산하지만 이 솔루션은 App 코드에 SecretKey를 노출하므로 SecretKey를 업그레이드하고 보호하려면 userSig의 계산 로직을 서버에 두는 것을 강력히 권장합니다. App은 TUICallKit 컴포넌트가 사용될 때마다 서버에서 실시간으로 계산된 userSig를 요청합니다.

5단계: 전화 걸기

1:1 영상 통화

TUICallKit의 call 함수를 호출하고 통화 유형 및 수신자의 userId를 지정하여 음성 또는 영상 통화를 시작할 수 있습니다.

- [Objective-C Objectivec](#)
- [Swift](#)

```
// 1:1 영상 통화 시작 (userId가 mike라고 가정)
[[TUICallKit sharedInstance] call:@ "mike" callMediaType:TUICallMediaTypeVideo];
```

매개변수	유형	의미
userId	String	타겟 사용자의 UserID: "mike"
callMediaType	TUICallMediaType	통화의 미디어 유형. 예시: TUICallMediaTypeVideo

그룹 영상 통화

TUICallKit의 groupCall 함수를 호출하고 통화 유형 및 수신자의 UserID 목록을 지정하여 그룹 내에서 음성 또는 영상 통화를 시작할 수 있습니다.

- [Objective-C Objectivec](#)
- [Swift](#)

```
[[TUICallKit sharedInstance] groupCall:@"12345678" userIdList:@[@"denny", @"mike", @"tommy"] callMediaType:TUICallMediaTypeVideo];
```

매개변수	유형	의미
groupId	String	그룹 ID. 예시: @"12345678"
userIdList	Array	대상 사용자의 userId 값 목록, 예시: @[@"denny", @"mike", @"tommy"]
callMediaType	TUICallMediaType	통화의 미디어 유형. 예시: TUICallMediaTypeVideo

설명 :

- [Android](#), [iOS](#), [macOS](#) 지침에 따라 그룹을 생성할 수 있습니다. [IM TUIKit](#)을 사용하여 채팅 및 통화 시나리오를 윈스톱으로 통합할 수도 있습니다.
- TUICallKit은 현재 그룹에 속하지 않은 사용자 간의 다중 사용자 영상 통화를 지원하지 않습니다. 필요한 경우 colleenyu@tencent.com으로 문의하십시오.

6단계: 전화 수신

4단계 이후에 전화가 오면 TUICallKit 컴포넌트가 자동으로 전화 응답 UI를 표시합니다.

7단계: 더 많은 특성

1. 닉네임 및 프로필 사진 설정

닉네임이나 프로필 사진을 사용자 지정하려면 다음 인터페이스를 사용하여 업데이트할 수 있습니다.

- [Objective-C Objectivec](#)
- [Swift](#)

```
[[TUICallKit sharedInstance] setSelfInfo:@"대화명" avatar:@"프로필 사진 url" succ:^
{

} fail:^(int code, NSString *errMsg) {

}];
```

주의 :

사용자 개인정보 제한으로 인해 친구가 아닌 사용자와의 통화 시 수신자의 닉네임 및 프로필 사진 업데이트가 딜레이될 수 있으며, 통화 성공 후 원활하게 업데이트 됩니다.

2. 오프라인 푸시

상기 단계를 완료한 후 음성 또는 영상 통화를 걸거나 받을 수 있습니다. 그러나 App이 백그라운드에 있거나 또는 APP이 닫힌 후 에도 사용자가 전화 초대를 받을 수 있도록 하려면 오프라인 푸시 기능도 구현해야 합니다. 자세한 내용은 [오프라인 푸시\(iOS\)](#)를 참고하십시오.

3. 플로팅 창 기능

플로팅 창 기능을 활성화하려면 TUICallKit 컴포넌트 초기화 시 다음 인터페이스를 호출하여 이 기능을 활성화할 수 있습니다.

- [Objective-C Objectivec](#)
- [Swift](#)

```
[[TUICallKit sharedInstance] enableFloatWindow:YES];
```

4. 통화 상태 수신

만약 귀하의 업무에서 통화 시작, 종료, 통화 중 네트워크 품질 등 **통화 상태를 수신**하려면 다음 이벤트를 수신할 수 있습니다.

- [Objective-C Objectivec](#)
- [Swift](#)

```
[[TUICallEngine sharedInstance] addObserver:self];

- (void)onCallBegin:(TUIRoomId *)roomId callMediaType:(TUICallMediaType)callMedia
aType callRole:(TUICallRole)callRole {
```

```

}
- (void)onCallEnd:(TUIRoomId *)roomId callMediaType:(TUICallMediaType) callMediaT
ype callRole:(TUICallRole) callRole totalTime:(float)totalTime {

}
- (void)onUserNetworkQualityChanged:(NSArray<tuinetworkqualityinfo *=""> *)netwo
rkQualityList {

}

```

5. 사용자 지정 벨소리

수신 전화의 벨소리를 사용자 지정하려면 다음 인터페이스를 통해 설정할 수 있습니다.

- [Objective-C Objectivec](#)
- [Swift](#)

```
[[TUICallKit sharedInstance] setCallingBell:filePath];
```

FAQ

‘The package you purchased does not support this ability’라는 오류 메시지가 표시되면 어떻게 해야 하나요?

오류 메시지는 애플리케이션의 오디오/비디오 통화 기능 패키지가 만료되었거나 활성화되지 않았음을 나타냅니다. TUICallKit을 계속 사용하려면 [1단계](#)의 지침에 따라 음성/영상 통화 기능을 요청하거나 활성화할 수 있습니다.

교류 및 피드백

사용 중 제안이나 의견이 있으시면 colleenyu@tencent.com으로 연락주시기 바랍니다.

Web

최종 업데이트 날짜: : 2023-02-17 16:32:38

TUICallKit 통합

본문에서는 TUICallKit 컴포넌트를 빠르게 통합하여 미리 만들어진 UI로 영상 통화 애플리케이션을 구축하는 방법을 보여줍니다.

컴포넌트를 사용해 보려면 [TUICallKit 기본 demo](#)를 참고하십시오.

시작하기 전에 데스크톱 브라우저가 TRTC를 지원하는지 확인하십시오. 자세한 내용은 [\[환경 요구 사항\]\(#3-환경 요구 사항\)](#)을 참고하십시오.

콘텐츠

[1단계: TRTC 및 IM 활성화](#)

[2단계: TUICallKit 통합](#)

[3단계: TUICallKit 호출](#)

[관련 문서](#)

[FAQ](#)

[1. UserSig는 어떻게 생성하나요?](#)

[2. vite 가져오기](#)

[3. 환경 요구 사항](#)

(1) 브라우저 버전-브라우저 버전)

(2) 네트워크

(3) 웹 사이트 도메인 프로토콜-웹 사이트 도메인 프로토콜)

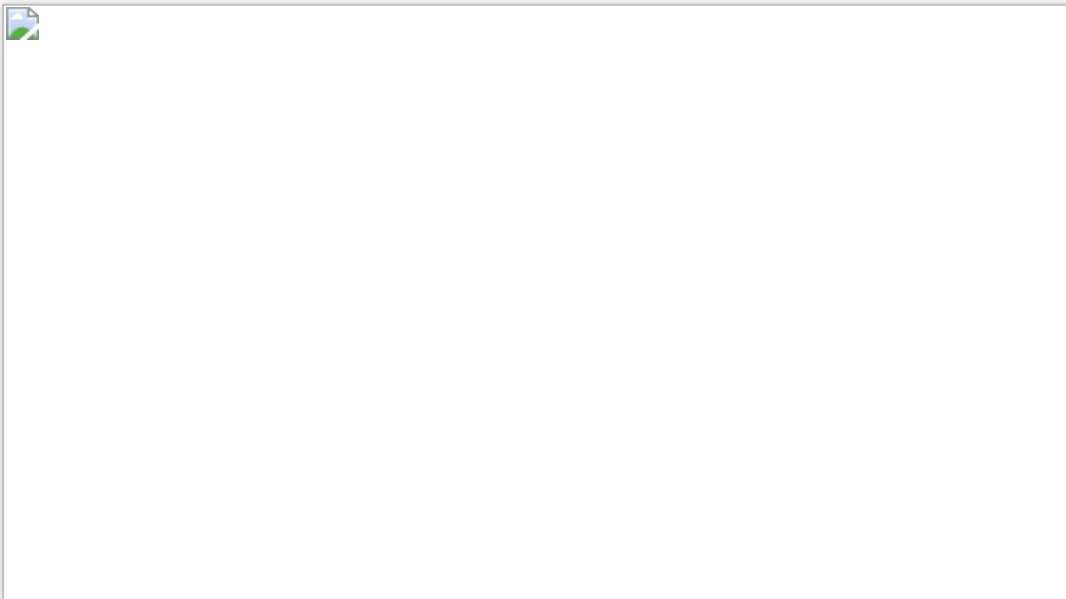
1단계: 서비스 활성화

TUICallKit은 Tencent Cloud의 두 가지 유료 PaaS 제품인 [Instant Messaging\(IM\)](#) 및 [Tencent Real-Time Communication\(TRTC\)](#)을 기반으로 하는 오디오/비디오 통신 컴포넌트입니다. 아래 단계에 따라 관련 서비스를 활성화하고 7일 무료 베타 서비스를 체험할 수 있습니다.

1. [IM 콘솔](#)에 로그인하고 [애플리케이션 생성](#)을 클릭합니다. 팝업 창에서 애플리케이션 이름을 입력하고 [확인](#)을 클릭합니다.



2. 방금 생성한 애플리케이션을 클릭하여 **기본 구성** 페이지로 이동하고 페이지 오른쪽 하단에서 **Tencent TRTC 서비스 활성화의 무료 체험**을 클릭하여 TUICallKit의 7일 무료 베타 서비스를 활성화합니다.



3. 같은 페이지에서 **SDKAppID**와 **Key**를 찾아 기록해둡니다. 나중에 사용하게 됩니다.



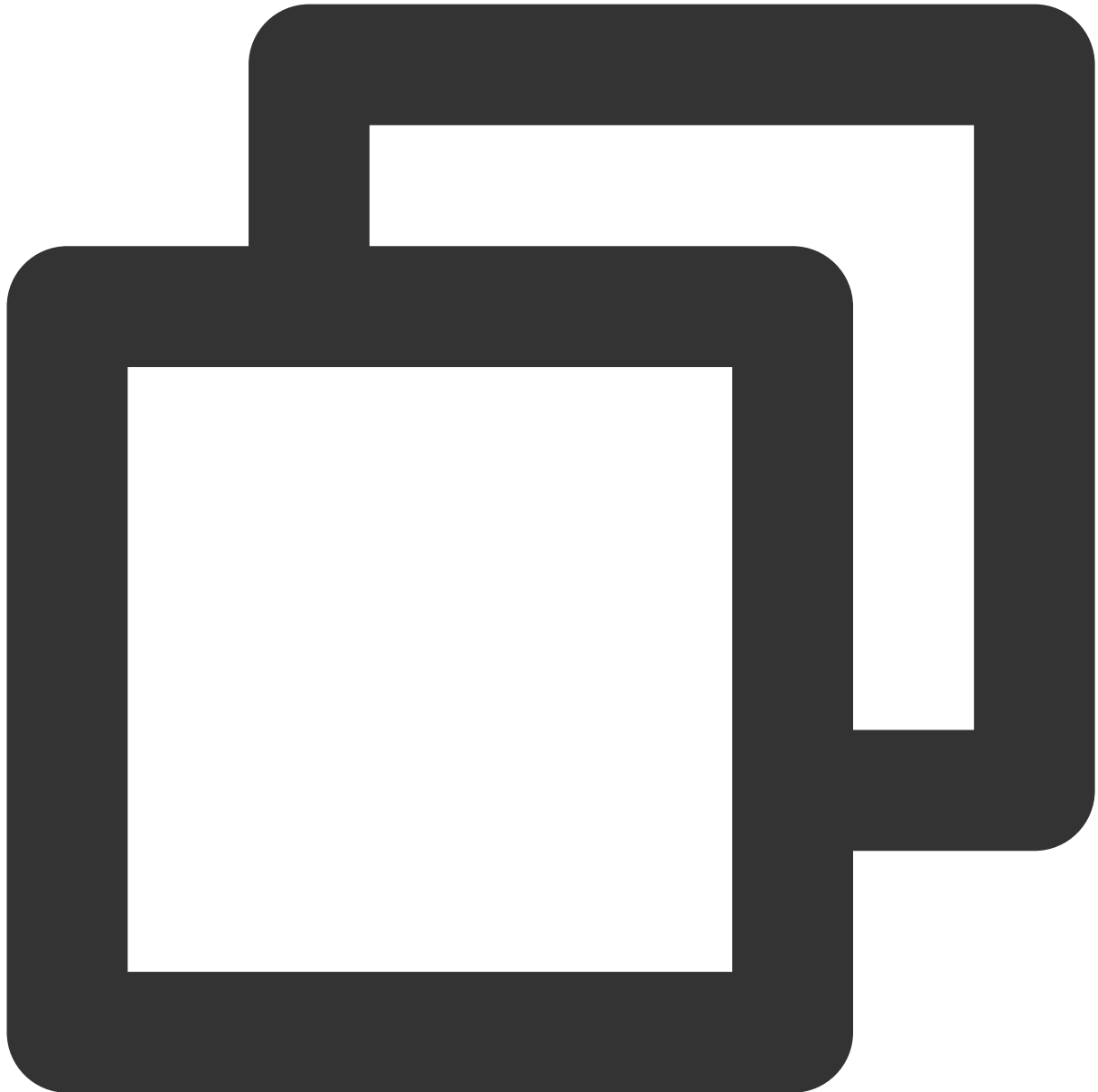
SDKAppID: 비즈니스 격리에 사용되는 IM 애플리케이션 ID. 즉, SDKAppID 값이 다른 호출은 상호 연결할 수 없습니다.

Secretkey: IM의 승인된 사용을 위한 인증 자격 증명 UserSig를 생성하기 위해 SDKAppID와 함께 사용해야 하는 IM 애플리케이션 키입니다. 5단계에서 사용됩니다.

2단계: TUICallKit 컴포넌트 가져오기

방법1: npm을 사용하여 통합

npm을 사용하여 TUICallKit 컴포넌트를 다운로드합니다. 향후 비즈니스를 확장해야 하는 경우 TUICallKit을 프로젝트의 `src/components/` 디렉터리에 복사하는 것이 좋습니다.



```
# macOS
yarn add @tencentcloud/call-uikit-vue # yarn을 설치하지 않았다면 npm install -g yarn을 설치하고
mkdir -p ./src/components/TUICallKit/Web && cp -r ./node_modules/@tencentcloud/call-uikit-vue

# windows
yarn add @tencentcloud/call-uikit-vue # yarn을 설치하지 않았다면 npm install -g yarn을 설치하고
xcopy .\node_modules\@tencentcloud\call-uikit-vue .\src\components\TUICallKit
```

방법2: 소스 코드 통합

1. [GitHub](#)에서 TUICallKit 소스 코드를 다운로드하고 `TUICallKit/Web` 폴더를 프로젝트의 `src/components` 폴더에 복사합니다.



2. 폴더를 입력하고 필요한 종속성을 설치합니다.



```
cd ./src/components/TUICallKit/Web  
yarn # yarn을 설치하지 않았다면 npm install -g yarn을 먼저 실행하십시오
```

3단계: TUICallKit 호출

원하는 페이지에서 TUICallKit 컴포넌트를 호출하여 호출 페이지를 표시할 수 있습니다.

1. TUICallKit UI 가져오기

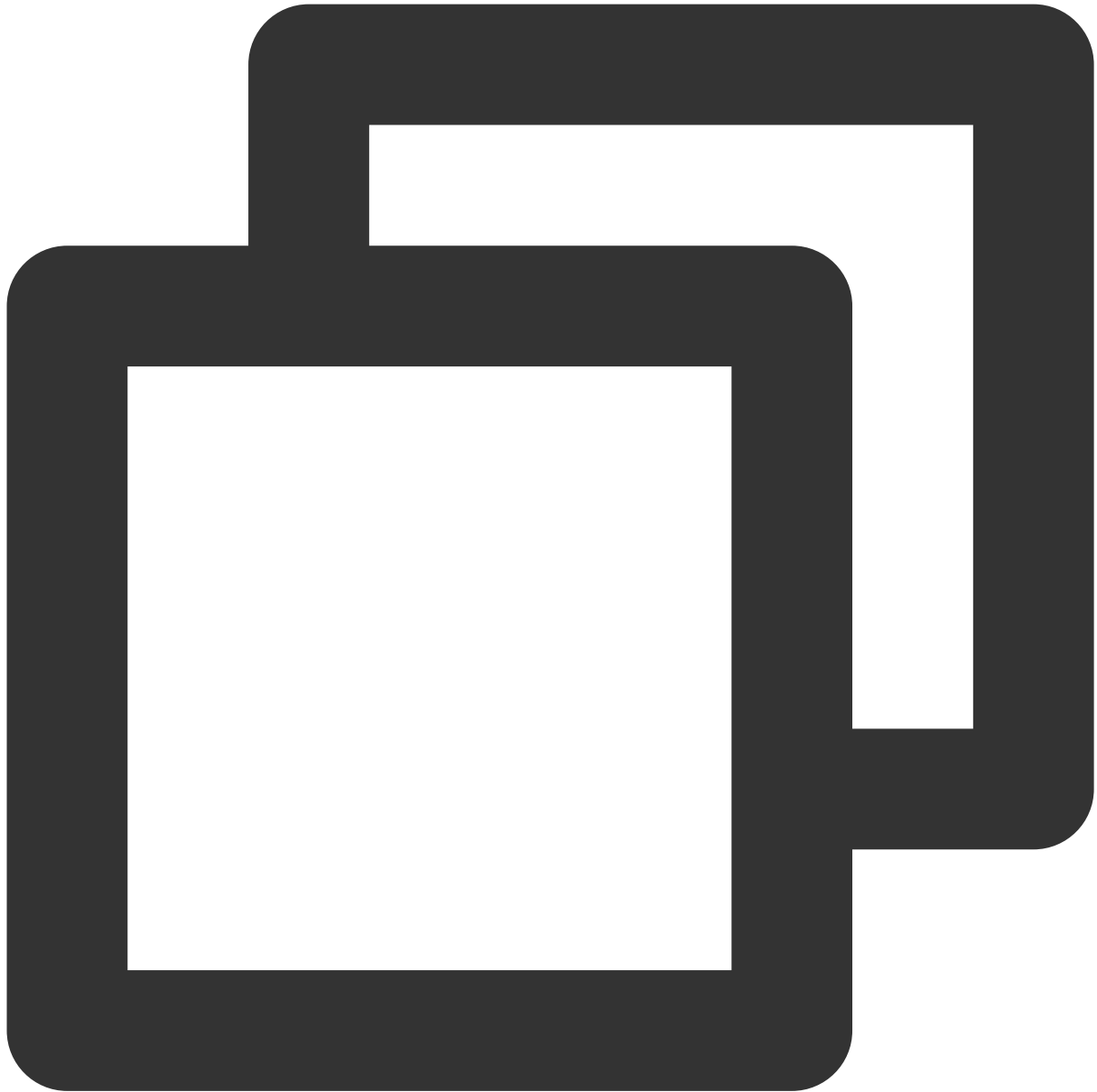


```
<script lang="ts" setup>
import { TUICallKit } from "../components/TUICallKit/Web";
</script>

<template>
  <TUICallKit />
</template>
```

2. 사용자 로그인 및 전화 걸기

2.1 이미 **TUIKit** 컴포넌트를 사용하고 있다면 아래 코드를 추가하여 **TUICallKit**을 플러그인으로 선언합니다. 아직 **TUIKit** 컴포넌트를 사용하고 있지 않다면 건너뛰십시오.



```
import { TUICallKit } from './components/TUICallKit/Web';
TUIKit.use(TUICallKit);
```

2.2 호출하기 전에 컴포넌트를 초기화합니다.



```
import { TUICallKitServer } from './components/TUICallKit/Web';
TUICallKitServer.init({ SDKAppID, userID, userSig });
```

explain

SDKAppID 및 SecretKey를 얻는 방법은 1단계를 참고하십시오.

임시 방법으로 `GenerateTestUserSig.js` 에서 `genTestUserSig(userID)` 를 사용하여 `userSig`를 생성할 수 있습니다.



```
import * as GenerateTestUserSig from "./components/TUICallKit/Web/demos/basic/public  
const { userSig } = GenerateTestUserSig.genTestUserSig(userID, SDKAppID, SecretKey)
```

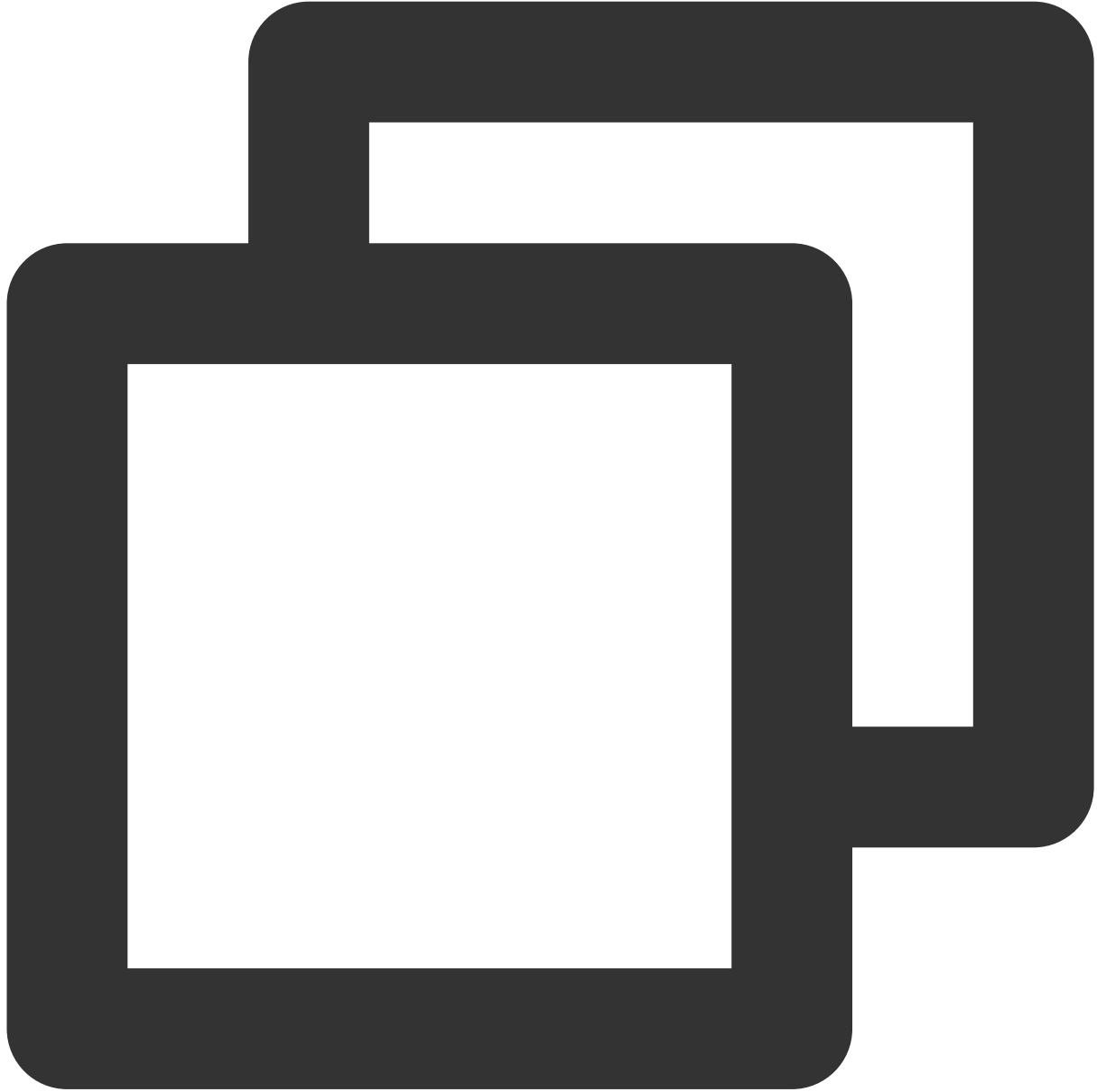
vite를 사용하여 프로젝트를 시작하는 경우 [vite 가져오기](#)도 수행해야 합니다.

notice

본문에서 UserSig를 얻는 방법은 프런트 엔드 코드에서 SECRETKEY를 구성하는 것입니다. 이 방법에서 SECRETKEY는 디컴파일 및 리버스 엔지니어링에 취약합니다. SECRETKEY가 공개되면 공격자가 Tencent Cloud 트래픽을 훔칠 수 있습니다. 따라서 이 방법은 로컬에서 프로젝트를 실행하고 디버깅하는 데에만 적합합니다. 올바른 UserSig 배포 방법은 UserSig의 계산 코드를 서버에 통합하고 App 지향 API를 제공하는 것입니다. UserSig가 필

요할 때 App은 동적 UserSig에 대한 요청을 비즈니스 서버에 보낼 수 있습니다. 자세한 내용은 [Calculating UserSig on the Server](#)를 참고하십시오.

2.3 호출을 구현해야 하는 경우 다음 코드를 실행합니다.



```
import { TUICallKitServer } from './components/TUICallKit/Web';
TUICallKitServer.call({ userID: "123", type: 2 }); // 일대일 통화
TUICallKitServer.groupCall({ userIDList: ["xxx"], groupID: "xxx", type: 2 }); // 그룹
```

이제 첫 번째 전화를 걸 수 있습니다~API 매개변수에 대한 자세한 내용은 [API 문서](#)를 참고하십시오

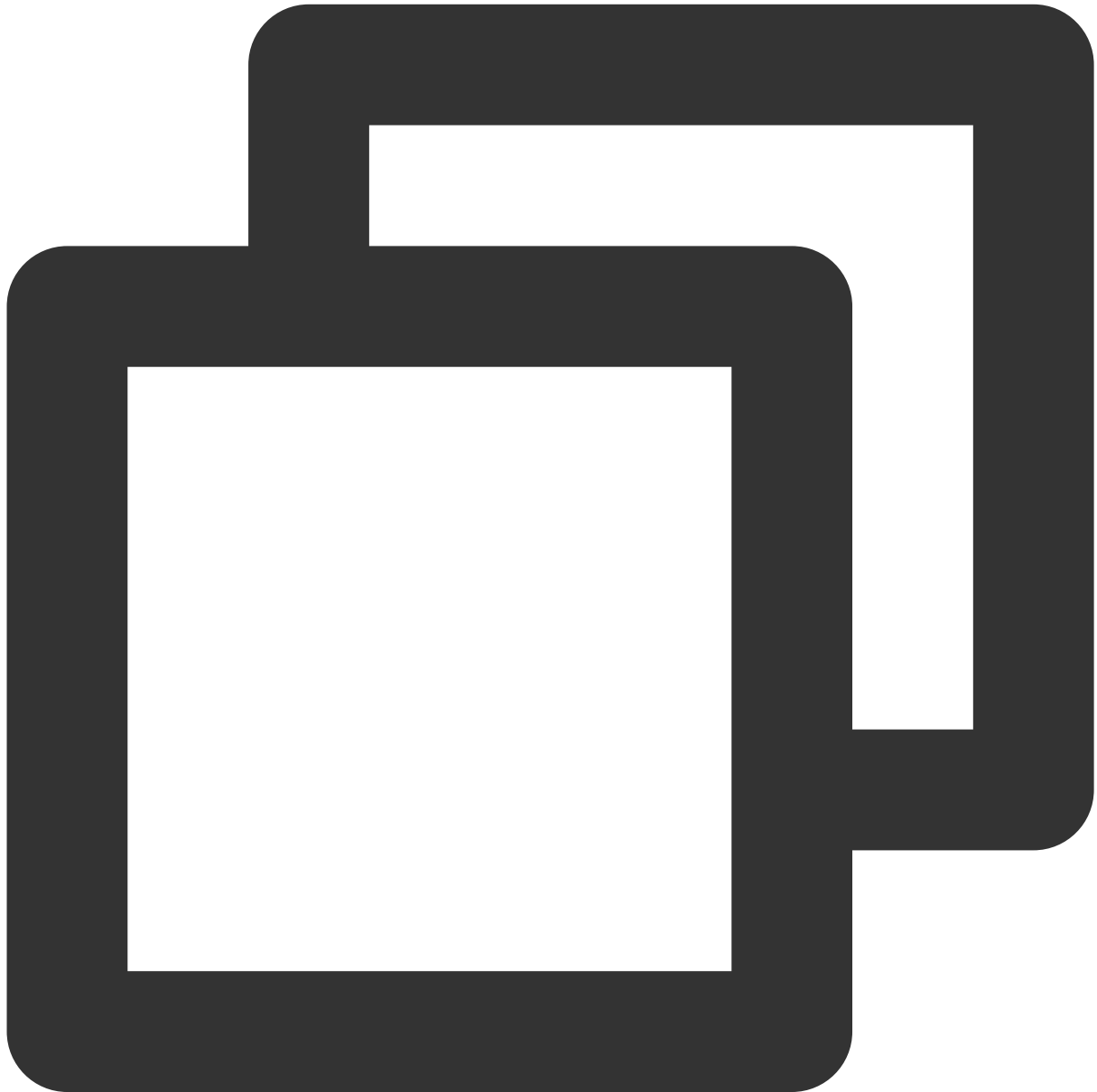
3. 고급 API

이 컴포넌트는 현재 통화 상태를 알려주는 데 사용할 수 있는 `beforeCalling` 및 `afterCalling` 콜백을 제공합니다.

`beforeCalling` : 통화 전에 반환

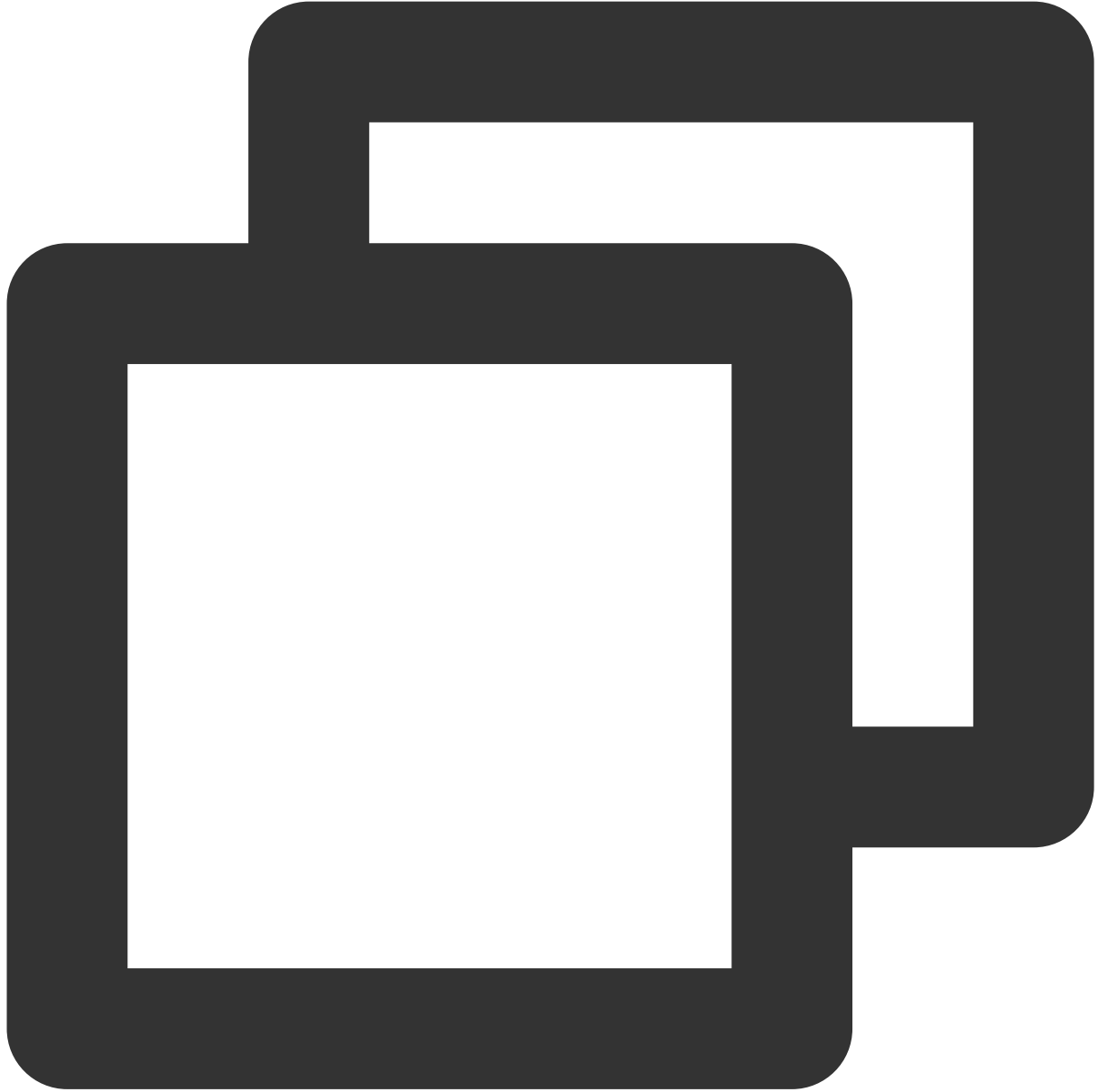
`afterCalling` : 통화 후에 반환

예를 들어, `<TUICallKit />` 컴포넌트를 확장 및 축소하는 데 사용할 수 있습니다([demo](#)에 표시됨).



```
function beforeCalling() {  
  console.log("이 함수는 통화 전에 실행됩니다");  
}  
function afterCalling() {
```

```
console.log("이 함수는 통화 후에 실행됩니다");  
}
```



```
<TUICallKit :beforeCalling="beforeCalling" :afterCalling="afterCalling"/>
```

기타 문서

[TUICallKit API](#)

[TUICallKit 기본 demo](#)[UI 사용자 정의\(TUICallKit\)](#)[TUICallKit \(Web\) FAQs](#)

FAQ

1. UserSig는 어떻게 생성합니까?

UserSig 배포 방식은 UserSig 계산 코드를 서버에 통합하고 프로젝트 지향 API를 제공하는 것입니다. UserSig가 필요한 경우, 프로젝트에서 동적 UserSig에 대한 요청을 비즈니스 서버에 보낼 수 있습니다. 자세한 내용은 [Generating UserSig](#)를 참고하십시오.

2. vite에서 가져오기 문제

vite에서 프로젝트를 생성하는 경우 vite는 다른 파일 패키징 방법을 사용하므로 `lib-generate-test-usersig.min.js` 를 `index.html` 로 가져와야 합니다.



```
// index.html  
<script src="./src/components/TUICallKit/Web/demos/basic/public/debug/lib-generate-
```

GenerateTestUserSig.js 에서 import한 메소드를 주석 처리합니다



```
// import * as LibGenerateTestUserSig from './lib-generate-test-usersig.min.js'
```

3. 환경 요구 사항

(1) 브라우저 버전 요구 사항

최신 버전의 Chrome 브라우저를 사용하여 체험하시기 바랍니다. 이 문서에 연결된 컴포넌트의 주요 데스크톱 브라우저 지원 현황은 다음과 같습니다.

OS	브라우저	최소 브라우저 버전 요구 사항

Mac OS	Safari 브라우저(데스크톱)	11+
Mac OS	Chrome 브라우저(데스크톱)	56+
Mac OS	Firefox 브라우저(데스크톱)	56+
Mac OS	Edge 브라우저(데스크톱)	80+
Windows	Chrome 브라우저(데스크톱)	56+
Windows	QQ 브라우저(데스크톱, WebKit 코어)	10.4+
Windows	Firefox 브라우저(데스크톱)	56+
Windows	Edge 브라우저(데스크톱)	80+

explain

브라우저 호환성에 대한 자세한 내용은 [지원되는 브라우저](#)를 참고하십시오. [TRTC 호환성 확인 페이지](#)를 사용하여 온라인 테스트를 실행할 수도 있습니다.

(2) 네트워크 환경 요구사항

TUICallKit 사용 시 방화벽 제한으로 인해 음성 및 영상 통화가 안될 수 있으니 [방화벽 제한 처리](#)를 참고하여 해당 포트 및 도메인 이름을 방화벽 열로우리스트에 추가하시기 바랍니다.

(3) 웹사이트 도메인 프로토콜의 요구 사항

보안 및 개인 정보 보호를 위해 HTTPS URL만 본 문서에 연결된 컴포넌트의 모든 기능에 액세스할 수 있습니다. 따라서 프로덕션 환경에서 사용자가 제품 기능을 원활하게 체험할 수 있도록 https:// 프로토콜 도메인에 귀하의 웹사이트를 배치하십시오.

시나리오	프로토콜	수신(재생)	발송(게시)	화면 공유	비고
프로덕션 환경	HTTPS 프로토콜	지원	지원	지원	권장
프로덕션 환경	HTTP 프로토콜	지원	미지원	미지원	-
로컬 개발 환경	http://localhost	지원	지원	지원	권장
로컬 개발 환경	http://127.0.0.1	지원	지원	지원	-
로컬 개발 환경	http://[로컬IP]	지원	미지원	미지원	-
로컬 개발 환경	file:///	지원	지원	지원	-

UI 사용자 정의(TUICallKit)

Android

최종 업데이트 날짜: : 2022-11-21 16:34:34

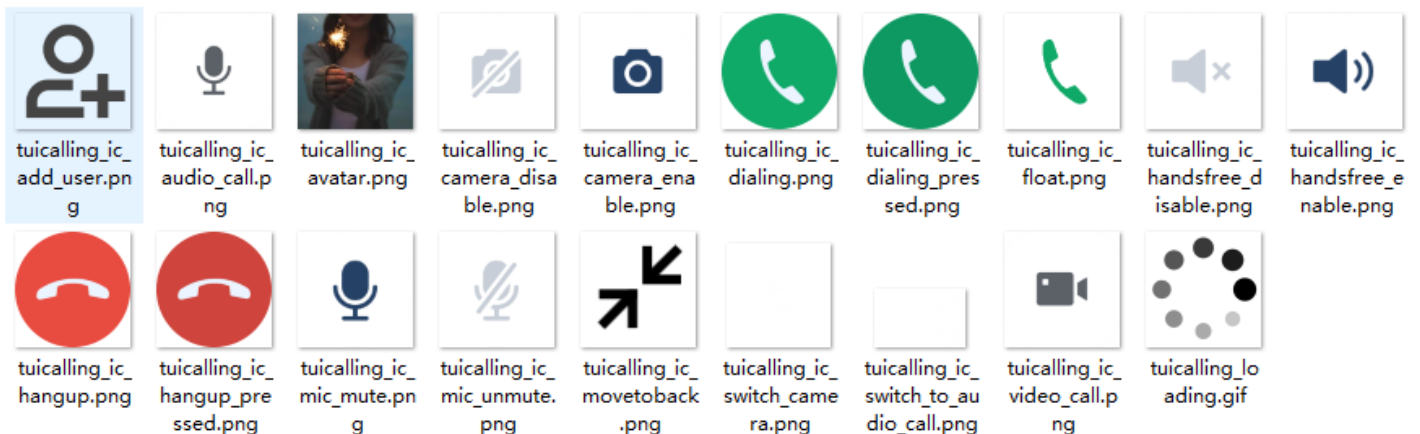
본문은 TUICallKit의 UI를 사용자 정의하는 방법을 설명하고 사용자 정의를 위한 두 가지 방식을 제공합니다: **UI 세밀 조정** 및 **사용자 정의 UI 구현**.

방법1: UI 세밀 조정

[Github](#)의 `Android/tuicallkit` 폴더에서 UI 소스 코드를 직접 수정하여 TUICallKit의 UI를 조정할 수 있습니다.

아이콘 교체

`res\drawable-xxhdpi` 폴더의 아이콘을 직접 교체하여 App에 있는 모든 아이콘의 색조와 스타일을 사용자 지정할 수 있습니다. 아이콘을 바꿀 때 파일 이름이 원래 아이콘과 동일한지 확인하십시오.



벨소리 교체

'res\raw' 폴더에 있는 3개의 오디오 파일을 교체하여 벨소리를 교체할 수 있습니다.

파일명	설명
phone_dialing.mp3	전화 걸기 소리
phone_hangup.mp3	끊는 소리
phone_ringing.mp3	수신 전화 벨소리

텍스트 바꾸기

values-zh 및 values-e의 `strings.xml` 파일을 수정하여 화상 통화 UI의 문자열을 수정할 수 있습니다.

방법2: 사용자 정의 UI 구현

TUICallKit의 전체 호출 기능은 UI가 없는 컴포넌트 TUICallEngine을 기반으로 구현됩니다. tuicallkit 폴더를 삭제하고 전적으로 TUICallEngine을 기반으로 고유한 UI를 구현할 수 있습니다.

TUICallEngine

TUICallEngine은 전체 TUICallKit 컴포넌트의 기본 API입니다. 일대일 음성/영상 및 그룹 통화 걸기, 받기, 거절하기, 끊기 및 기기 작업 등 주요 API를 제공합니다.

API	설명
<code>createInstance</code>	TUICallEngine 인스턴스 생성(싱글톤)
<code>destroyInstance</code>	TUICallEngine 인스턴스(싱글톤) 종료
<code>init</code>	기본적인 음성/영상 통화 기능 인증 완료
<code>addObserver</code>	이벤트 리스너 등록
<code>removeObserver</code>	이벤트 리스너 등록 취소
<code>call</code>	1v1 통화 하기
<code>groupCall</code>	그룹 통화 하기
<code>accept</code>	통화 하기
<code>reject</code>	통화 거부
<code>hangup</code>	전화 끊기
<code>ignore</code>	호출 무시
<code>inviteUser</code>	그룹 통화 중에 사용자 초대
<code>joinInGroupCall</code>	현재 그룹 통화 참가
<code>switchCallMediaType</code>	영상 통화에서 음성 통화로 통화 미디어 유형 전환
<code>startRemoteView</code>	원격 사용자의 비디오 스트림 구독
<code>stopRemoteView</code>	원격 사용자의 비디오 스트림 구독 취소

API	설명
openCamera	카메라 활성화
closeCamera	카메라 비활성화
switchCamera	전면 및 후면 카메라 간 전환
openMicrophone	마이크 활성화
closeMicrophone	마이크 비활성화
selectAudioPlaybackDevice	오디오 재생 장치 선택(장치의 수신기/스피커)
setSelfInfo	사용자 닉네임과 프로필 사진 설정
enableMultiDeviceAbility	TUICallEngine의 다중 장치 로그인 모드 활성화/비활성화(프리미엄 플랜에서 지원)

TUICallObserver

TUICallObserver는 TUICallEngine의 콜백 이벤트 클래스입니다. 이를 사용하여 원하는 콜백 이벤트를 수신할 수 있습니다.

API	설명
onError	통화 중 오류 발생
onCallReceived	통화가 수신됨
onCallCancelled	통화가 취소됨
onCallBegin	통화가 연결됨
onCallEnd	통화가 종료됨
onCallMediaTypeChanged	통화 미디어 유형 변경
onUserReject	xxxx 사용자가 통화를 거부함
onUserNoResponse	xxxx 사용자가 응답하지 않음
onUserLineBusy	xxxx 사용자가 통화 중임
onUserJoin	xxxx 사용자가 통화에 참여함
onUserLeave	xxxx 사용자가 통화를 종료함

API	설명
onUserVideoAvailable	xxx 사용자에게 비디오 스트림이 있는지 여부
onUserAudioAvailable	xxx 사용자에게 오디오 스트림이 있는지 여부
onUserVoiceVolumeChanged	모든 사용자의 볼륨 수준
onUserNetworkQualityChanged	모든 사용자의 네트워크 품질

주요 클래스의 정의

API	설명
TUICallDefine.MediaType	통화 미디어 유형. 열거: 영상 통화 및 음성 통화
TUICallDefine.Role	통화 역할. 열거: 호출자 및 호출 수신자
TUICallDefine.Status	통화 상태. 열거: 유휴, 대기 및 응답 중
TUICommonDefine.RoomId	오디오/비디오 방 ID. 숫자 또는 문자열
TUICommonDefine.Camera	카메라 유형. 열거: 전면 카메라 및 후면 카메라
TUICommonDefine.AudioPlaybackDevice	오디오 재생 장치 유형. 열거: 스피커 및 수신기
TUICommonDefine.NetworkQualityInfo	현재 네트워크 품질에 대한 정보

iOS

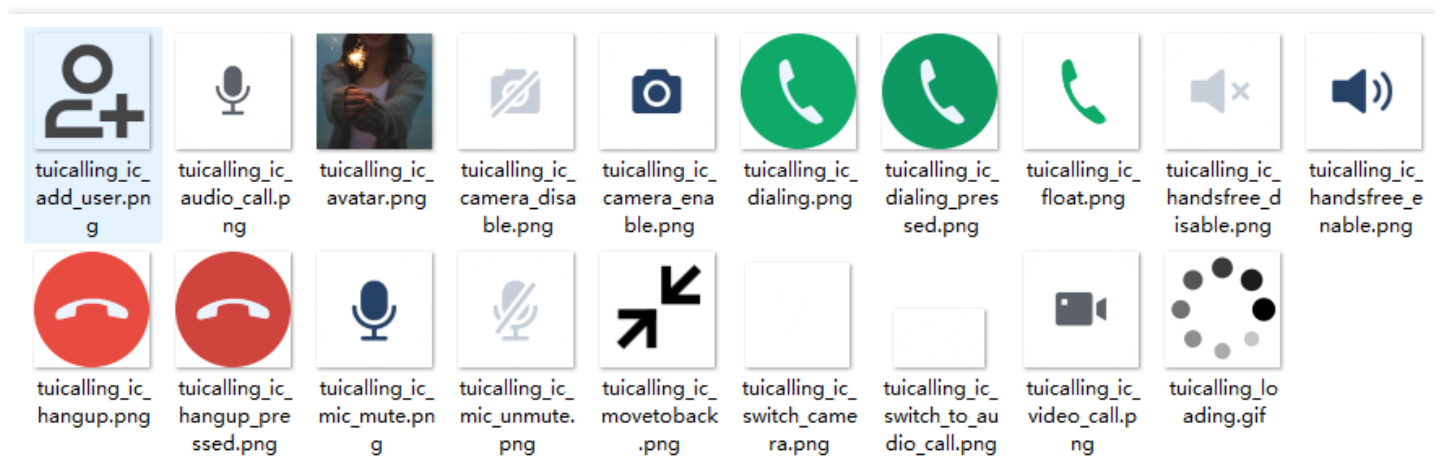
최종 업데이트 날짜: : 2022-11-21 16:34:35

본문은 TUICallKit의 UI를 사용자 정의하는 방법을 설명하고 사용자 정의를 위한 두 가지 방식을 제공합니다: **UI 세밀 조정 및 사용자 정의 UI 구현**.

방법1: UI 세밀 조정

[Github](#)의 'iOS/TUICallKit' 폴더에서 UI 소스 코드를 직접 수정하여 TUICallKit의 UI를 조정할 수 있습니다.

아이콘 교체: `Resources\Calling.xcassets` 폴더의 아이콘을 직접 교체하여 app에 있는 모든 아이콘의 색조와 스타일을 사용자 지정할 수 있습니다. 아이콘을 바꿀 때 파일 이름이 원래 아이콘과 동일한지 확인하십시오.



벨소리 교체: `Resources\AudioFile` 폴더에 있는 3개의 오디오 파일을 교체하여 벨소리를 교체할 수 있습니다.

파일명	설명
phone_dialing.m4a	전화를 거는 소리
phone_hangup.mp3	전화 끊어지는 소리
phone_ringing.flac	수신 전화의 벨소리

텍스트 바꾸기: zh-Hans.lproj 및 en.lproj의 `CallingLocalized.strings` 파일을 수정하여 화상 통화 UI에서 문자열을 수정할 수 있습니다.

방법2: 사용자 정의 UI 구현

TUICallKit의 전체 호출 기능은 UI가 없는 컴포넌트 TUICallEngine을 기반으로 구현됩니다. tuicallkit 폴더를 삭제하고 전적으로 TUICallEngine을 기반으로 고유한 UI를 구현할 수 있습니다.

TUICallEngine

TUICallEngine은 전체 TUICallKit 컴포넌트의 기본 API입니다. 일대일 음성/영상 및 그룹 통화 걸기, 받기, 거절하기, 끊기 및 기기 작업 등 주요 API를 제공합니다.

API	설명
createInstance	TUICallEngine 인스턴스 생성(싱글톤)
destroyInstance	TUICallEngine 인스턴스(싱글톤) 종료
init	기본적인 음성/영상 통화 기능 인증 완료
addObserver	이벤트 리스너 등록
removeObserver	이벤트 리스너 등록 취소
call	1v1 통화 걸기
groupCall	그룹 통화 걸기
accept	통화 하기
reject	통화를 거부함
hangup	전화를 끊음
ignore	호출 무시
inviteUser	그룹 통화 중에 사용자 초대
joinInGroupCall	현재 그룹 통화 참가
switchCallMediaType	영상 통화에서 음성 통화로 통화 미디어 유형 전환
startRemoteView	원격 사용자의 비디오 스트림 구독
stopRemoteView	원격 사용자의 비디오 스트림 구독 취소
openCamera	카메라 활성화
closeCamera	카메라 비활성화

API	설명
switchCamera	전면 및 후면 카메라 간 전환
openMicrophone	마이크 활성화
closeMicrophone	마이크 비활성화
selectAudioPlaybackDevice	오디오 재생 장치(수신기/스피커) 선택
setSelfInfo	사용자 닉네임과 프로필 사진 설정
enableMultiDeviceAbility	TUICallEngine의 다중 장치 로그인 모드를 활성화/비활성화(프리미엄 플랜에서 지원)

TUICallObserver

TUICallObserver는 TUICallEngine의 콜백 이벤트 클래스입니다. 이를 사용하여 원하는 콜백 이벤트를 수신할 수 있습니다.

API	설명
onError	통화 중 오류 발생
onCallReceived	통화가 수신됨
onCallCancelled	통화가 취소됨
onCallBegin	통화가 연결됨
onCallEnd	통화가 종료됨
onCallMediaTypeChanged	통화 미디어 유형이 변경됨
onUserReject	xxxx 사용자가 통화를 거부함
onUserNoResponse	xxxx 사용자가 응답하지 않음
onUserLineBusy	xxxx 사용자가 통화 중임
onUserJoin	xxxx 사용자가 통화에 참여함
onUserLeave	xxxx 사용자가 통화를 종료함
onUserVideoAvailable	xxx 사용자에게 비디오 스트림이 있는지 여부
onUserAudioAvailable	xxx 사용자에게 오디오 스트림이 있는지 여부

API	설명
onUserVoiceVolumeChanged	모든 사용자의 볼륨 수준
onUserNetworkQualityChanged	모든 사용자의 네트워크 품질

주요 클래스의 정의

API	설명
TUICallMediaType	통화 미디어 유형. 열거: 영상 통화 및 음성 통화
TUICallRole	통화 역할. 열거: 호출자 및 호출 수신자
TUICallStatus	통화 상태. 열거: 유휴, 대기 및 응답 중
TUIRoomId	오디오/비디오 방 ID. 숫자 또는 문자열
TUICallCamera	카메라 유형. 열거: 전면 카메라 및 후면 카메라
TUIAudioPlaybackDevice	오디오 재생 장치 유형. 열거: 스피커 및 수신기
TUINetworkQualityInfo	현재 네트워크 품질에 대한 정보

Web

최종 업데이트 날짜: : 2023-02-10 18:25:28

TUICallKit UI 사용자 정의 가이드

본문은 TUICallKit의 UI를 사용자 정의하는 방법을 설명하고 사용자 정의를 위한 두 가지 방식(약간의 UI 조정 및 사용자 정의 UI 구현)을 제공합니다.

방법1: UI 소스 코드 사용

UI 소스 코드를 수정하여 우리가 제공하는 미리 만들어진 UI를 약간 변경할 수 있습니다. UI 소스 코드는 [TUICallKit 프로젝트](#)의 `Web/` 디렉터리에 있습니다.

아이콘 교체

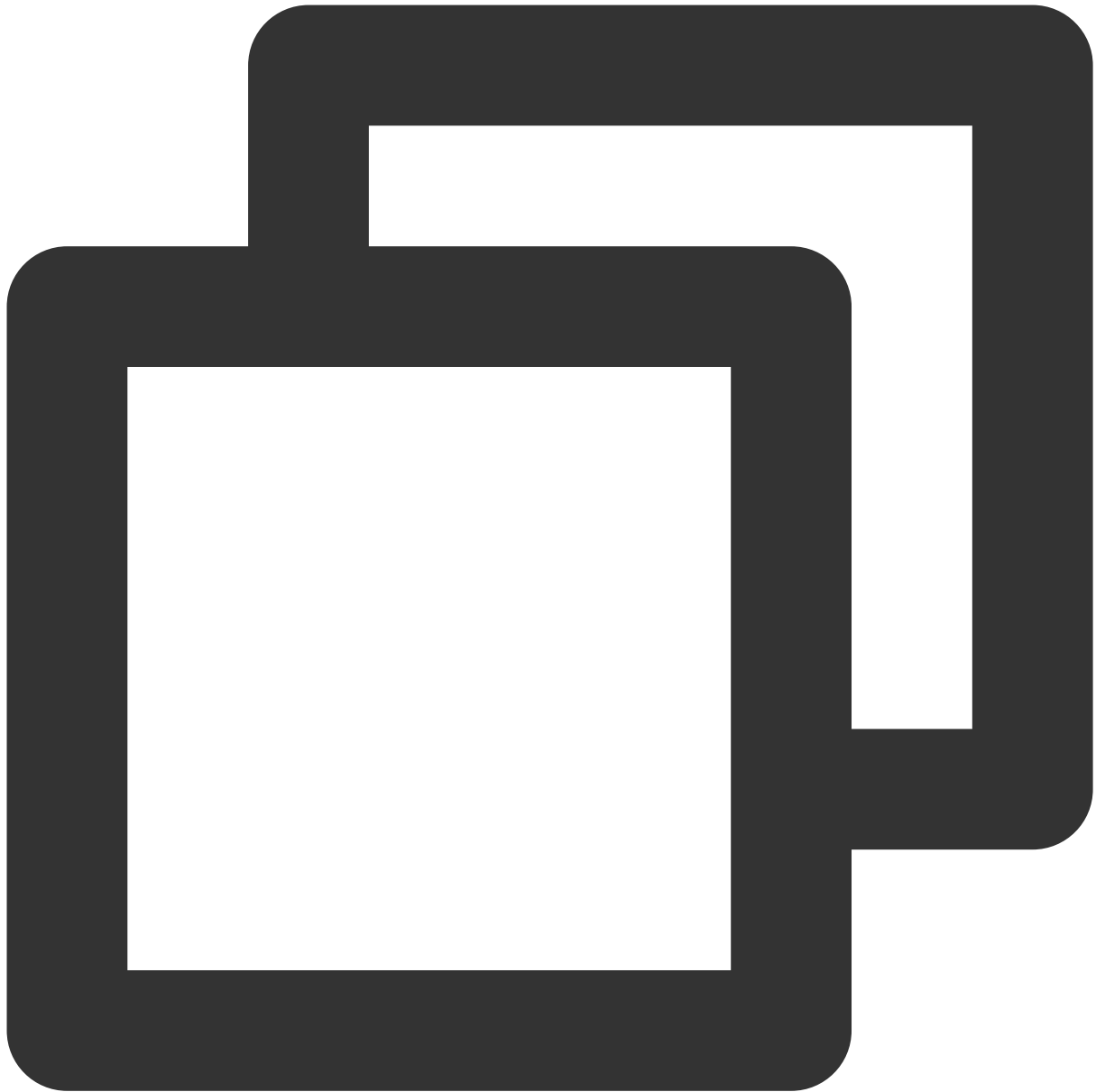
`src/icons` 폴더의 아이콘을 직접 교체하여 애플리케이션에 있는 모든 아이콘의 색조와 스타일을 사용자 지정할 수 있습니다. 아이콘을 바꿀 때 파일 이름이 원래 아이콘과 동일한지 확인하십시오. 아이콘을 미리 보려면

`src/assets` 로 이동하십시오.



UI 레이아웃 변경

src/components/ 폴더에서 UI를 수정하여 음성/영상 통화 UI를 수정할 수 있습니다.



- components/
 - Calling-C2CVideo.vue 일대일 영상 통화
 - Calling-Group.vue 그룹 음성/영상 통화
 - ControlPanel.vue 제어판
 - ControlPanelItem.vue 제어판 항목
 - Dialing.vue 발신 UI, 착신 UI, 일대일 음성 통화 UI
 - MicrophoneIcon.vue 볼륨 레벨의 변화를 표시할 수 있는 마이크 Icon
 - TUICallKit.vue 전체 TUICallKit 컴포넌트

양식 수정

양식 파일은 `src/style.css` 입니다. 원하는 UI 양식을 구현하도록 조정할 수 있습니다.

방법2: 사용자 정의 UI 구현

TUICallKit의 기능은 UI 요소를 포함하지 않는 TUICallEngine SDK를 기반으로 구현됩니다. TUICallEngine을 사용하여 고유한 UI를 구현할 수 있습니다. 자세한 지침은 아래 문서를 참고하십시오.

[TUICallEngine 통합 가이드](#)

[TUICallEngine API](#)

오프라인 푸시(TUICallKit)

Android

최종 업데이트 날짜: : 2022-11-21 16:34:35

오프라인 푸시 기능을 사용하면 애플리케이션이 '백그라운드에서 실행'되거나 '오프라인 상태'에도 오디오/비디오 통화를 수신할 수 있습니다. TUICallKit은 TUIOfflinePush 컴포넌트를 통합하여 오프라인 푸시 기능을 구현합니다. 이 문서는 음성/영상 통화 프로젝트에 `TUIOfflinePush` 컴포넌트를 통합하는 방법을 설명합니다.

준비 사항

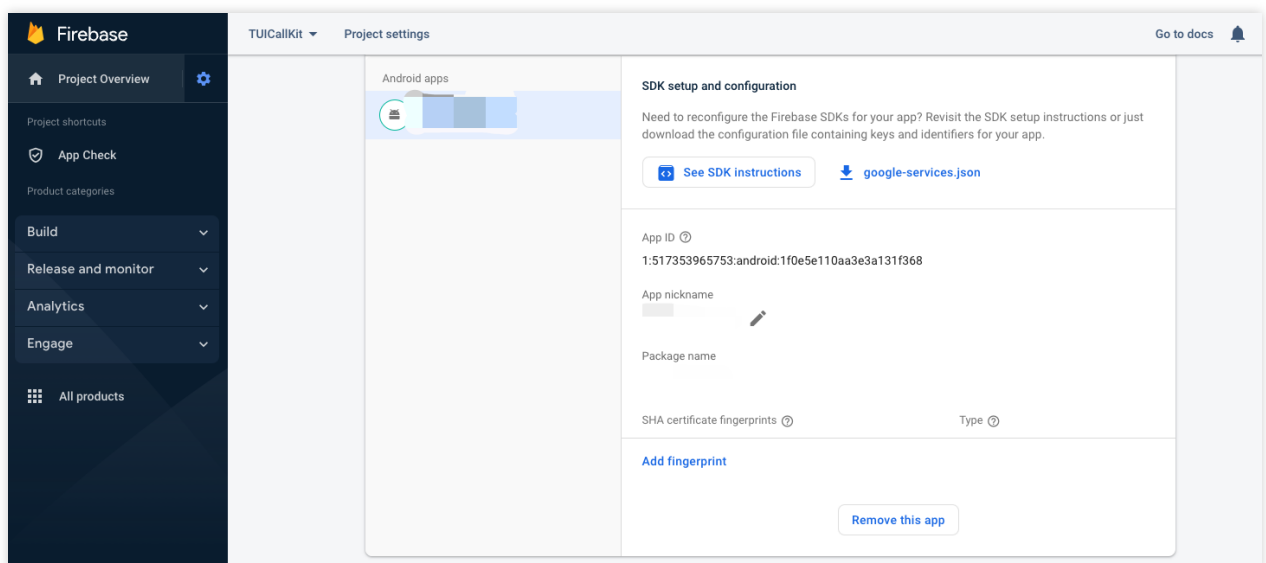
1. **벤더의 푸시 플랫폼에 애플리케이션을 등록합니다.** 오프라인 푸시 기능은 벤더의 자체 채널에 의존합니다. APPID 및 APPKEY와 같은 매개변수를 가져오려면 벤더의 푸시 플랫폼에 애플리케이션을 등록해야 합니다.

주의 :

다음 두 파일은 이후 단계에서 사용됩니다.

- Huawei Push에 등록할 때 `agconnect-services.json` 파일을 다운로드하여 저장합니다.
- Google FCM에 등록할 때 `google-services.json` 파일을 다운로드하여 저장합니다.

Google FCM



2. IM 콘솔에서 구성

메시지 연동을 위해 각 벤더 채널에 등록할 때 동일한 패키지 이름을 입력해야 합니다. 생성된 ID APPID, APPKEY를 기록해 두십시오. 이는 이후 단계에서 사용됩니다.

1단계: 컴포넌트 다운로드 및 가져오기

1. [Github](#)로 이동하여 코드를 복제/다운로드한 다음 아래와 같이 Android 디렉터리의 tuiofflinepush 서브 디렉터리를 프로젝트의 app과 동일한 디렉터리에 복사합니다.



2. 프로젝트 루트 디렉터리에서 `setting.gradle` 파일을 찾아 다음 코드를 추가합니다.

```
include ':tuiofflinepush'
```

3. app 디렉터리에서 'build.gradle' 파일을 찾아 다음 코드를 추가하여 방금 추가한 tuiofflinepush 컴포넌트에 대한 현재 app의 종속성을 선언합니다.

```
api project(':tuiofflinepush')
```

2단계: 프로젝트 구성

1. app 디렉터리에서 `build.gradle` 파일을 찾아 필요에 따라 애플리케이션 패키지의 이름을 바꿉니다.

```
applicationId 'com.****.trtc'
```

2. app 디렉터리에서 `build.gradle` 파일을 찾아 `ViVo` 액세스 매개변수 `VIVO_APPKEY`, `VIVO_APPID` 및 `HONOR_APPID` 를 설정하여 컴파일 또는 실행 오류를 방지합니다.

```
manifestPlaceholders = [
    "VIVO_APPKEY": "PLACEHOLDER",
    "VIVO_APPID" : "PLACEHOLDER",
    "HONOR_APPID": "PLACEHOLDER"
]
```

3. Huawei 및 Google 푸시 플랫폼용 파일 구성: app 디렉터리에 **준비** 과정에서 Google FCM에 등록할 때 저장한 `google-services.json` 파일을 교체합니다. app 디렉터리에 **준비** 과정에서 Huawei Push에 등록할 때 저장한 `agconnect-services.json` 파일을 추가합니다.

4. `PrivateConstants` 파일에 **준비** 과정에서 기록된 ID, APPID, APPKEY를 입력하고 매개변수 설정이 올바른지 확인합니다. 다음 매개변수를 입력합니다.

```
public class PrivateConstants {
    /***** Xiaomi 오프라인 푸시 매개변수 start *****/
    // Tencent Cloud 콘솔에서 타사 푸시 인증서를 업로드한 후 생성된 인증서 ID
    public static final long XM_PUSH_BUZID = 애플리케이션에 할당된 인증서의 ID;
    // Xiaomi 개방형 플랫폼에서 할당한 APPID 및 APPKEY
    public static final String XM_PUSH_APPID = "애플리케이션에 할당된 인증서의 APPID";
    public static final String XM_PUSH_APPKEY = "애플리케이션에 할당된 인증서의 APPKEY";
    /***** Xiaomi 오프라인 푸시 매개변수 end *****/
}
```

주의 :

이 단계는 매우 중요합니다. 매개변수 구성이 올바른지 주의 깊게 확인하십시오.

상기 단계를 완료하면 프로젝트에서 `TUICallKit` 의 오프라인 깨우기 기능을 사용할 수 있습니다.

3단계: 오프라인 알림 콘텐츠 사용자 지정

TUICallKit은 기본 알림 형식을 제공합니다. 단, 알림 내용을 사용자 지정 하려면 [OfflinePushInfoConfig.java](#) 파일을 수정하십시오.

```
public static TUIOfflinePushInfo createOfflinePushInfo(Context context) {
    TUIOfflinePushInfo pushInfo = new TUIOfflinePushInfo();
    pushInfo.setTitle("mike");
    pushInfo.setDesc("You have receive a new call");
    // OPPO는 ChannelID를 설정해야 푸시 메시지를 수신할 수 있으며, 이 channelId는 콘솔과 일치
    // 해야 함
    // OPPO must set a ChannelID to receive push messages. This channelId needs to be
    // the same as the console.
    pushInfo.setAndroidOPPOChannelID("tuikit");
    pushInfo.setIgnoreIOSBadge(false);
    pushInfo.setIOSSound("phone_ringing.mp3");
    return pushInfo;
}
```

FAQ

사용자가 오프라인 푸시 알림을 수신할 수 없는 경우 문제 해결을 위해 [문의하기](#) 하십시오.

1. 알림이 수신되지 않으면 어떻게 해야 하나요?

벤더 콘솔에서 푸시 서비스를 테스트합니다. 알림을 성공적으로 푸시할 수 있으면 벤더 채널이 정상입니다. 그런 다음 `TUIOfflinePush` 콘솔에서 벤더 매개변수가 올바르게 구성되었는지 확인하고, 그렇지 않은 경우 필요에 따라 매개변수를 입력합니다(테스트한 대로, 콘솔에서 vivo x9용 메시지 유형을 구성해야 합니다).

일부 휴대폰은 '중요하지 않은 알림' 폴더로 알림을 보냅니다. 이 경우 상태 표시줄을 아래로 드래그하여 알림이 '중요하지 않은 알림' 폴더에 있는지 확인하십시오.

다음 로그를 필터링하여 `TUIOfflinePush`가 성공적으로 등록되었는지 확인합니다.

```
TUIOfflinePush
```

2. 알림을 받았을 때 잠금 화면이 켜지지 않는 이유는 무엇입니까?

벤더 및 플랫폼에 의해 부과된 제한으로 인해 Android 휴대폰은 화면이 잠겨 있을 때 다른 권한이 필요합니다. 다음과 같이 문제를 해결하십시오.

시스템 잠금 화면 알림 권한이 활성화되어 있는지 확인하십시오. 일부 벤더에서는 통합 제한을 적용했습니다. 예를 들어 오프라인 알림을 수신할 때 Xiaomi 휴대폰의 화면이 켜지지 않으면 **설정 > 잠금 화면**을 선택하고 **알림을 위해 잠금 화면 깨우기**를 켭니다.

애플리케이션의 잠금 화면 알림 권한이 활성화되어 있는지 확인합니다. 예를 들어 Xiaomi 휴대폰에서 **잠금 화면 알림 허용**을 토글해야 합니다.

설명 :

이 문제에 대한 호환성 처리가 필요한 경우 지원을 위해 당사(QQ 그룹 592465424)에 문의하십시오.

3. 오프라인 푸시 알림을 클릭하면 통화 UI를 가져올 수 없는 이유는 무엇인가요?

다음 로그를 필터링하여 통화 요청이 감지되었는지 확인합니다.

```
onReceiveNewInvitation
```

4. 애플리케이션이 백그라운드에서 실행 중일 때 호출 UI를 포그라운드로 가져올 수 없는 이유는 무엇입니까?

애플리케이션을 백그라운드에서 포그라운드로 자동으로 끌어오기 위해서는 App에서 '백그라운드 자동 시작' 또는 '플로팅 창' 권한이 활성화되어 있는지 확인해야 합니다.

다른 벤더 또는 동일한 벤더의 다른 Android 버전이라도 다른 권한 및 권한 이름을 제공합니다. 예를 들어, Xiaomi 6에 대한 **백그라운드 팝업 창** 권한만 활성화하면 되고 **백그라운드 팝업 창 및 플로팅 창 표시** 권한을 모두 활성화해야 합니다.

설명 :

테스트 중에 모든 권한을 수동으로 활성화하면 여전히 호출 UI를 자동으로 포그라운드로 가져올 수 없으며 호환성 처리가 필요합니다.

온클라우드 녹화(TUICallKit)

최종 업데이트 날짜: : 2022-11-21 16:34:35

본문은 TUICallKit의 클라우드 녹화를 활성화하여 중요한 통화를 보관하고 검토하는 방법을 설명합니다. 여기서 [자동 녹화](#)와 [REST API 기반 녹화](#)의 두 가지 방식을 제공합니다.

설명 :

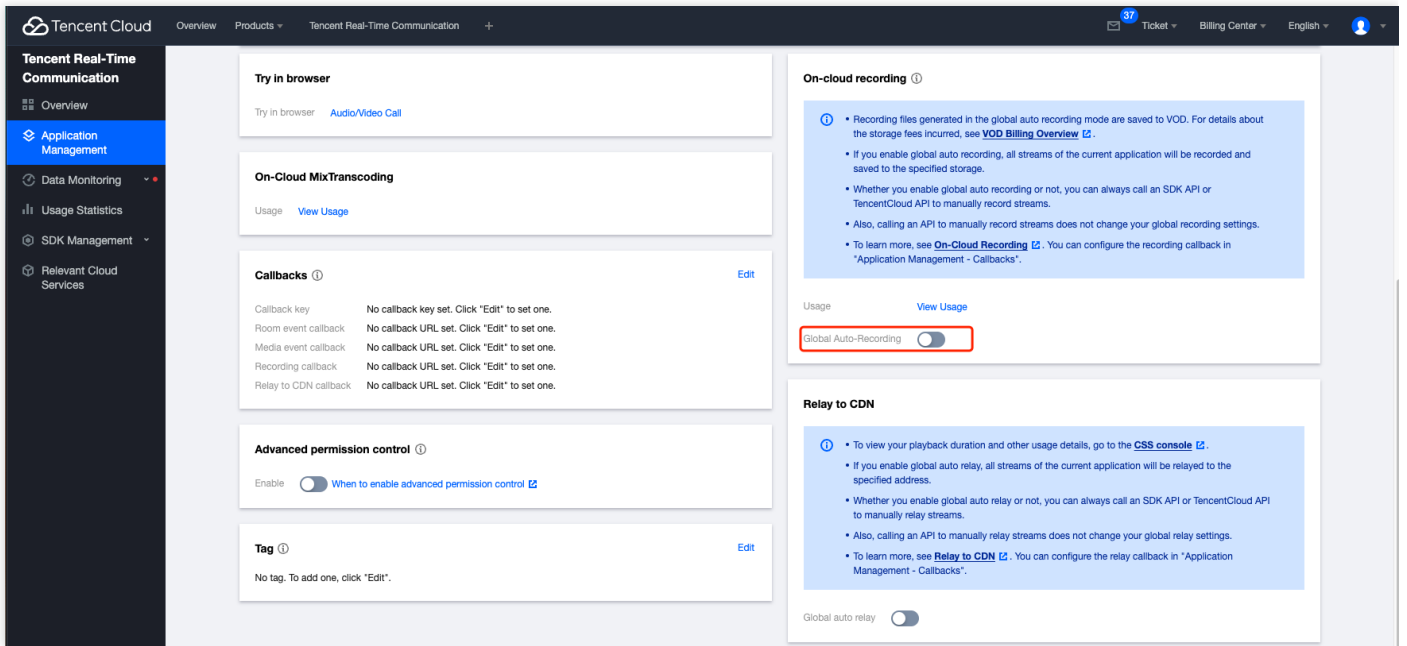
TUICallKit은 여러 기본 Tencent Cloud PaaS 서비스를 통합하며 오디오/비디오 기능은 [TRTC](#)에 의존합니다.

TUICallKit의 온-클라우드 녹화 기능을 사용하려면 [TRTC 콘솔](#)에서 설정해야 합니다.

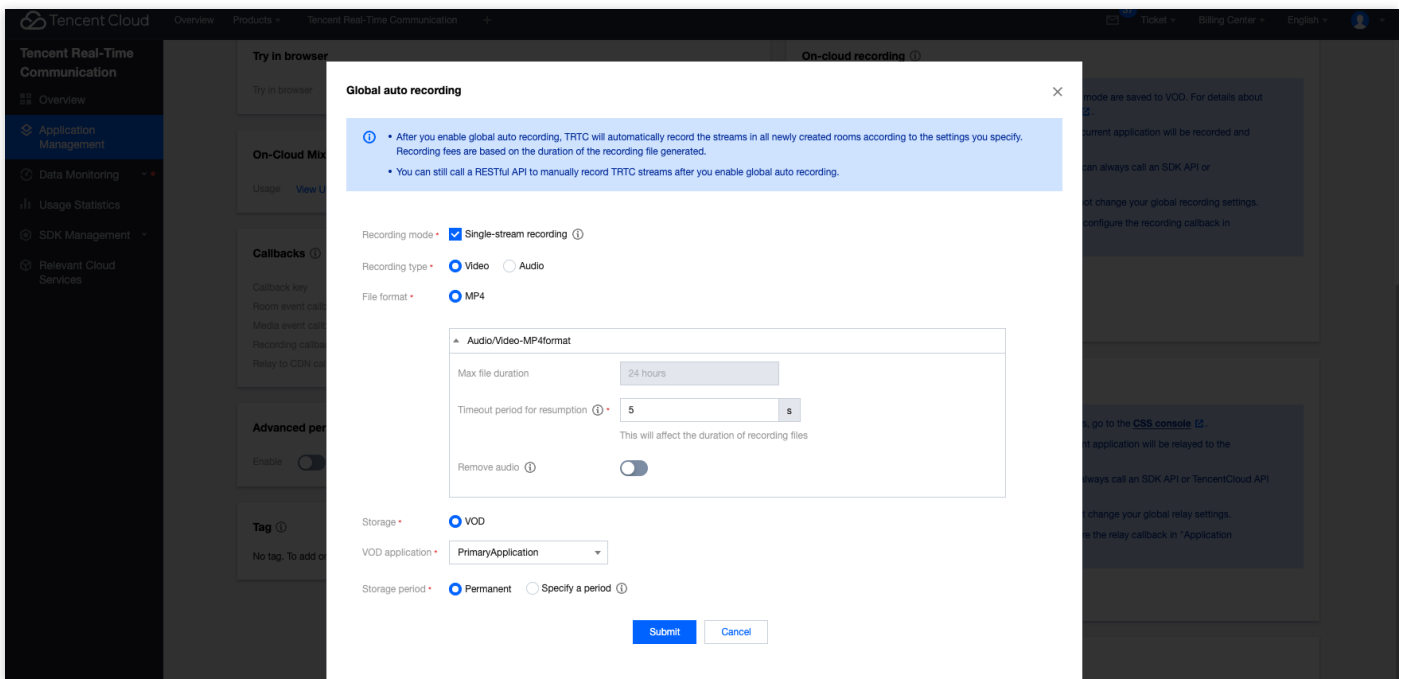
스키마1: 자동 녹화(권장)

자동 녹화 스키마를 사용하는 것이 좋습니다. 이 스키마에서는 사용자가 수동으로 녹음을 시작하거나 중지하지 않습니다. 대신 녹음 작업은 TRTC 백엔드에서 관리되며 업스트림되는 오디오/비디오 스트림이 있을 때 통화가 자동으로 녹음됩니다. **다음과 같이 쉽고 빠르게** 통합할 수 있습니다.

1. TRTC 콘솔의 [애플리케이션 관리](#)에서 대상 SDKAppId의 애플리케이션을 찾아 아래와 같이 기능 구성 페이지로 이동합니다.
2. 기능 구성 페이지의 온클라우드 녹화 구성 카드에서 [온클라우드 녹화 기능을 활성화](#)합니다. [글로벌 자동 녹화 템플릿](#) 카드에서 [템플릿 생성](#)을 클릭합니다.



3. 1v1 및 그룹 통화와 같은 음성/영상 통화 비즈니스 시나리오에 대해 다음 구성 매개변수 설정을 권장합니다. 필요에 따라 녹화 템플릿을 사용자 정의할 수도 있습니다.



주의 :

글로벌 녹화는 최대 8명의 사용자 스트림을 혼합할 수 있습니다. 통화에 8명 이상의 사용자(로컬 사용자 포함)가 포함된 경우 마지막 사용자의 스트림은 녹화되지 않습니다.

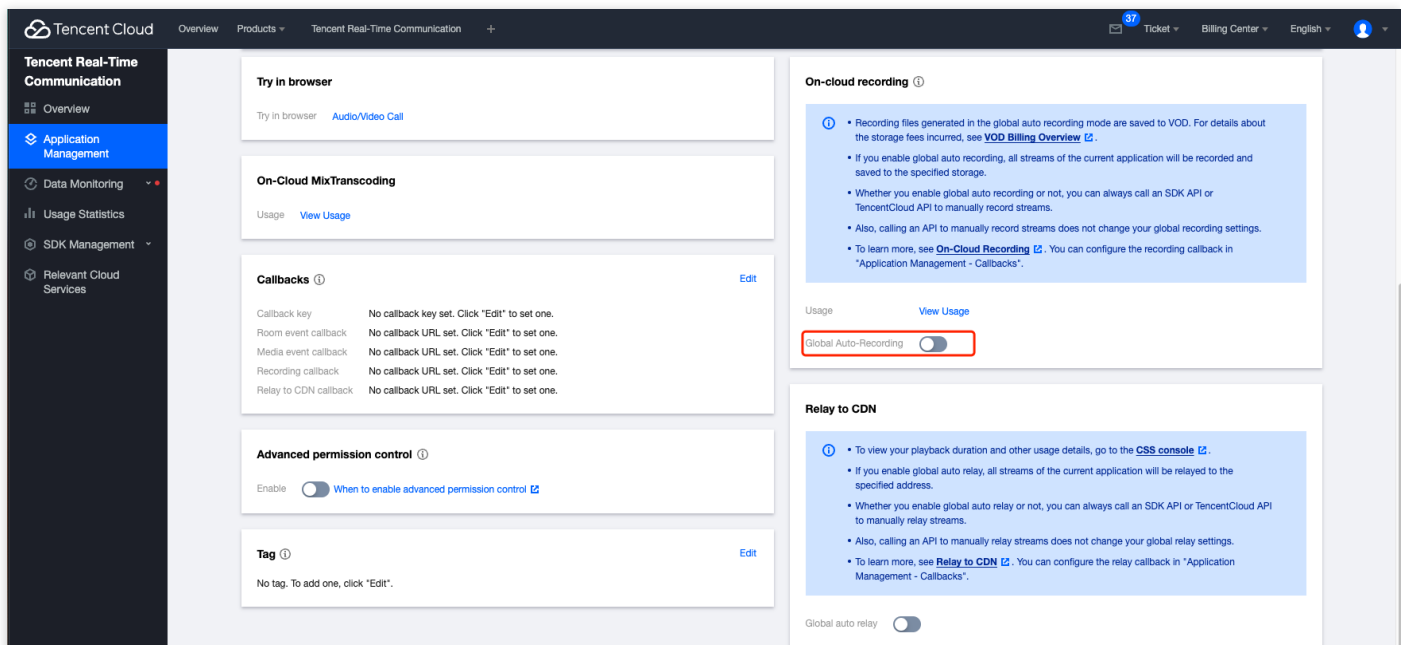
글로벌 자동 녹화 기능이 활성화된 후 전화를 받고 오디오/비디오가 업스트림되면 녹화 작업이 자동으로 실행되고 통화가 종료되면 녹화가 자동으로 중지됩니다. 사용자가 네트워크 상태가 좋지 않거나 기타 예외로 인해 방을 나가는 경우 녹화 백엔드는 불필요한 요금이 발생하지 않도록 구성된 MaxIdleTime 값(최대 유휴 시간, 기본적으로 5s)에 따라 녹화 작업을 자동으로 중지합니다.

4. 템플릿이 생성되면 글로벌 자동 녹화를 선택합니다.

스키마2: REST API 기반 녹화

자동 녹화 스키마가 요구 사항을 충족하지 않는 경우 더 유연한 REST API 기반 녹화 스키마를 사용할 수도 있습니다. 이 스키마에서는 방의 지정된 앵커를 녹화 및 구독하고, 혼합 스트림의 레이아웃을 사용자 지정하고, 녹화 중에 레이아웃 및 구독을 업데이트할 수 있습니다. 그러나 해당 기능을 사용하려면 비즈니스 백엔드 서비스와 함께 사용하고 복잡한 통합 작업을 수행해야 합니다:

1. TRTC 콘솔의 [애플리케이션 관리](#)에서 대상 SDKAppId의 애플리케이션을 찾아 아래와 같이 기능 구성 페이지로 이동합니다.
2. 기능 구성 페이지의 온클라우드 녹화 구성 카드에서 [온클라우드 녹화 기능을 활성화](#)합니다. 수동 사용자 지정 녹화, 즉 REST API 기반 녹화 모드가 기본적으로 선택되어 있습니다.



3. 그런 다음 REST API [CreateCloudRecording](#)를 호출하여 클라우드 녹화를 시작할 수 있습니다. 여기에서 음성/영상 통화가 시작될 때 녹음을 시작하려면 [TUICallObserver](#)의 알림 이벤트를 청취하는 것이 좋습니다. 다음은 Java

코드 샘플입니다.

```
TUICallEngine.createInstance(context).addObserver(new TUICallObserver() {
    @Override
    public void onCallBegin(TUICommonDefine.RoomId roomId, TUICallDefine.MediaType callMediaType, TUICallDefine.Role callRole) {
        // 관련 REST API를 사용하여 녹화 작업을 시작하도록 비즈니스 백엔드에 지시합니다.
    }
});
```

4. 네트워크 불량이나 프로세스 종료 등의 예외적인 상황으로 클라이언트에서 통화가 끊길 수 있으므로 녹음을 중지하려면 TRTC 방 상태 콜백을 구독하고(자세한 내용은 [서버 이벤트 콜백 수신](#) 참고), REST API [DeleteCloudRecording](#)를 호출하여 녹음을 중지하는 것이 좋습니다. TRTC 룸 해제 콜백을 수신하면 온클라우드 녹화 작업을 중지합니다.

FAQ

1. 자세한 녹화 시간은 어떻게 확인하나요?

자세한 녹화 시간은 TRTC 콘솔의 [온클라우드 녹화](#) 페이지에서 확인할 수 있습니다.

2. 녹화된 파일은 어떻게 보나요?

[VOD 콘솔](#)에 로그인하고 왼쪽 사이드바에서 [비디오/오디오 관리](#)를 선택하고 목록 상단의 [접두사 검색](#)을 클릭한 다음 [접두사 검색](#)을 선택하고 검색 창에 키워드를 입력합니다. 녹화 파일 이름은 다음과 같은 형식입니다.

- MP4 단일 스트림 녹화 파일의 파일 이름 형식:

```
<sdkappid>_<roomid>_UserId_s_<userid>_UserId_e_<mediaid>_<index>.mp4
```

- MP4 혼합 스트림 녹화 파일의 파일 이름 형식: `<sdkappid>_<roomid>_<index>.mp4`

Client APIs (TUICallKit)

Android

API 개요

최종 업데이트 날짜: : 2022-11-21 16:34:35

TUICallKit (UI 포함)

TUICallKit은 **UI 요소를 포함**하는 음성/영상 통화 컴포넌트입니다. API를 사용하여 WeChat과 유사한 음성/영상 통화 애플리케이션을 빠르게 구현할 수 있습니다.

API	설명
createInstance	TUICallKit 인스턴스 생성(싱글톤 모드)
setSelfInfo	사용자 대화명과 프로필 사진 설정
call	1v1 통화 호출
groupCall	그룹 통화 호출
joinInGroupCall	그룹 통화 참여
setCallingBell	벨소리 설정
enableMuteMode	음소거 모드 활성화 여부 설정
enableFloatWindow	플로팅 창 활성화 여부 설정

TUICallEngine (UI 없음)

TUICallEngine은 **UI 요소를 포함하지 않는** 오디오/비디오 통화 컴포넌트입니다. TUICallKit이 요구 사항을 충족하지 않는 경우 TUICallEngine의 API를 사용하여 프로젝트를 사용자 정의할 수 있습니다.

API	설명
createInstance	TUICallEngine 인스턴스 생성(싱글톤 모드)
destroyInstance	TUICallEngine 인스턴스 종료(싱글톤 모드)

API	설명
init	기본 음성/영상 통화 기능 인증
addObserver	이벤트 리스너 등록
removeObserver	이벤트 리스너 등록 취소
call	1v1 통화 호출
groupCall	그룹 통화 호출
accept	통화 수신
reject	통화 거절
hangup	통화 종료
ignore	통화 무시
inviteUser	현재 그룹 통화에 사용자 초대
joinInGroupCall	그룹 통화에 참여
switchCallMediaType	화상 통화에서 음성 통화로 통화 미디어 유형 전환
startRemoteView	원격 사용자의 비디오 스트림 구독
stopRemoteView	원격 사용자의 비디오 스트림 구독 취소
openCamera	카메라를 켜
closeCamera	카메라 끄
switchCamera	전면 및 후면 카메라 간 전환
openMicrophone	마이크 활성화
closeMicrophone	마이크 비활성화
selectAudioPlaybackDevice	오디오 재생 장치(수신기/스피커) 선택
setSelfInfo	사용자 대화명과 프로필 사진 설정
enableMultiDeviceAbility	TUICallEngine에 대한 다중 장치 로그인 활성화 여부 설정(프리미엄 패키지에서 지원)

TUICallObserver

TUICallObserver는 TUICallEngine의 콜백 클래스입니다. 이벤트를 수신하는 데 사용할 수 있습니다.

API	설명
onError	통화 중 오류 발생
onCallReceived	통화 수신
onCallCancelled	통화 취소
onCallBegin	통화 연결
onCallEnd	통화 종료
onCallMediaTypeChanged	통화 유형 변경
onUserReject	xxxx 사용자가 통화를 거부함
onUserNoResponse	xxxx 사용자가 응답하지 않음
onUserLineBusy	xxxx 사용자가 통화 중
onUserJoin	xxxx 사용자가 통화에 참여함
onUserLeave	xxxx 사용자가 통화를 종료함
onUserVideoAvailable	xxx 사용자에게 비디오 스트림이 있는지 여부
onUserAudioAvailable	xxx 사용자에게 오디오 스트림이 있는지 여부
onUserVoiceVolumeChanged	모든 사용자의 볼륨 수준
onUserNetworkQualityChanged	모든 사용자의 네트워크 품질

주요 클래스의 정의

API	설명
TUICallDefine.MediaType	통화 유형, 열거: 영상 통화 및 음성 통화
TUICallDefine.Role	통화 역할, 열거: 호출자 및 호출 수신자
TUICallDefine.Status	통화 상태, 열거: 유힬, 대기 및 응답 중

API	설명
TUICommonDefine.RoomId	오디오/비디오 방 Id, 숫자 또는 문자열
TUICommonDefine.Camera	카메라 유형, 열거: 전면 카메라 및 후면 카메라
TUICommonDefine.AudioPlaybackDevice	오디오 재생 장치 유형, 열거: 스피커 및 수신기
TUICommonDefine.NetworkQualityInfo	현재 네트워크 품질

TUICallKit

최종 업데이트 날짜: : 2022-11-21 16:34:35

TUICallKit API

TUICallKit은 **UI 요소를 포함**하는 음성/영상 통화 컴포넌트입니다. API를 사용하여 WeChat과 유사한 음성/영상 통화 애플리케이션을 빠르게 구현할 수 있습니다. 통합에 대한 지침은 [TUICallKit 통합](#)을 참고하십시오.

API 개요

API	설명
createInstance	TUICallKit 인스턴스 생성(싱글톤 모드)
setSelfInfo	대화명 및 프로필 사진 설정
call	1v1 통화 호출
groupCall	그룹 통화 호출
joinInGroupCall	그룹 통화 참여
setCallingBell	벨소리 설정
enableMuteMode	음소거 모드 활성화 여부 설정
enableFloatWindow	플로팅 창 활성화 여부 설정

API 세부 정보

createInstance

이 API는 TUICallKit 싱글톤을 생성하는 데 사용됩니다.

```
TUICallKit createInstance(Context context)
```

setSelfInfo

이 API는 대화명과 프로필 사진을 설정하는 데 사용됩니다. 대화명은 500바이트를 초과할 수 없으며 프로필 사진은 URL로 지정됩니다.

```
void setSelfInfo(String nickname, String avatar, TUICommonDefine.Callback callback)
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
nickname	String	대화명
avatar	String	프로필 사진

call

이 API는 (1v1) 통화를 수행하는 데 사용됩니다.

```
void call(String userId, TUICallDefine.MediaType callMediaType)
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	String	대상 userId
callMediaType	TUICallDefine.MediaType	비디오 또는 오디오일 수 있는 통화 유형

groupCall

이 API는 그룹 통화에 참여하는 데 사용됩니다.

주의 :
그룹 통화를 하기 전에 먼저 IM 그룹을 생성해야 합니다.

```
void groupCall(String groupId, List<String> userIdList, TUICallDefine.MediaType callMediaType) {
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
groupId	String	그룹 ID
userIdList	List	대상 userId
callMediaType	TUICallDefine.MediaType	비디오 또는 오디오일 수 있는 통화 유형

joinInGroupCall

이 API는 그룹 통화에 참여하는 데 사용됩니다.

주의 :

그룹 통화를 하기 전에 먼저 IM 그룹을 생성해야 합니다.

```
void joinInGroupCall(TUICommonDefine.RoomId roomId, String groupId, TUICallDefine.MediaType callMediaType, TUICommonDefine.Callback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	TUICommonDefine.RoomId	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정입니다
groupId	String	그룹 ID
callMediaType	TUICallDefine.MediaType	비디오 또는 오디오일 수 있는 통화 유형

setCallingBell

이 API는 벨소리를 설정하는 데 사용됩니다. filePath는 액세스 가능한 로컬 파일 URL이어야 합니다.

벨소리 세트는 장치와 연결되어 있으며 사용자에게 따라 변경되지 않습니다.

벨소리를 재설정하려면 filePath 에 빈 문자열을 전달하십시오.

```
void setCallingBell(String filePath);
```

enableMuteMode

이 API는 음소거 모드를 켜지 여부를 설정하는 데 사용됩니다.

```
void enableMuteMode(boolean enable);
```

enableFloatWindow

이 API는 플로팅 창 활성화 여부를 설정하는 데 사용됩니다.

기본값은 'false'이며 통화 보기의 왼쪽 상단 모서리에 있는 플로팅 창 버튼이 숨겨져 있습니다. true로 설정하면 버튼이 표시됩니다.

```
void enableFloatWindow(boolean enable);
```

TUICallEngine

최종 업데이트 날짜: : 2022-11-21 16:34:36

TUICallEngine API

TUICallEngine은 **UI 요소를 포함하지 않는** 오디오/비디오 통화 컴포넌트입니다. TUICallKit이 요구 사항을 충족하지 않는 경우 TUICallEngine의 API를 사용하여 프로젝트를 사용자 정의할 수 있습니다.

API 개요

API	설명
createInstance	TUICallEngine 인스턴스 생성(싱글톤 모드)
destroyInstance	TUICallEngine 인스턴스 종료(싱글톤 모드)
init	기본 음성/영상 통화 기능 인증
addObserver	이벤트 리스너 등록
removeObserver	이벤트 리스너 등록 취소
call	1v1 통화 호출
groupCall	그룹 통화 호출
accept	통화 수신
reject	통화 거절
hangup	통화 종료
ignore	통화 무시
inviteUser	현재 그룹 통화에서 사용자 초대
joinInGroupCall	그룹 통화에서 참여
switchCallMediaType	통화 유형 변경. 예를 들어, 화상 통화에서 음성 통화로 전환
startRemoteView	원격 사용자의 비디오 스트림 구독
stopRemoteView	원격 사용자의 비디오 스트림 구독 취소

API	설명
openCamera	카메라를 켜
closeCamera	카메라를 끄
switchCamera	전면 및 후면 카메라 간 전환
openMicrophone	마이크를 켜
closeMicrophone	마이크를 끄
selectAudioPlaybackDevice	오디오 재생 장치(수신기 또는 스피커) 선택
setSelfInfo	대화명 및 프로필 사진 설정
enableMultiDeviceAbility	TUICallEngine에 대한 다중 장치 로그인 활성화 여부 설정(프리미엄 패키지에서 지원)

API 세부 정보

createInstance

이 API는 TUICallEngine 싱글톤을 생성하는 데 사용됩니다.

```
TUICallEngine createInstance(Context context)
```

destroyInstance

이 API는 TUICallEngine 싱글톤을 종료하는 데 사용됩니다.

```
void destroyInstance();
```

init

이 API는 TUICallEngine을 초기화하는 데 사용됩니다. 다른 API를 호출하기 전에 호출 서비스를 인증하고 다른 필수 작업을 수행하기 위해 호출합니다.

```
void init(int sdkAppId, String userId, String userSig, TUICommonDefine.Callback callback)
```


매개변수는 다음과 같습니다.

매개변수	유형	의미
sdkAppId	int	TRTC 콘솔 > 애플리케이션 관리 > 애플리케이션 정보 에서 SDKAppID를 확인할 수 있음
userId	String	현재의 사용자 ID. 문자열 유형은 영어 알파벳(a-z, A-Z), 숫자(0-9), 대시 부호(-), 언더바(_)만 허용
userSig	String	Tencent Cloud의 독점 보안 서명. 계산 및 사용 방법은 UserSig 참고
callback	TUICommonDefine.Callback	초기화 콜백, 'onSuccess'는 초기화가 성공했음을 표시

addObserver

이 API는 'TUICallObserver' 이벤트를 수신할 이벤트 리스너를 등록하는 데 사용됩니다.

```
void addObserver(TUICallObserver observer);
```

removeObserver

이 API는 이벤트 리스너를 등록 해제하는 데 사용됩니다.

```
void removeObserver(TUICallObserver observer);
```

call

이 API는 (1v1) 통화를 수행하는 데 사용됩니다.

```
void call(TUICommonDefine.RoomId roomId, String userId, TUICallDefine.MediaType callMediaType,
TUICallDefine.CallParams params, TUICommonDefine.Callback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	TUICommonDefine.RoomId	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정입니다
userId	String	대상 userId
callMediaType	TUICallDefine.MediaType	비디오 또는 오디오일 수 있는 통화 유형

매개변수	유형	의미
params	TUICallDefine.CallParams	추가 매개변수. 예를 들어 오프라인 알림을 사용자 지정하는 데 사용할 수 있음

groupCall

이 API는 그룹 통화를 하는 데 사용됩니다. 그룹 통화를 하기 전에 IM 그룹을 생성해야 합니다.

```
void groupCall(TUICommonDefine.RoomId roomId, String groupId, List<String> userIdList,
TUICallDefine.MediaType callMediaType, TUICallDefine.CallParams params,
TUICommonDefine.Callback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	TUICommonDefine.RoomId	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정임
groupId	String	그룹 ID
userIdList	List	대상 userId
callMediaType	TUICallDefine.MediaType	비디오 또는 오디오일 수 있는 통화 유형
params	TUICallDefine.CallParams	추가 매개변수. 예를 들어 오프라인 알림을 사용자 지정하는 데 사용할 수 있음

accept

이 API는 통화를 수락하는 데 사용됩니다. 'onCallReceived()' 콜백을 수신한 후 이 API를 호출하여 통화를 수락할 수 있습니다.

```
void accept(TUICommonDefine.Callback callback);
```

reject

이 API는 통화를 거절하는 데 사용됩니다. 'onCallReceived()' 콜백을 수신한 후 이 API를 호출하여 통화를 거부할 수 있습니다.

```
void reject(TUICommonDefine.Callback callback);
```

ignore

이 API는 통화를 무시하는 데 사용됩니다. `onCallReceived()`를 수신한 후 이 API를 호출하여 통화를 무시할 수 있습니다. 호출자는 'onUserLineBusy' 콜백을 수신합니다.

참고: 프로젝트에 라이브 스트리밍 또는 회의가 포함된 경우 이 API를 사용하여 '회의 중' 또는 '방송 중' 기능을 구현할 수도 있습니다.

```
void ignore(TUICommonDefine.Callback callback);
```

hangup

이 API는 통화를 종료하는 데 사용됩니다.

```
void hangup(TUICommonDefine.Callback callback);
```

inviteUser

이 API는 현재 그룹 통화에서 사용자를 초대하는 데 사용됩니다.

이 API는 그룹 통화의 참가자가 새 사용자를 초대하기 위해 호출합니다.

```
void inviteUser(List<String> userIdList, TUICallDefine.CallParams params,
TUICommonDefine.ValueCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userIdList	List	대상 userId
params	TUICallDefine.CallParams	추가 매개변수. 예를 들어 오프라인 알림을 사용자 지정하는 데 사용할 수 있음

joinInGroupCall

이 API는 그룹 통화에서 참여하는 데 사용됩니다.

이 API는 그룹 구성원이 그룹 통화에서 참여하기 위해 호출합니다.

```
void joinInGroupCall(TUICommonDefine.RoomId roomId, String groupId,
TUICallDefine.MediaType callMediaType, TUICommonDefine.Callback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	TUICommonDefine.RoomId	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정임
groupId	String	그룹 ID
callMediaType	TUICallDefine.MediaType	비디오 또는 오디오일 수 있는 통화 유형

switchCallMediaType

이 API는 통화 유형을 변경하는 데 사용됩니다.

```
void switchCallMediaType(TUICallDefine.MediaType callMediaType);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callMediaType	TUICallDefine.MediaType	비디오 또는 오디오일 수 있는 통화 유형

startRemoteView

이 API는 원격 사용자의 비디오 스트림을 구독하는 데 사용됩니다. 작동하려면 `setRenderView` 다음에 호출해야 합니다.

```
void startRemoteView(String userId, TUIVideoView videoView, TUICommonDefine.PlayCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	String	대상 userId
videoView	TUIVideoView	렌더링할 뷰

stopRemoteView

이 API는 원격 사용자의 비디오 스트림 구독을 취소하는 데 사용됩니다.

```
void stopRemoteView(String userId);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	String	대상 userId

openCamera

이 API는 카메라를 켜는 데 사용됩니다.

```
void openCamera(TUICommonDefine.Camera camera, TUIVideoView videoView, TUICommonDefine.Callback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
camera	TUICommonDefine.Camera	전면/후면 카메라
videoView	TUIVideoView	렌더링할 뷰

closeCamera

이 API는 카메라를 끄는 데 사용됩니다.

```
void closeCamera();
```

switchCamera

이 API는 전면/후면 카메라 간에 전환하는 데 사용됩니다.

```
void switchCamera(TUICommonDefine.Camera camera);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
camera	TUICommonDefine.Camera	전면/후면 카메라

openMicrophone

이 API는 마이크를 켜는 데 사용됩니다.

```
void openMicrophone(TUICommonDefine.Callback callback);
```

closeMicrophone

이 API는 마이크를 끄는 데 사용됩니다.

```
void closeMicrophone();
```

selectAudioPlaybackDevice

이 API는 오디오 재생 장치(리시버 또는 스피커)를 선택하는 데 사용됩니다. 통화 시나리오에서 이 API를 사용하여 핸즈프리 모드를 켜고 끌 수 있습니다.

```
void selectAudioPlaybackDevice(TUICommonDefine.AudioPlaybackDevice device);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
device	TUICommonDefine.AudioPlaybackDevice	스피커/수신기

setSelfInfo

이 API는 대화명과 프로필 사진을 설정하는 데 사용됩니다. 대화명은 500바이트를 초과할 수 없으며 프로필 사진은 URL로 지정됩니다.

```
void setSelfInfo(String nickname, String avatar, TUICommonDefine.Callback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
nickname	String	대화명
avatar	String	프로필 사진의 URL

enableMultiDeviceAbility

이 API는 TUICallEngine(프리미엄 패키지에서 지원)에 대한 다중 장치 로그인 활성화 여부를 설정하는 데 사용됩니다.

```
void enableMultiDeviceAbility(boolean enable, TUICommonDefine.Callback callback);
```

TUICallObserver

최종 업데이트 날짜: : 2022-11-24 10:46:14

TUICallObserver API

TUICallObserver는 TUICallEngine의 콜백 클래스입니다. 이벤트를 수신하는 데 사용할 수 있습니다.

개요

API	설명
onError	통화 중 오류 발생
onCallReceived	통화 수신
onCallCancelled	통화 취소
onCallBegin	통화 연결
onCallEnd	통화 종료
onCallMediaTypeChanged	통화 유형 변경
onUserReject	xxxx 사용자 통화 거절
onUserNoResponse	xxxx 사용자가 응답하지 않음
onUserLineBusy	xxxx 사용자 통화 중
onUserJoin	xxxx 사용자가 통화에 참여함
onUserLeave	xxxx 사용자가 통화를 종료함
onUserVideoAvailable	xxx 사용자에게 비디오 스트림이 있는지 여부
onUserAudioAvailable	xxx 사용자에게 오디오 스트림이 있는지 여부
onUserVoiceVolumeChanged	모든 사용자의 볼륨 수준
onUserNetworkQualityChanged	모든 사용자의 네트워크 품질

세부 정보

onError

오류가 발생했습니다.

설명 :

이 콜백은 SDK에 복구할 수 없는 오류가 발생했음을 나타냅니다. 이러한 오류를 수신해야 하며 필요한 경우 UI 알림을 사용자에게 보내야 합니다.

```
void onError(int code, String msg);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
code	int	오류 코드
msg	String	오류 정보

onCallReceived

통화 초대가 수신되었습니다. 이 콜백은 초대받은 사람이 받게됩니다. 이 이벤트를 수신하여 수신 통화 보기 표시 여부를 결정할 수 있습니다.

```
void onCallReceived(String callerId, List<String> calleeIdList, String groupId, TUICallDefine.MediaType callMediaType);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callerId	String	초대자의 사용자 ID
calleeIdList	List	초대받은 사람 목록
groupId	String	그룹 ID
callMediaType	TUICallDefine.MediaType	비디오 또는 오디오일 수 있는 통화 유형

onCallCancelled

초대한 사람이 통화를 취소했거나 시간이 초과되었습니다. 이 콜백은 초대 받은 사람이 받게 됩니다. 이 이벤트를 수신하여 부재 중 전화 메시지를 표시할지 여부를 결정할 수 있습니다.

```
void onCallCancelled(String callerId);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callerId	String	취소 사용자 ID

onCallBegin

통화가 연결되었습니다. 이 콜백은 초대자와 초대받은 사람 모두에게 수신됩니다. 이 이벤트를 수신하여 클라우드 녹화, 콘텐츠 조정 또는 기타 작업 시작 여부를 결정할 수 있습니다.

```
void onCallBegin(TUICommonDefine.RoomId roomId, TUICallDefine.MediaType callMediaType, TUICallDefine.Role callRole);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	TUICommonDefine.RoomId	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정입니다
callMediaType	TUICallDefine.MediaType	비디오 또는 오디오일 수 있는 통화 유형
callRole	TUICallDefine.Role	호출자 또는 호출 수신자가 될 수 있는 역할

onCallEnd

통화가 종료되었습니다. 이 콜백은 초대자와 초대 받은 사람 모두에게 수신됩니다. 이 이벤트를 수신하여 통화 시간 및 통화 유형과 같은 통화 정보를 표시하거나 클라우드 녹음을 중지할 시기를 결정할 수 있습니다.

```
void onCallEnd(TUICommonDefine.RoomId roomId, TUICallDefine.MediaType callMediaType, TUICallDefine.Role callRole, long totalTime);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	TUICommonDefine.RoomId	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정입니다

매개변수	유형	의미
callMediaType	TUICallDefine.MediaType	비디오 또는 오디오일 수 있는 통화 유형
callRole	TUICallDefine.Role	호출자 또는 호출 수신자가 될 수 있는 역할
totalTime	long	통화 시간

주의 :

예를 들어 프로세스가 닫힐 때와 같이 오류가 발생하면 클라이언트 측 콜백이 손실되는 경우가 많습니다. 과금 등의 목적으로 통화 시간을 측정해야 하는 경우 REST API를 사용하는 것이 좋습니다.

onCallMediaTypeChanged

통화 유형이 변경되었습니다.

```
void onCallMediaTypeChanged(TUICallDefine.MediaType oldCallMediaType, TUICallDefine.MediaType newCallMediaType);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
oldCallMediaType	TUICallDefine.MediaType	변경 전 통화 유형
newCallMediaType	TUICallDefine.MediaType	변경 후 통화 유형

onUserReject

통화가 거절되었습니다. 1v1 통화에서는 초대된 사람만 이 콜백을 받습니다. 그룹 통화에서 모든 초대 대상자가 이 콜백을 받습니다.

```
void onUserReject(String userId);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	String	통화를 거절한 초대받은 사람의 사용자 ID

onUserNoResponse

사용자가 응답하지 않습니다.

```
void onUserNoResponse (String userId);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	String	응답하지 않은 초대자의 사용자 ID

onUserLineBusy

사용자가 사용 중입니다.

```
void onUserLineBusy (String userId);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	String	통화 중인 초대자의 사용자 ID

onUserJoin

사용자가 통화에 참여했습니다.

```
void onUserJoin (String userId);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	String	통화에 참여한 사용자의 ID

onUserLeave

사용자가 통화를 종료했습니다.

```
void onUserLeave (String userId);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
------	----	----

매개변수	유형	의미
userId	String	통화 퇴장 사용자 ID

onUserVideoAvailable

사용자가 비디오를 보내고 있는지 여부입니다.

```
void onUserVideoAvailable(String userId, boolean isVideoAvailable);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	String	사용자 ID
isVideoAvailable	boolean	사용자가 비디오를 가지고 있는지 여부

onUserAudioAvailable

사용자가 오디오를 보내고 있는지 여부입니다.

```
void onUserAudioAvailable(String userId, boolean isAudioAvailable);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	String	사용자 ID
isAudioAvailable	boolean	사용자에게 오디오가 있는지 여부

onUserVoiceVolumeChanged

모든 사용자의 볼륨입니다.

```
void onUserVoiceVolumeChanged(Map<String, Integer> volumeMap);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
volumeMap	Map< String, Integer>	각 사용자의 볼륨을 포함하는 볼륨 테이블(<code>userId</code>), 값 범위: 0-100

onUserNetworkQualityChanged

모든 사용자의 네트워크 품질입니다.

```
void onUserNetworkQualityChanged(List<TUICallDefine.NetworkQualityInfo> networkQualityList);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
networkQualityList	List	모든 사용자의 현재 네트워크 상태(userId)

iOS

API 개요

최종 업데이트 날짜: : 2022-11-21 16:34:36

TUICallKit (UI 포함)

TUICallKit은 **UI 요소를 포함**하는 음성/영상 통화 컴포넌트입니다. API를 사용하여 WeChat과 유사한 음성/영상 통화 애플리케이션을 빠르게 구현할 수 있습니다.

API	설명
createInstance	TUICallKit 인스턴스 생성(싱글톤 모드)
setSelfInfo	사용자 대화명과 프로필 사진 설정
call	1v1 통화 호출
groupCall	그룹 통화를 함
joinInGroupCall	그룹 통화에 참여함
setCallingBell	벨소리를 설정함
enableMuteMode	음소거 모드 활성화 여부를 설정함
enableFloatWindow	플로팅 창 활성화 여부를 설정함

TUICallEngine (UI 없음)

TUICallEngine은 **UI 요소를 포함하지 않는** 오디오/비디오 통화 컴포넌트입니다. TUICallKit이 요구 사항을 충족하지 않는 경우 TUICallEngine의 API를 사용하여 프로젝트를 사용자 정의할 수 있습니다.

API	설명
createInstance	TUICallEngine 인스턴스 생성(싱글톤 모드)
destroyInstance	TUICallEngine 인스턴스 종료(싱글톤)
init	기본 음성/영상 통화 기능 인증

API	설명
addObserver	이벤트 리스너 등록
removeObserver	이벤트 리스너 등록 취소
call	1v1 전화 걸기
groupCall	그룹 통화 하기
accept	통화에 응답
reject	통화 거절
hangup	통화 종료
ignore	통화 무시
inviteUser	현재 그룹 통화에 사용자 초대
joinInGroupCall	그룹 통화에 참여
switchCallMediaType	영상 통화에서 음성 통화로의 통화 유형 변경
setRenderView	비디오를 표시하도록 View 객체 설정
startRemoteView	비디오를 표시하도록 View 객체 설정
stopRemoteView	원격 사용자의 비디오 스트림 구독 취소
openCamera	카메라 켜기
closeCamera	카메라 끄기
switchCamera	전면 및 후면 카메라 전환
openMicrophone	마이크 활성화
closeMicrophone	마이크 비활성화
selectAudioPlaybackDevice	오디오 재생 장치 선택(핸드셋/핸즈프리)
setSelfInfo	대화명 및 프로필 사진 설정
enableMultiDeviceAbility	TUICallEngine 에 대한 다중 장치 로그인 활성화 여부 설정(프리미엄 패키지에 서 지원)

TUICallObserver

TUICallObserver는 TUICallEngine의 콜백 클래스입니다. 이벤트를 수신하는 데 사용할 수 있습니다.

API	설명
onError	통화하는 동안 오류 발생
onCallReceived	통화가 수신됨
onCallCancelled	통화가 취소됨
onCallBegin	통화가 연결됨
onCallEnd	통화 종료
onCallMediaTypeChanged	통화 유형이 변경됨
onUserReject	xxxx 사용자가 통화를 거부함
onUserNoResponse	xxxx 사용자가 응답하지 않음
onUserLineBusy	xxxx 사용자가 통화 중
onUserJoin	xxxx 사용자가 통화에 참여함
onUserLeave	xxxx 사용자가 통화를 종료함
onUserVideoAvailable	xxx 사용자가 비디오 스트림을 가지고 있는지 여부
onUserAudioAvailable	xxx 사용자에게 오디오 스트림이 있는지 여부
onUserVoiceVolumeChanged	모든 사용자의 볼륨 수준
onUserNetworkQualityChanged	모든 사용자의 네트워크 품질

주요 클래스의 정의

API	설명
TUICallMediaType	통화 유형. 열거: 영상 통화 및 음성 통화
TUICallRole	통화 역할. 열거: 호출자 및 호출 수신자
TUICallStatus	통화 상태. 열거: 유휴, 대기 중 및 응답 중

API	설명
TUIRoomId	방 Id. 숫자 또는 문자열
TUICallCamera	카메라 유형. 열거: 전면 카메라 및 후면 카메라
TUIAudioPlaybackDevice	오디오 재생 장치 유형. 열거: 스피커 및 수신기
TUINetworkQualityInfo	현재 네트워크 품질

TUICallKit

최종 업데이트 날짜: : 2022-11-21 16:34:36

TUICallKit API

TUICallKit은 **UI 요소를 포함**하는 음성/영상 통화 컴포넌트입니다. API를 사용하여 WeChat과 유사한 음성/영상 통화 애플리케이션을 빠르게 구현할 수 있습니다. 통합에 대한 지침은 [TUICallKit 통합](#)을 참고하십시오.

API 개요

API	설명
createInstance	TUICallKit 인스턴스 생성(싱글톤 모드)
setSelfInfo	대화명 및 프로필 사진 설정
call	1v1 전화 걸기
groupCall	그룹 전화 걸기
joinInGroupCall	그룹 통화에 참여
setCallingBell	벨소리 설정
enableMuteMode	음소거 모드를 켜지 여부 설정
enableFloatWindow	플로팅 창 활성화 여부 설정

API 세부 정보

createInstance

이 API는 TUICallKit 싱글톤을 생성하는 데 사용됩니다.

```
- (instancetype) createInstance;
```

setSelfInfo

이 API는 대화명과 프로필 사진을 설정하는 데 사용됩니다. 대화명은 500바이트를 초과할 수 없으며 프로필 사진은 URL로 지정됩니다.

```
- (void)setSelfInfo:(NSString * _Nullable)nickname avatar:(NSString * _Nullable)avatar succ:(TUICallSucc)succ fail:(TUICallFail)fail
```

매개변수	유형	의미
nickName	NSString	대화명
avatar	NSString	프로필 사진

call

이 API는 (1v1) 통화를 수행하는 데 사용됩니다.

```
- (void)call:(NSString *)userId callMediaType:(TUICallMediaType)callMediaType;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	NSString	대상 userId
callMediaType	TUICallMediaType	비디오 또는 오디오일 수 있는 통화 유형

groupCall

이 API는 그룹 통화를 하는 데 사용됩니다. 그룹 통화를 하기 전에 IM 그룹을 생성해야 합니다.

```
- (void)groupCall:(NSString *)groupId userIdList:(NSArray<NSString *> *)userIdList callMediaType:(TUICallMediaType)callMediaType;
```

매개변수	유형	의미
groupId	NSString	그룹 ID
userIdList	NSArray	대상 userId
callMediaType	TUICallMediaType	비디오 또는 오디오일 수 있는 통화 유형

joinInGroupCall

이 API는 그룹 통화를 하는 데 사용됩니다. 그룹 통화를 하기 전에 IM 그룹을 생성해야 합니다.

```
- (void)joinInGroupCall:(TUIRoomId *)roomId groupId:(NSString *)groupId callMedia
Type:(TUICallMediaType)callMediaType;
```

매개변수	유형	의미
roomId	TUIRoomId	방 ID, 현재는 숫자 방 ID만 사용할 수 있으며, 앞으로는 문자열 형식의 방 ID가 지원될 예정입니다
groupId	NSString	그룹 ID
callMediaType	TUICallMediaType	비디오 또는 오디오일 수 있는 통화 유형

setCallingBell

이 API는 벨소리를 설정하는 데 사용됩니다. `filePath` 는 액세스 가능한 로컬 파일 URL이어야 합니다.

벨소리는 장치와 연결되어 있으며 사용자에게 따라 변경되지 않습니다.

벨소리를 재설정하려면 `filePath` 에 빈 문자열을 전달하십시오.

```
- (void)setCallingBell:(NSString *)filePath;
```

enableMuteMode

이 API는 음소거 모드를 켜지 여부를 설정하는 데 사용됩니다.

```
- (void)enableMuteMode:(BOOL)enable;
```

enableFloatWindow

이 API는 플로팅 창 활성화 여부를 설정하는 데 사용됩니다. 기본값은 'false'이며 통화 보기의 왼쪽 상단 모서리에 있는 플로팅 창 버튼이 숨겨져 있습니다. true로 설정하면 버튼이 표시됩니다.

```
- (void)enableFloatWindow:(BOOL)enable;
```

TUICallEngine

최종 업데이트 날짜: : 2022-11-21 16:34:36

TUICallEngine API

TUICallEngine은 **UI 요소를 포함하지 않는** 오디오/비디오 통화 컴포넌트입니다. TUICallKit이 요구 사항을 충족하지 않는 경우 TUICallEngine의 API를 사용하여 프로젝트를 사용자 정의할 수 있습니다.

API 개요

API	설명
createInstance	TUICallEngine 인스턴스 생성(싱글톤 모드)
destroyInstance	TUICallEngine 인스턴스 종료(싱글톤 모드)
init	기본 음성/영상 통화 기능을 인증함
addObserver	이벤트 리스너 등록
removeObserver	이벤트 리스너 등록 취소
call	1v1 통화 걸기
groupCall	그룹 통화 걸기
accept	통화 수락
reject	통화 거절
hangup	통화 종료
ignore	통화 무시
inviteUser	현재 그룹 통화에서 사용자 초대
joinInGroupCall	그룹 통화에서 참여
switchCallMediaType	통화 유형 변경. 예를 들어, 화상 통화에서 음성 통화로 전환.
startRemoteView	원격 사용자의 비디오 스트림 구독
stopRemoteView	원격 사용자의 비디오 스트림 구독 취소

API	설명
openCamera	카메라 켜기
closeCamera	카메라 끄기
switchCamera	전면 및 후면 카메라 전환
openMicrophone	마이크 켜기
closeMicrophone	마이크 끄기
selectAudioPlaybackDevice	오디오 재생 장치(핸드셋 또는 스피커) 선택
setSelfInfo	대화명 및 프로필 사진 설정
enableMultiDeviceAbility	TUICallEngine에 대한 다중 장치 로그인 활성화 여부 설정(프리미엄 패키지에서 지원)

API 세부 정보

createInstance

이 API는 TUICallEngine 싱글톤을 생성하는 데 사용됩니다.

```
- (TUICallEngine *)createInstance;
```

destroyInstance

이 API는 TUICallEngine 싱글톤을 종료하는 데 사용됩니다.

```
- (void)destroyInstance;
```

init

이 API는 TUICallEngine을 초기화하는 데 사용됩니다. 다른 API를 호출하기 전에 통화 서비스를 인증하고 다른 필수 작업을 수행하기 위해 호출합니다.

```
- (void)init:(NSString *)sdkAppID userId:(NSString *)userId userSig:(NSString *)userSig succ:(TUICallSucc)succ fail:(TUICallFail)fail;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
sdkAppID	int	TRTC 콘솔 > 애플리케이션 관리 > 애플리케이션 정보 에서 SDKAppID 확인
userId	String	현재 사용자의 ID, 문자열 유형은 영어 알파벳(a-z, A-Z), 숫자(0-9), 하이픈(-), 언더바(_)만 허용
userSig	String	Tencent Cloud의 독점 보안 서명. 계산 및 사용 방법은 UserSig 참고
callback	TUIDefine.Callback	초기화 콜백. <code>onSuccess</code> 는 초기화가 성공했음을 표시

addObserver

이 API는 'TUICallObserver' 이벤트를 수신할 이벤트 리스너를 등록하는 데 사용됩니다.

```
- (void) addObserver: (id<TUICallObserver>) observer;
```

removeObserver

이 API는 이벤트 리스너를 등록 해제하는 데 사용됩니다.

```
- (void) removeObserver: (id<TUICallObserver>) observer;
```

call

이 API는 (1v1) 통화를 수행하는 데 사용됩니다.

```
- (void) call: (TUIRoomId *) roomId userId: (NSString *) userId callMediaType: (TUICallMedia
Type) callMediaType params: (TUICallParams *) params succ: (TUICallSucc) succ fail: (TUICallFail) fail
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	TUIRoomId	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정입니다
userId	NSString	대상 userId
callMediaType	TUICallMediaType	비디오 또는 오디오일 수 있는 통화 유형
params	TUICallParams	추가 매개변수. 예를 들어 오프라인 알림을 사용자 정의하는 데 사용할 수 있음

groupCall

이 API는 그룹 통화를 하는 데 사용됩니다. 그룹 통화를 하기 전에 IM 그룹을 생성해야 합니다.

```
- (void)groupCall:(TUIRoomId *)roomId groupId:(NSString *)groupId userIdList:(NSArray <NSString *> *)userIdList callMediaType:(TUICallMediaType)callMediaType params:(TUICallParams *)params succ:(TUICallSucc)succ fail:(TUICallFail)fail
```

매개변수	유형	의미
roomId	TUIRoomId	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정입니다
groupId	NSString	그룹 ID
userIdList	NSArray	대상 userId
callMediaType	TUICallMediaType	비디오 또는 오디오일 수 있는 통화 유형
params	TUICallParams	추가 매개변수. 예를 들어 오프라인 알림을 사용자 지정하는 데 사용할 수 있음

accept

이 API는 통화를 수락하는 데 사용됩니다. 'onCallReceived()' 콜백을 수신한 후 이 API를 호출하여 통화를 수락할 수 있습니다.

```
- (void)accept:(TUICallSucc)succ fail:(TUICallFail)fail;
```

reject

이 API는 통화를 거절하는 데 사용됩니다. 'onCallReceived()' 콜백을 수신한 후 이 API를 호출하여 통화를 거절할 수 있습니다.

```
- (void)reject:(TUICallSucc)succ fail:(TUICallFail)fail;
```

ignore

이 API는 통화를 무시하는 데 사용됩니다. 'onCallReceived()'를 수신한 후 이 API를 호출하여 통화를 무시할 수 있습니다. 호출자는 'onUserLineBusy' 콜백을 수신합니다. 프로젝트에 라이브 스트리밍 또는 회의가 포함된 경우 이 API를 사용하여 '회의 중' 또는 '방송 중' 기능을 구현할 수도 있습니다.

```
- (void)ignore:(TUICallSucc)succ fail:(TUICallFail)fail;
```


hangup

이 API는 통화를 종료하는 데 사용됩니다.

```
- (void)hangup:(TUICallSucc)succ fail:(TUICallFail)fail;
```

inviteUser

이 API는 그룹 통화 참가자가 사용자를 통화로 초대하는 데 사용됩니다.

```
- (void)inviteUser:(NSArray<NSString *> *)userIdList params:(TUICallParams *)params succ:(void (^)(NSArray <NSString *> *userIdList))succ fail:(TUICallFail)fail
```

매개변수	유형	의미
userIdList	NSArray	대상 userId
params	TUICallParams	추가 매개변수. 예를 들어 오프라인 알림을 사용자 지정하는 데 사용할 수 있음.

joinInGroupCall

이 API는 그룹 구성원이 그룹 통화로 참여하는 데 사용됩니다.

```
- (void)joinInGroupCall:(TUIRoomId *)roomId groupId:(NSString *)groupId callMediaType:(TUICallMediaType)callMediaType succ:(TUICallSucc)succ fail:(TUICallFail)fail;
```

매개변수	유형	의미
roomId	TUIRoomId	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정입니다.
groupId	NSString	그룹 ID
callMediaType	TUICallMediaType	비디오 또는 오디오일 수 있는 통화 유형

switchCallMediaType

이 API는 그룹 구성원이 그룹 통화로 참여하는 데 사용됩니다.

```
- (void)switchCallMediaType:(TUICallMediaType)newType;
```

매개변수	유형	의미
------	----	----

매개변수	유형	의미
callMediaType	TUICallMediaType	비디오 또는 오디오일 수 있는 통화 유형

startRemoteView

이 API는 원격 비디오를 표시하도록 View 객체를 설정하는 데 사용됩니다.

```
- (void)startRemoteView:(NSString *)userId videoView:(TUIVideoView *)videoView onPlaying:(void (^)(NSString *userId))onPlaying onLoading:(void (^)(NSString *userId))onLoading onError:(void (^)(NSString *userId, int code, NSString *errMsg))onError;
```

매개변수	유형	의미
userId	NSString	대상 userId
videoView	TUIVideoView	렌더링할 뷰

stopRemoteview

이 API는 원격 사용자의 비디오 스트림 구독을 취소하는 데 사용됩니다.

```
- (void)stopRemoteView:(NSString *)userId;
```

매개변수	유형	의미
userId	NSString	대상 userId

openCamera

카메라 활성화

```
- (void)openCamera:(TUICallCamera)camera videoView:(TUIVideoView *)videoView success:(TUICallSucc)succ fail:(TUICallFail)fail;
```

매개변수	유형	의미
camera	TUICallCamera	전면 또는 후면 카메라 사용 여부
videoView	TUIVideoView	렌더링할 뷰

closeCamera

이 API는 카메라를 끄는 데 사용됩니다.

```
- (void)closeCamera;
```

switchCamera

이 API는 전면 및 후면 카메라 간에 전환하는 데 사용됩니다.

```
- (void)switchCamera:(TUICallCamera) camera;
```

매개변수	유형	의미
camera	TUICallCamera	전면 또는 후면 카메라 사용 여부

openMicrophone

이 API는 마이크를 켜는 데 사용됩니다.

```
- (void)openMicrophone:(TUICallSucc) succ fail:(TUICallFail) fail;
```

closeMicrophone

이 API는 마이크를 끄는 데 사용됩니다.

```
- (void)closeMicrophone;
```

selectAudioPlaybackDevice

이 API는 오디오 재생 장치(리시버 또는 스피커)를 선택하는 데 사용됩니다. 통화 시나리오에서 이 API를 사용하여 핸즈프리 모드를 켜고 끌 수 있습니다.

```
- (void)selectAudioPlaybackDevice:(TUIAudioPlaybackDevice) device;
```

매개변수	유형	의미
device	TUIAudioPlaybackDevice	스피커 또는 핸드셋

setSelfInfo

이 API는 대화명과 프로필 사진을 설정하는 데 사용됩니다. 대화명은 500바이트를 초과할 수 없으며 프로필 사진은 URL로 지정됩니다.

```
- (void)setSelfInfo:(NSString * _Nullable)nickName avatar:(NSString * _Nullable)avatar succ:(TUICallSucc)succ fail:(TUICallFail)fail;
```

enableMultiDeviceAbility

이 API는 `TUICallEngine` 에 대한 다중 장치 로그인 활성화 여부를 설정하는 데 사용됩니다(프리미엄 패키지에
서 지원).

```
- (void)enableMultiDeviceAbility:(BOOL)enable succ:(TUICallSucc)succ fail:(TUICallFail)fail;
```

TUICallObserver

최종 업데이트 날짜: : 2022-11-21 16:34:36

TUICallObserver API

TUICallObserver는 TUICallEngine의 콜백 클래스입니다. 이벤트를 수신하는 데 사용할 수 있습니다.

개요

API	설명
onError	통화 중 오류 발생
onCallReceived	통화 초대를 받음
onCallCancelled	통화가 취소됨
onCallBegin	통화가 연결됨
onCallEnd	통화가 종료됨
onCallMediaTypeChanged	통화 유형이 변경됨
onUserReject	xxxx 사용자가 통화를 거절함
onUserNoResponse	xxxx 사용자가 응답하지 않음
onUserLineBusy	xxxx 사용자가 통화 중임
onUserJoin	xxxx 사용자가 통화에 참여함
onUserLeave	xxxx 사용자가 통화를 종료함
onUserVideoAvailable	xxx 사용자가 비디오 스트림이 있는지 여부
onUserAudioAvailable	xxx 사용자에게 오디오 스트림이 있는지 여부
onUserVoiceVolumeChanged	모든 사용자의 볼륨 수준
onUserNetworkQualityChanged	모든 사용자의 네트워크 품질

세부 정보

onError

오류가 발생했습니다.

설명 :

이 콜백은 SDK에 복구할 수 없는 오류가 발생했음을 나타냅니다. 이러한 오류를 수신해야 하며 필요한 경우 UI 알림을 사용자에게 보내야 합니다.

```
- (void)onError:(int)code message:(NSString * _Nullable)message;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
code	int	오류 코드
message	NSString	오류 정보

onCallReceived

통화 초대가 수신되었습니다. 이 콜백은 초대받은 사람이 받았습니다. 이 이벤트를 수신하여 수신 통화 보기 표시 여부를 결정할 수 있습니다.

```
- (void)onCallReceived:(NSString *)callerId calleeIdList:(NSArray<NSString * > *)calleedIdList groupId:(NSString * _Nullable)groupId callMediaType:(TUICallMediaType)callMediaType
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callerId	NSString	초대자의 사용자 ID
calleedIdList	NSArray	초대받은 사람 목록
groupId	NSString	그룹 ID
callMediaType	TUICallMediaType	비디오 또는 오디오일 수 있는 통화 유형

onCallCancelled

초대한 사람이 통화를 취소했거나 시간이 초과되었습니다. 이 콜백은 초대받은 사람이 받았습니다. 이 이벤트를 수신하여 부재중 전화 메시지를 표시할지 여부를 결정할 수 있습니다.

```
- (void)onCallCancelled:(NSString *)callerId;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callerId	NSString	초대자의 사용자 ID

onCallBegin

통화가 연결되었습니다. 이 콜백은 초대자와 초대받은 사람 모두에게 수신됩니다. 이 이벤트를 수신하여 클라우드 녹화, 콘텐츠 조정 또는 기타 작업 시작 여부를 결정할 수 있습니다.

```
- (void)onCallBegin:(TUIRoomId *)roomId callMediaType:(TUICallMediaType)callMediaType callRole:(TUICallRole)callRole;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	TUIRoomId	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정입니다
callMediaType	TUICallMediaType	비디오 또는 오디오일 수 있는 통화 유형
callRole	TUICallRole	호출자 또는 호출 수신자가 될 수 있는 역할

onCallEnd

통화가 종료되었습니다. 이 콜백은 초대자와 초대받은 사람 모두에게 수신됩니다. 이 이벤트를 수신하여 통화 시간 및 통화 유형과 같은 통화 정보를 표시하거나 클라우드 녹음을 중지할 시기를 결정할 수 있습니다.

```
- (void)onCallEnd:(TUIRoomId *)roomId callMediaType:(TUICallMediaType)callMediaType callRole:(TUICallRole)callRole totalTime:(float)totalTime;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	TUIRoomId	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정입니다

매개변수	유형	의미
callMediaType	TUICallMediaType	비디오 또는 오디오일 수 있는 통화 유형
callRole	TUICallRole	호출자 또는 호출 수신자가 될 수 있는 역할
totalTime	float	통화 시간

주의 :

예를 들어 프로세스가 닫힐 때와 같이 오류가 발생하면 클라이언트 측 콜백이 손실되는 경우가 많습니다. 과금 등의 목적으로 통화 시간을 측정해야 하는 경우 REST API를 사용하는 것이 좋습니다.

onCallMediaTypeChanged

통화 유형이 변경되었습니다.

```
- (void)onCallMediaTypeChanged:(TUICallMediaType)oldCallMediaType newCallMediaType:
(TUICallMediaType)newCallMediaType;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
oldCallMediaType	TUICallMediaType	변경 전의 통화 유형
newCallMediaType	TUICallMediaType	변경 후의 호출 유형

onUserReject

통화가 거절되었습니다. 1v1 통화에서는 초대된 사람만 이 콜백을 받습니다. 그룹 통화에서 모든 초대 대상자는 이 콜백을 받습니다.

```
- (void)onUserReject:(NSString *)userId;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	NSString	통화를 거절한 사용자의 ID

onUserNoResponse

사용자가 응답하지 않았습니다.

```
- (void)onUserNoResponse:(NSString *)userId;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	NSString	응답하지 않은 사용자의 ID

onUserLineBusy

통화 중입니다.

```
- (void)onUserLineBusy:(NSString *)userId;
```

onUserJoin

사용자가 통화에 참여했습니다.

```
- (void)onUserJoin:(NSString *)userId;
```

onUserLeave

사용자가 통화를 종료했습니다.

```
- (void)onUserLeave:(NSString *)userId;
```

onUserAudioAvailable

사용자가 오디오를 보내고 있는지 여부입니다.

```
- (void)onUserAudioAvailable:(NSString *)userId isAudioAvailable:(BOOL)isAudioAvailable;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	NSString	사용자 ID
isAudioAvailable	BOOL	사용자에게 오디오가 있는지 여부

onUserVideoAvailable

사용자가 비디오를 보내고 있는지 여부입니다.

```
- (void)onUserVideoAvailable:(NSString *)userId isVideoAvailable:(BOOL)isVideoAvailable;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	NSString	사용자 ID
isVideoAvailable	BOOL	사용자에게 비디오가 있는지 여부

onUserVoiceVolumeChanged

모든 사용자의 볼륨입니다.

```
- (void)onUserVoiceVolumeChanged:(NSDictionary <NSString *, NSNumber *> *)volumeMap;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
volumeMap	NSDictionary <NSString *, NSNumber *>	각 사용자의 볼륨을 포함하는 볼륨 테이블(userId). 값 범위: 0-100
volumeMap	NSDictionary	각 사용자의 볼륨을 포함하는 볼륨 테이블(userId). 값 범위: 0-100

onUserNetworkQualityChanged

모든 사용자의 네트워크 품질입니다.

```
- (void)onUserNetworkQualityChanged:(NSArray<TUINetworkQualityInfo *> *)networkQualityList;
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
networkQualityList	NSArray	모든 사용자의 현재 네트워크 상태(userId)

Web

API 개요

최종 업데이트 날짜: : 2022-11-21 16:34:37

TUICallEngine (UI 없음)

TUICallEngine은 **UI 요소를 포함하지 않는** 오디오/비디오 통화 컴포넌트입니다. API를 사용하여 프로젝트를 사용자 지정할 수 있습니다.

API	설명
createInstance	TUICallEngine 인스턴스 생성(싱글톤 모드)
destroyInstance	TUICallEngine 인스턴스 종료(싱글톤 모드)
on	이벤트 수신
off	이벤트 수신 중지
login	로그인
logout	로그아웃
setSelfInfo	대화명 및 프로필 사진 설정
call	C2C 통화 걸기
groupCall	그룹 통화 걸기
accept	통화 수락
reject	통화 거절
hangup	통화 종료
switchCallingType	통화 유형 변경
startRemoteView	원격 비디오 렌더링 시작
stopRemoteView	원격 비디오 렌더링 중지
startLocalView	로컬 비디오 렌더링 시작
stopLocalView	로컬 비디오 렌더링 중지

API	설명
openCamera	카메라 켜기
closeCamara	카메라 끄기
openMicrophone	마이크 켜기
closeMicrophone	마이크 끄기
setMicMute	마이크 음소거 여부 설정
setVideoQuality	비디오 품질 설정
getDeviceList	장치 목록 가져오기
switchDevice	다른 카메라/마이크로 변경

이벤트 유형

TUICallEvent는 TUICallEngine의 콜백 클래스입니다. 이벤트를 수신하는 데 사용할 수 있습니다.

EVENT	설명
TUICallEvent.ERROR	SDK 내부 오류 발생
TUICallEvent.SDK_READY	SDK는 ready 상태임
TUICallEvent.KICKED_OUT	반복된 로그인으로 인해 현재 사용자가 방에서 제거 됨
TUICallEvent.USER_ACCEPT	사용자가 전화를 수락함
TUICallEvent.USER_ENTER	사용자가 통화 참여에 동의함
TUICallEvent.USER_LEAVE	사용자가 통화를 종료하는 데 동의함
TUICallEvent.REJECT	사용자가 통화를 거부함
TUICallEvent.NO_RESP	초대된 사용자가 응답하지 않음
TUICallEvent.LINE_BUSY	초대자가 통화 중임
TUICallEvent.CALLING_TIMEOUT	통화 시간 초과(초대 대상자가 수신)
TUICallEvent.USER_VIDEO_AVAILABLE	원격 사용자가 카메라를 켜고 끄

EVENT	설명
TUICallEvent.USER_AUDIO_AVAILABLE	원격 사용자가 마이크 켜기/끄기
TUICallEvent.USER_VOICE_VOLUME	원격 사용자가 통화 볼륨을 조정함
TUICallEvent.GROUP_CALL_INVITEE_LIST_UPDATE	그룹 통화 초대 목록이 업데이트됨
TUICallEvent.INVITED	통화에 초대됨
TUICallEvent.CALLING_CANCEL	통화가 취소됨(초대받은 사람이 수신함)
TUICallEvent.CALLING_END	통화가 종료됨
TUICallEvent.DEVICED_UPDATED	장치 목록이 업데이트됨
TUICallEvent.CALL_TYPE_CHANGED	통화 유형이 변경됨

TUICallKit

최종 업데이트 날짜: : 2023-02-10 18:26:20

API 소개

TUICallKit API는 **UI 요소를 포함**하는 음성/영상 통화 컴포넌트입니다. API를 사용하여 WeChat과 유사한 음성/영상 통화 애플리케이션을 빠르게 구현할 수 있습니다. 통합에 대한 지침은 [TUICallKit 통합](#)을 참고하십시오.

API 개요

`<TUICallKit/>` : 통화 컴포넌트

`<TUICallKitMini/>` : 플로팅 통화 창, `<TUICallKit/>allowedMinimized` 가 `true` 인 경우

`<TUICallKitMini/>` 필요

`TUICallKitServer` : 다음 API가 있는 통화 인스턴스:

`init` TUICallKit 초기화

`call` 1v1 통화 걸기

`groupCall` 그룹 통화 걸기

`destroyed` TUICallKit 종료

`<TUICallKit/>` API 세부 사항

속성

매개변수	설명	유형	필수	기본값
<code>allowedMinimized</code>	창 최소화 허용 여부. <code>true</code> 이면 <code><TUICallKitMini/></code> 가 필요합니다. <code>false</code> 이면 최소화 버튼이 숨겨집니다.	boolean	아니오	false
<code>allowedFullScreen</code>	전체 화면 허용 여부입니다. <code>false</code> 이면 전체 화면 버튼이 숨겨집니다.	boolean	아니오	true

방법

매개변수	설명	유형	필수	기본 값
beforeCalling	사용자가 전화를 걸거나 받을 때 실행됩니다	function(type, error)	아니오	-
afterCalling	통화 종료 후 실행됩니다	function()	아니오	-
onMinimized	컴포넌트가 최소화될 때 실행됩니다	function(oldStatus, newStatus)	아니오	-

<TUICallKitMini/> API 세부 사항

없음

예시 코드



```
/**
 * beforeCalling
 * @param { string } type 유효한 값은 "invited" | "call" | "groupCall", 이를 사용하여
 * @param { number } error.code 에러 코드
 * @param { string } error.type 에러 유형
 * @param { string } error.code 에러 메시지
 */
function beforeCalling(type, error) {
  console.log("이 함수는 통화 시 실행됩니다. 유형: ", type, error);
}
function afterCalling() {
```



```
    console.log("이 함수는 통화 후에 실행됩니다");
}
/**
 * onMinimized
 * @param { boolean } oldStatus
 * @param { boolean } newStatus
 */
function onMinimized(oldStatus, newStatus) {
    if (newStatus === true) {
        console.log("TUICallKit이 최소화되었습니다");
    } else {
        console.log("TUICallKit이 더 이상 최소화되지 않습니다");
    }
}
```

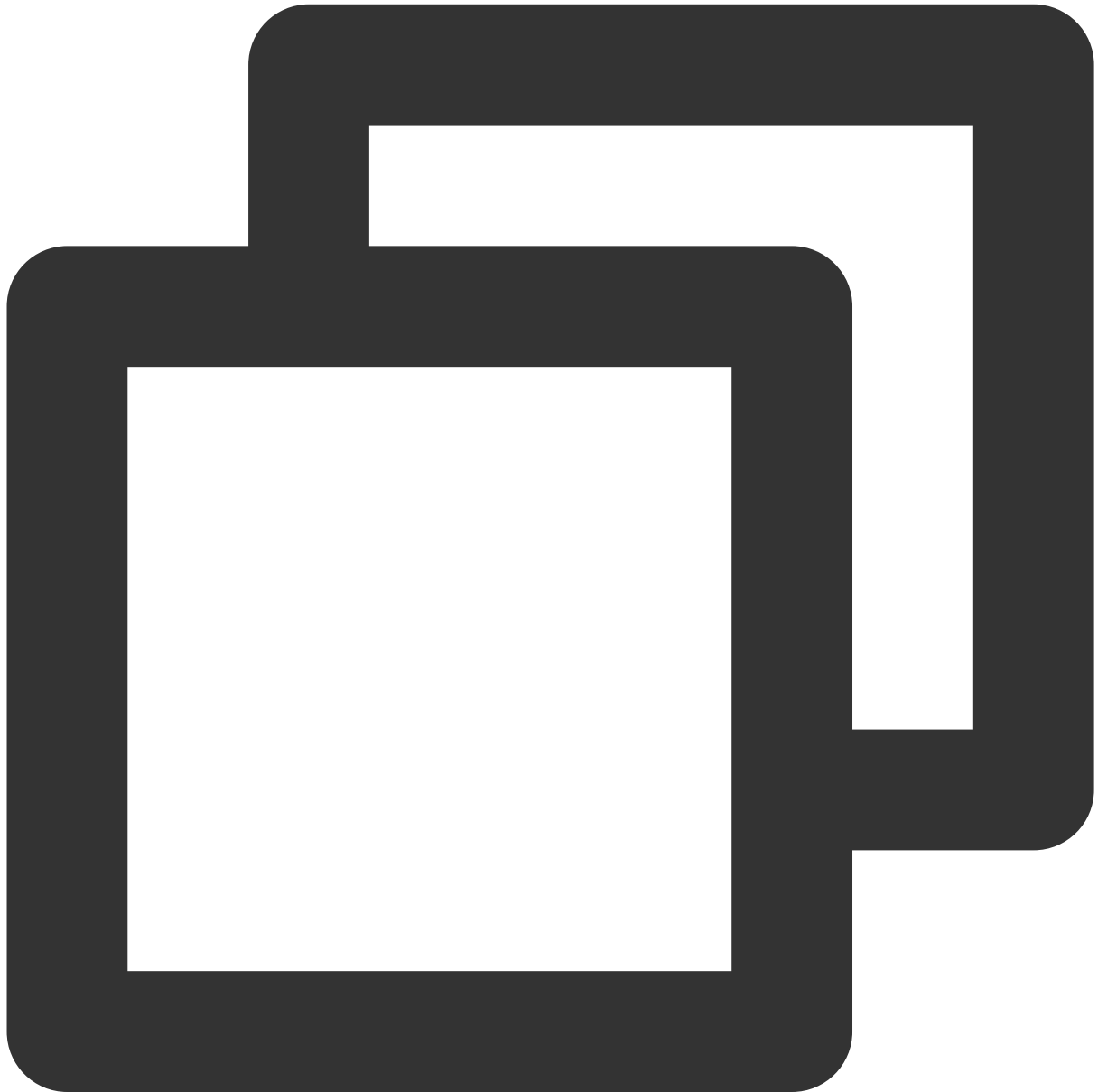


```
<TUICallKit
  :allowedMinimized="true"
  :allowedFullScreen="true"
  :beforeCalling="beforeCalling"
  :afterCalling="afterCalling"
  :onMinimized="onMinimized"
/>
<TUICallKitMini />
```

TUICallKitServer API 세부 사항

init

이 API는 TUICallKit을 초기화하는 데 사용됩니다. call과 groupCall보다 먼저 호출되어야 합니다.



```
import { TUICallKitServer } from "../components/TUICallKit/Web";
TUICallKitServer.init({
  SDKAppID,
  userID,
  userSig,
```

```
tim,
});
```

매개변수는 다음과 같습니다.

매개변수	유형	필수	설명
SDKAppID	Number	예	IM 애플리케이션의 SDKAppID
userID	String	예	현재 사용자 ID, 영어 알파벳(a-z, A-Z), 숫자(0-9), 하이픈(-), 언더바(_)의 문자열만 포함할 수 있습니다
userSig	String	예	Tencent Cloud가 설계한 일종의 보안 서명입니다. 가져오는 방법은 UserSig 계산 방법 을 참고하십시오.
TIM 인스턴스	Any	아니오	이미 TIM 인스턴스가 있는 경우 이 tim 매개변수를 사용하여 TIM 인스턴스의 고유성을 보장할 수 있습니다

call

이 API는 (1v1) 통화를 수행하는 데 사용됩니다.



```
import { TUICallKitServer } from "../components/TUICallKit/Web";
TUICallKitServer.call({
  userID: 'jack',
  type: 1,
});
```

매개변수는 다음과 같습니다.

매개변수	유형	필수	설명
userID	String	예	호출할 userID

type	Number	예	통화 유형, 음성 통화(type = 1), 영상 통화(type = 2)
timeout	Number	아니오	제한 시간(초)(옵션), 0은 호출이 시간 초과되지 않음을 나타냄, 기본값은 30s
offlinePushInfo	Object	아니오	사용자 지정 오프라인 푸시 메시지(옵션), tsignaling 버전이 0.8.0 이상인 경우에만 유효

offlinePushInfo

매개변수	유형	필수	설명
offlinePushInfo.title	String	아니오	메시지 제목(옵션)
offlinePushInfo.description	String	아니오	메시지 내용(옵션)
offlinePushInfo.androidOPPOChannelID	String	아니오	v8.0 이상에서 OPPO 휴대폰의 오프라인 푸시 메시지에 대한 채널 ID(옵션)
offlinePushInfo.extension	String	아니오	통과 콘텐츠(옵션), tsignaling 버전이 0.9.0 이상인 경우에만 유효

groupCall

이 API는 그룹 통화를 하는 데 사용됩니다.



```
import { TUICallKitServer } from "../components/TUICallKit/Web";
TUICallKitServer.groupCall({
  userIDList: ['jack', 'tom'],
  groupID: 'xxx',
  type: 1,
});
```

매개변수는 다음과 같습니다.

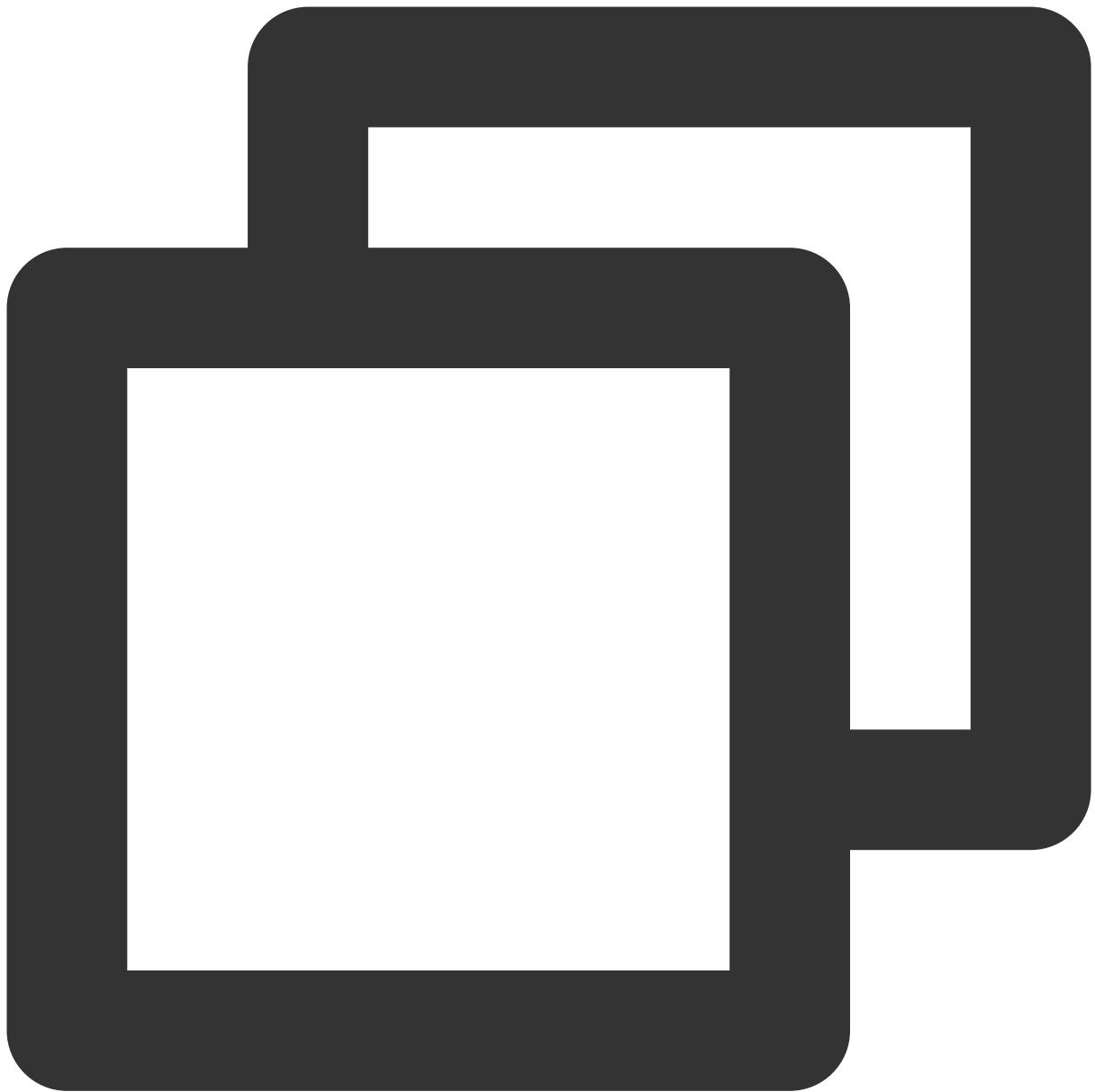
매개변수	유형	필수	설명

userIDList	Array	예	호출할 사용자의 ID
type	Number	예	통화 유형, 음성 통화(type = 1), 영상 통화(type = 2)
groupID	String	예	그룹 ID
timeout	Number	아니 오	제한 시간(초)(옵션), 0은 호출이 시간 초과되지 않음을 나타냄, 기본값은 30s
offlinePushInfo	Object	아니 오	사용자 지정 오프라인 푸시 메시지(옵션), tsignaling 버전이 0.8.0 이상인 경우에만 유효

`offlinePushInfo` 의 매개변수는 위를 참고하십시오.

destroyed

이 API는 TUICallKit을 종료하는 데 사용됩니다.



```
import { TUICallKitServer } from "../components/TUICallKit/Web";  
TUICallKitServer.destroyed();
```

TUICallEngine

최종 업데이트 날짜: : 2023-03-13 16:39:21

TUICallEngine API

TUICallEngine은 **UI 요소를 포함하지 않는** 음성/영상 통화 컴포넌트입니다.

API 개요

API	설명
createInstance	TUICallEngine 인스턴스 생성(싱글톤 모드)
destroyInstance	TUICallEngine 인스턴스 종료(싱글톤 모드)
on	이벤트 수신
off	이벤트 수신 중지
login	로그인
logout	로그아웃
setSelfInfo	대화명 및 프로필 사진 설정
call	C2C 통화 하기
groupCall	그룹 통화하기
accept	통화 수락
reject	통화 거절
hangup	통화 종료
inviteUser	현재 그룹 통화에 사용자를 초대
joinInGroupCall	현재 그룹 통화에 참여
switchCallMediaType	통화 유형 변경
startRemoteView	원격 비디오 렌더링 시작

<code>stopRemoteView</code>	원격 비디오 렌더링 중지
<code>startLocalView</code>	로컬 비디오 렌더링 시작
<code>stopLocalView</code>	로컬 비디오 렌더링 중지
<code>openCamera</code>	카메라 켜기
<code>closeCamara</code>	카메라 끄기
<code>openMicrophone</code>	마이크 켜기
<code>closeMicrophone</code>	마이크 끄기
<code>setVideoQuality</code>	비디오 품질 설정
<code>getDeviceList</code>	장치 목록 가져오기
<code>switchDevice</code>	다른 카메라/마이크로 변경
<code>enableAIvoice</code>	AI 노이즈 캔슬링 활성화/비활성화

API 세부 사항

createInstance

이 API는 TUICallEngine 싱글톤을 생성하는 데 사용됩니다.



```
const tuiCallEngine = TUICallEngine.createInstance({
  SDKAppID: 0, // 0을 IM 애플리케이션의 SDKAppID로 대체
  tim: tim      // 이미 TIM 인스턴스가 있는 경우 이 tim 매개변수를 사용하여 TIM 인스턴
});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명
SDKAppID	Number	IM 애플리케이션의 SDKAppID

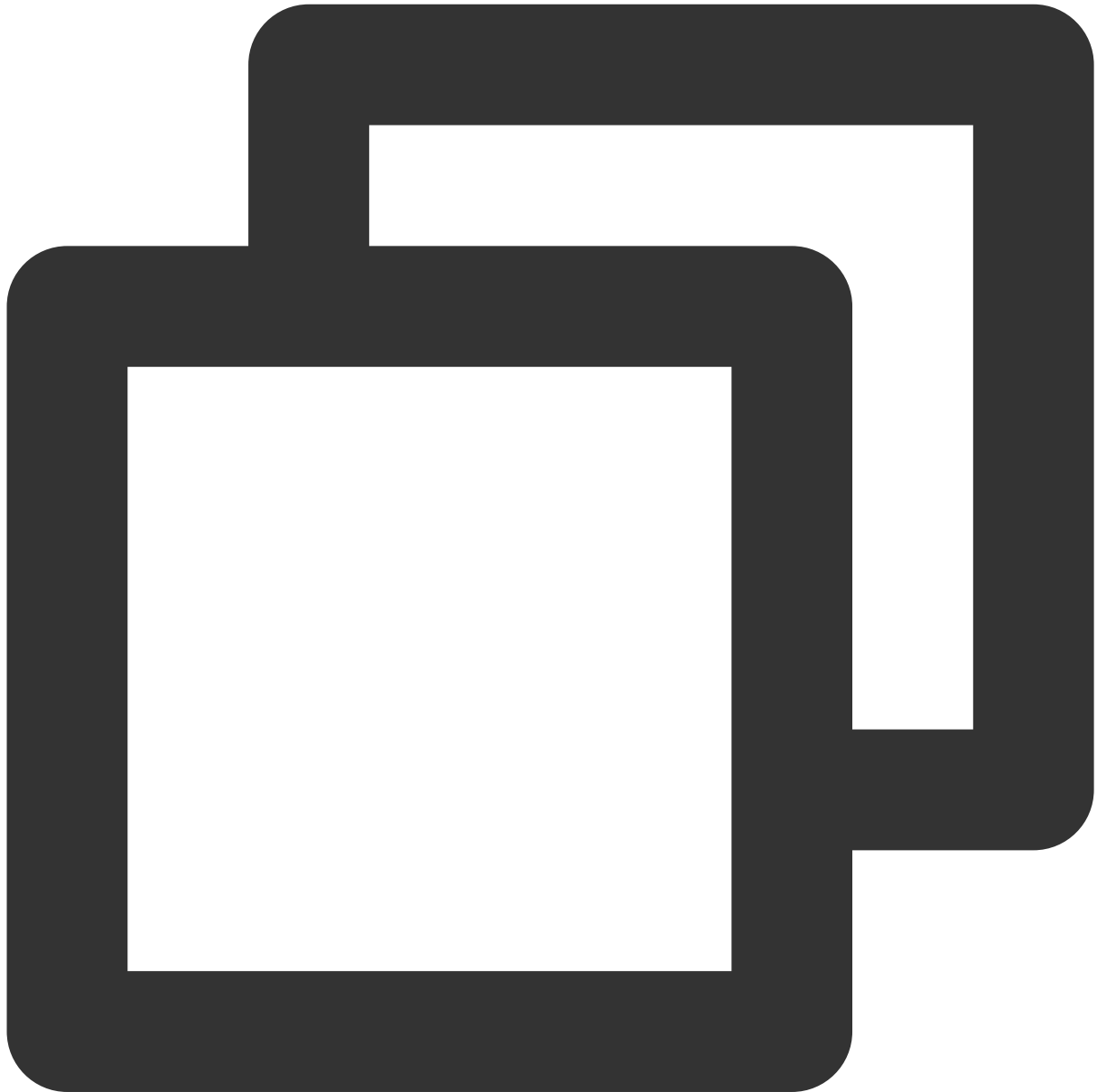
tim

Any

TIM 인스턴스(선택 사항)

destroyInstance

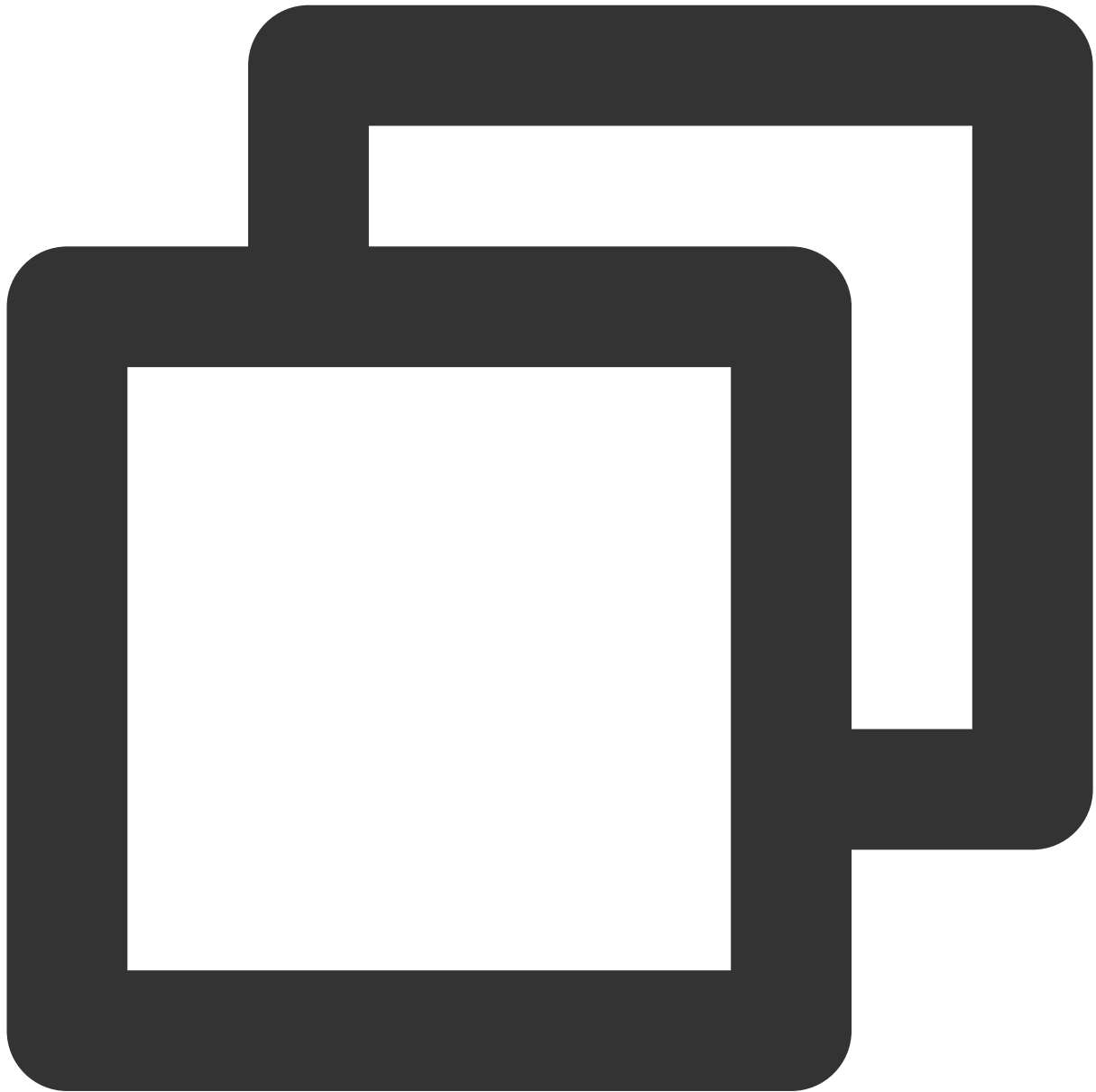
이 API는 TUICallEngine 싱글톤을 종료하는 데 사용됩니다.



```
tuiCallEngine.destroyInstance().then(() => {  
  //success  
}).catch(error => {  
  console.warn('destroyInstance error:', error);  
});
```

on

이 API는 이벤트를 수신하는 데 사용됩니다.



```
let onError = function(error) {  
    console.log(error);  
};  
tuiCallEngine.on(TUICallEvent.ERROR, onError, this);
```

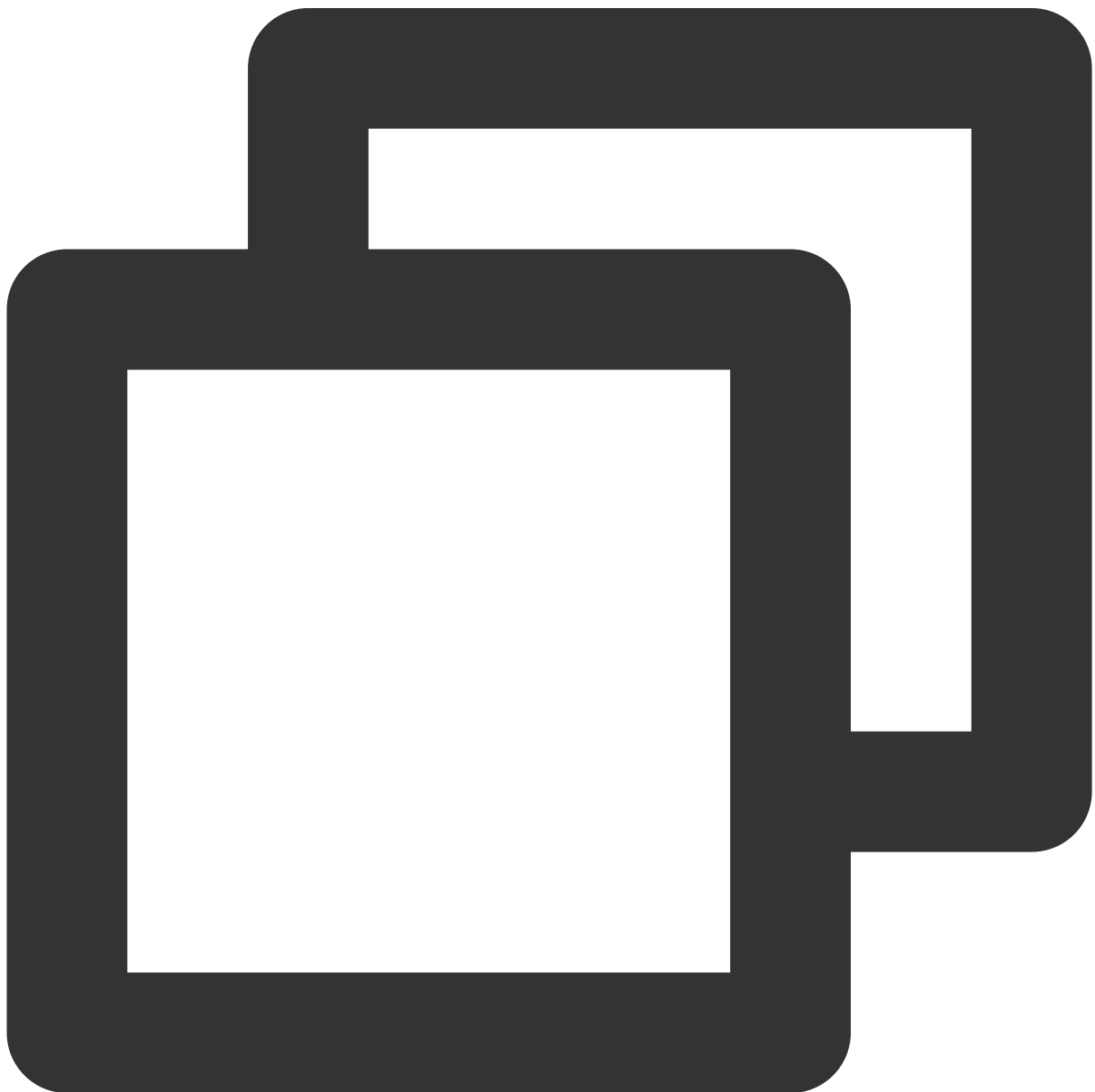
매개변수는 다음과 같습니다:

--	--	--

매개변수	유형	설명
eventName	String	이벤트 이름
callback	function	이벤트 콜백
context	Any	컨텍스트

off

이 API는 이벤트 수신을 중지하는 데 사용됩니다.



```
let onError = function(error) {
    console.log(error);
};
tuiCallEngine.off(TUICallEvent.ERROR, onError, this);
```

매개변수는 다음과 같습니다:

매개변수	유형	설명
eventName	String	이벤트 이름
callback	function	이벤트 콜백
context	Any	컨텍스트

login

이 API는 로그인에 사용됩니다.



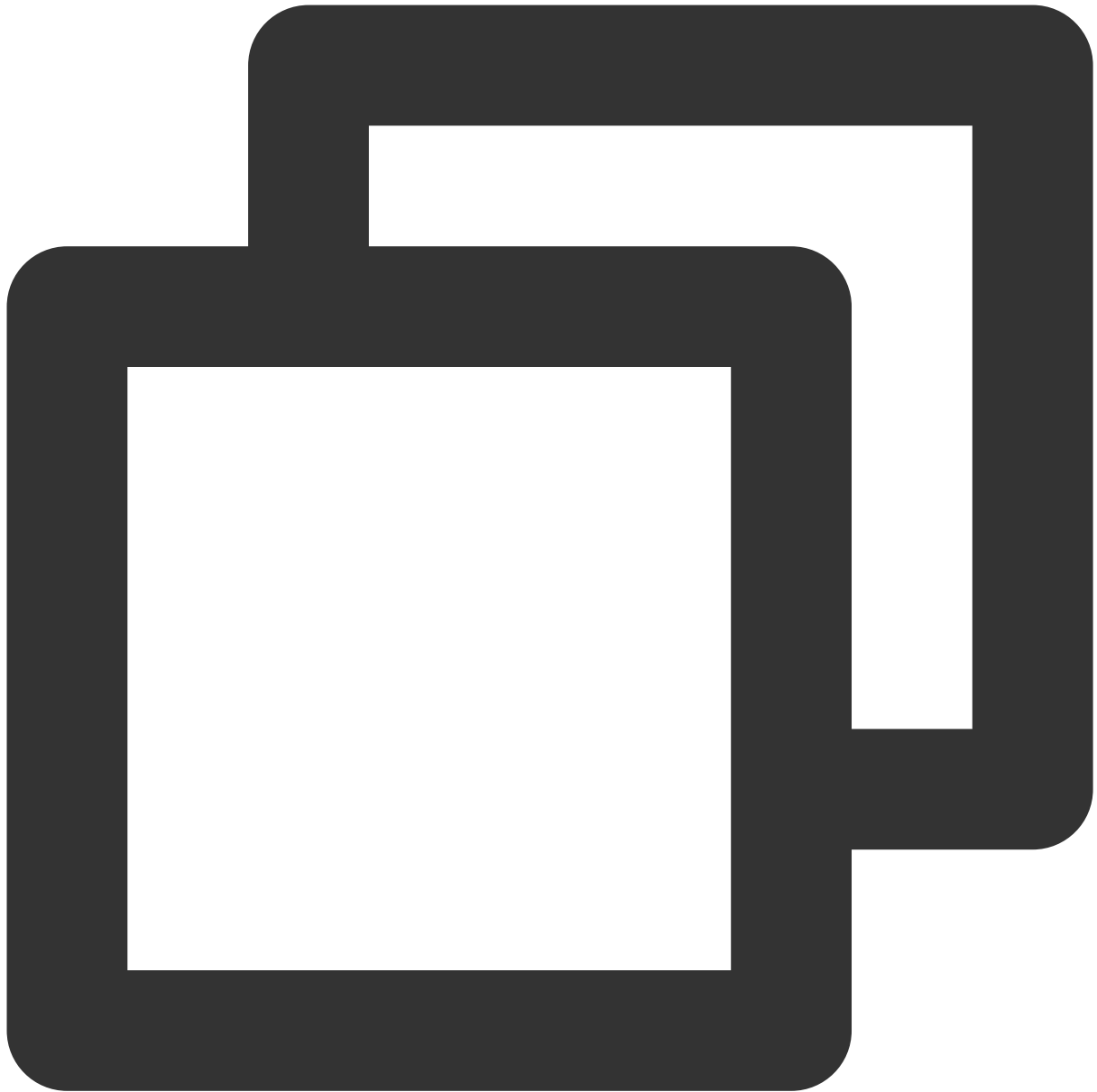
```
const params = {
  userID: 'john', // your userID
  userSig: 'xxxx', // 'your userSig'
  assetsPath: 'https://xx/'
};
let promise = tuiCallEngine.login(params);
promise.then(() => {
  //success
}).catch(error => {
  console.warn('login error:', error);
});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명
userID	String	현재 사용자 ID이며, 문자열 유형은 영어 알파벳(a-z, A-Z), 숫자(0-9), 대시부호(-), 언더바(_)만 허용됩니다
userSig	String	Tencent Cloud의 독점 보안 서명. 자세한 내용은 UserSig 계산 방법 을 참고하십시오.
assetsPath	String	denoiser-wasm.js 파일이 배포된 CDN 또는 정적 리소스 서버의 경로 이 파일은 AI 노이즈 캔슬링에 필요합니다. 자세한 내용은 튜토리얼: AI 노이즈 캔슬링 활성화 를 참고하십시오.

logout

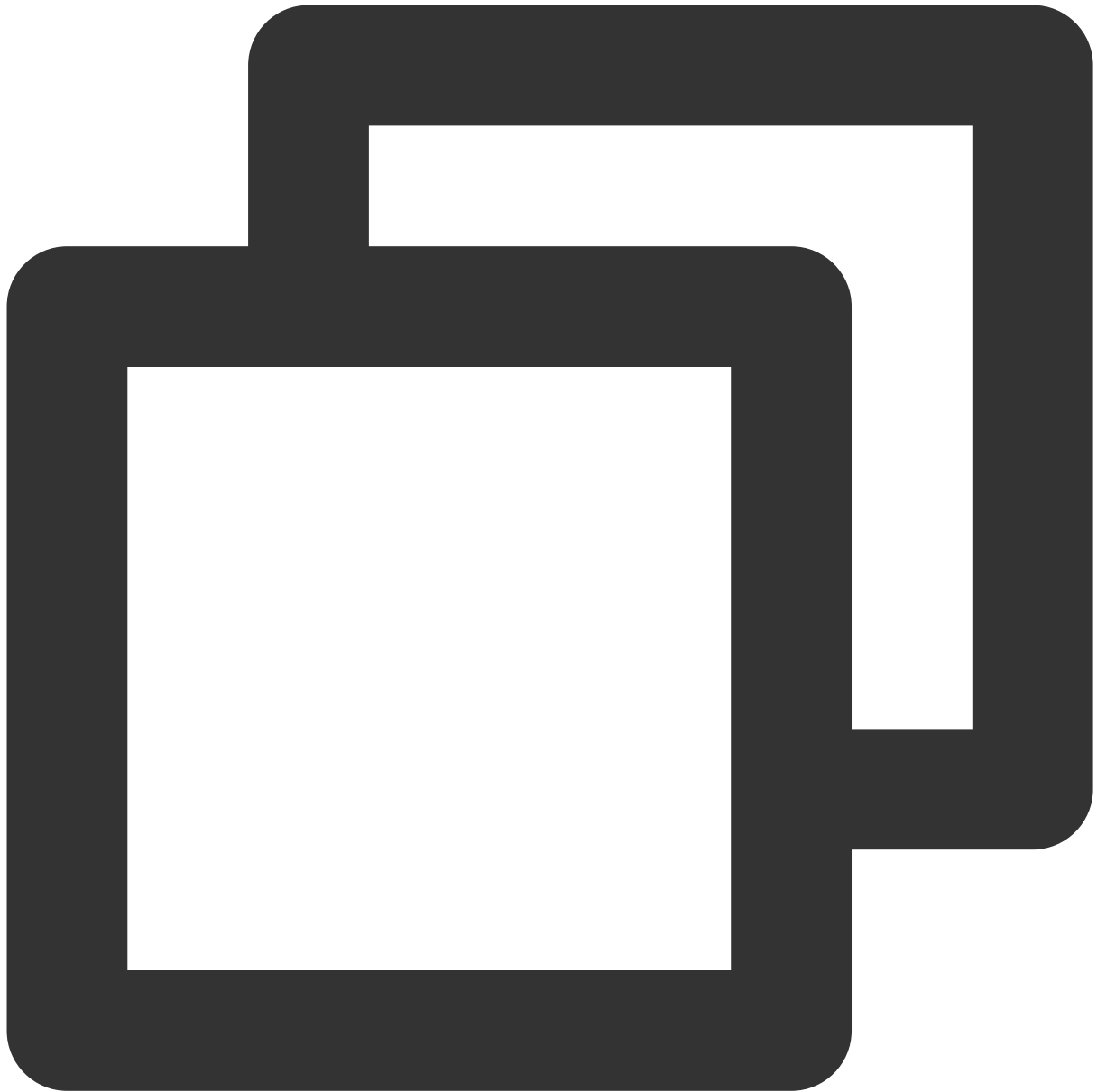
이 API는 로그아웃하는 데 사용됩니다.



```
let promise = tuiCallEngine.logout();
promise.then(() => {
  //success
}).catch(error => {
  console.warn('logout error:', error);
});
```

setSelfInfo

이 API는 대화명 및 프로필 사진을 설정하는 데 사용됩니다.



```
let promise = tuiCallEngine.setSelfInfo({
  nickName: 'video',
  avatar: 'http(s)://url/to/image.jpg'
});
promise.then(() => {
  //success
}).catch(error => {
  console.warn('setSelfInfo error:', error);
});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명
nickName	String	대화명
avatar	String	프로필 사진 URL

call

이 API는 C2C 통화를 수행하는 데 사용됩니다. 초대받은 사람은 `TUICallEvent.INVITED` 콜백을 수신합니다.

참고

Android 및 iOS에서 오프라인 알림이 지원됩니다. Web 및 WeChat 미니프로그램에서는 지원되지 않습니다.



```
let promise = tuiCallEngine.call({
  userID: 'user1',
  type: 1,
});
promise.then(() => {
  //success
}).catch(error => {
  console.warn('call error:', error);
});
```

매개변수는 다음과 같습니다:

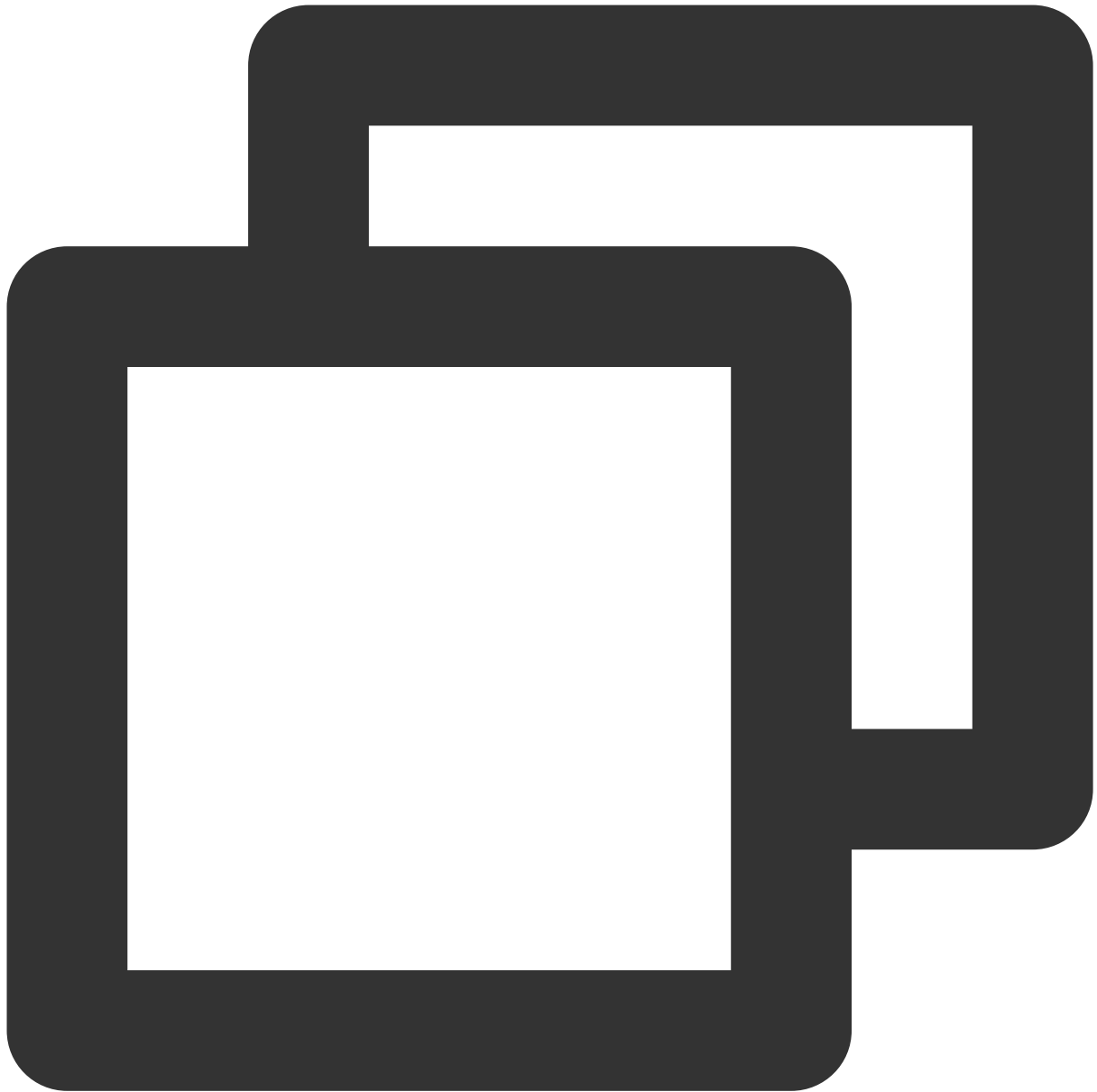
매개변수	유형	설명
userID	String	초대받은 사람의 userID
type	Number	통화 유형. 0-알 수 없음, 1-음성 통화, 2-영상 통화

groupCall

이 API는 그룹 통화를 하는 데 사용됩니다. 초대받은 사람은 'EVENT.INVITED' 콜백을 받게 됩니다.

참고

Android 및 iOS에서 오프라인 알림이 지원됩니다. Web 및 WeChat 미니프로그램에서는 지원되지 않습니다.



```
let promise = tuiCallEngine.groupCall({
  userIDList: ['user1', 'user2'],
  type: 1,
  groupID: 'IM 그룹 ID',
});
promise.then(() => {
  //success
}).catch(error => {
  console.warn('groupCall error:', error);
});
```


매개변수는 다음과 같습니다:

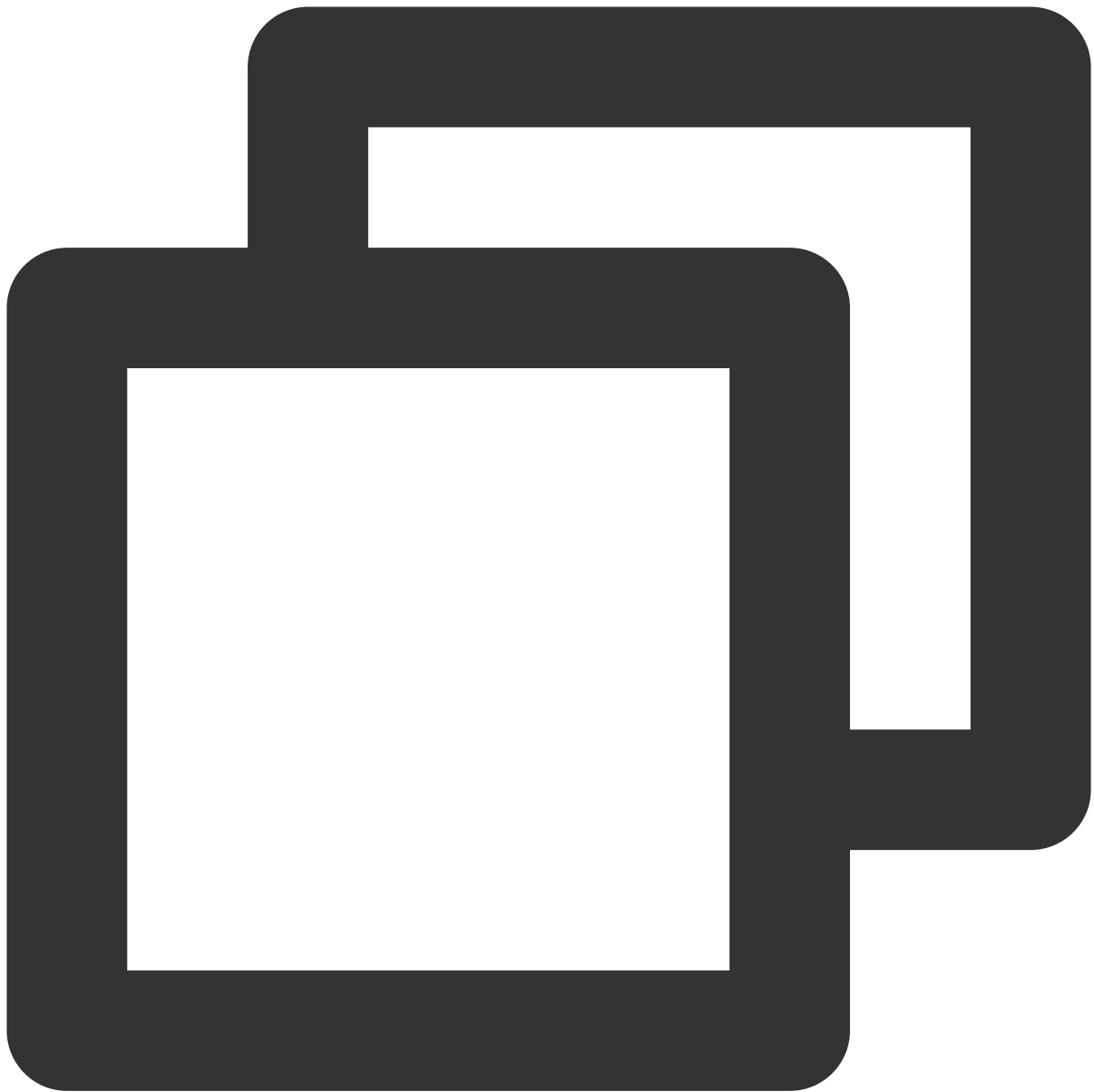
매개변수	유형	설명
userIDList	Array	호출할 사용자의 ID
type	Number	0-알 수 없음, 1-음성 통화, 2-영상 통화
groupID	String	IM 그룹 ID
timeout	String	제한 시간(선택 사항)
roomID	String	방 ID(선택 사항)
offlinePushInfo	Object	사용자 지정 오프라인 알림(이 매개변수는 선택 사항이며 tsignaling 0.8.0 이상에서만 유효)

offlinePushInfo

매개변수	유형	설명
title	string	오프라인 알림 제목(선택 사항)
description	string	오프라인 알림 내용(선택 사항)
androidOPPOChannelID	string	OPPO 8.0 이상에서 오프라인 알림을 위한 채널 ID(선택 사항)
extension	string	통과 콘텐츠(이 매개변수는 선택 사항이며 tsignaling 0.9.0 이상에서만 유효)

accept

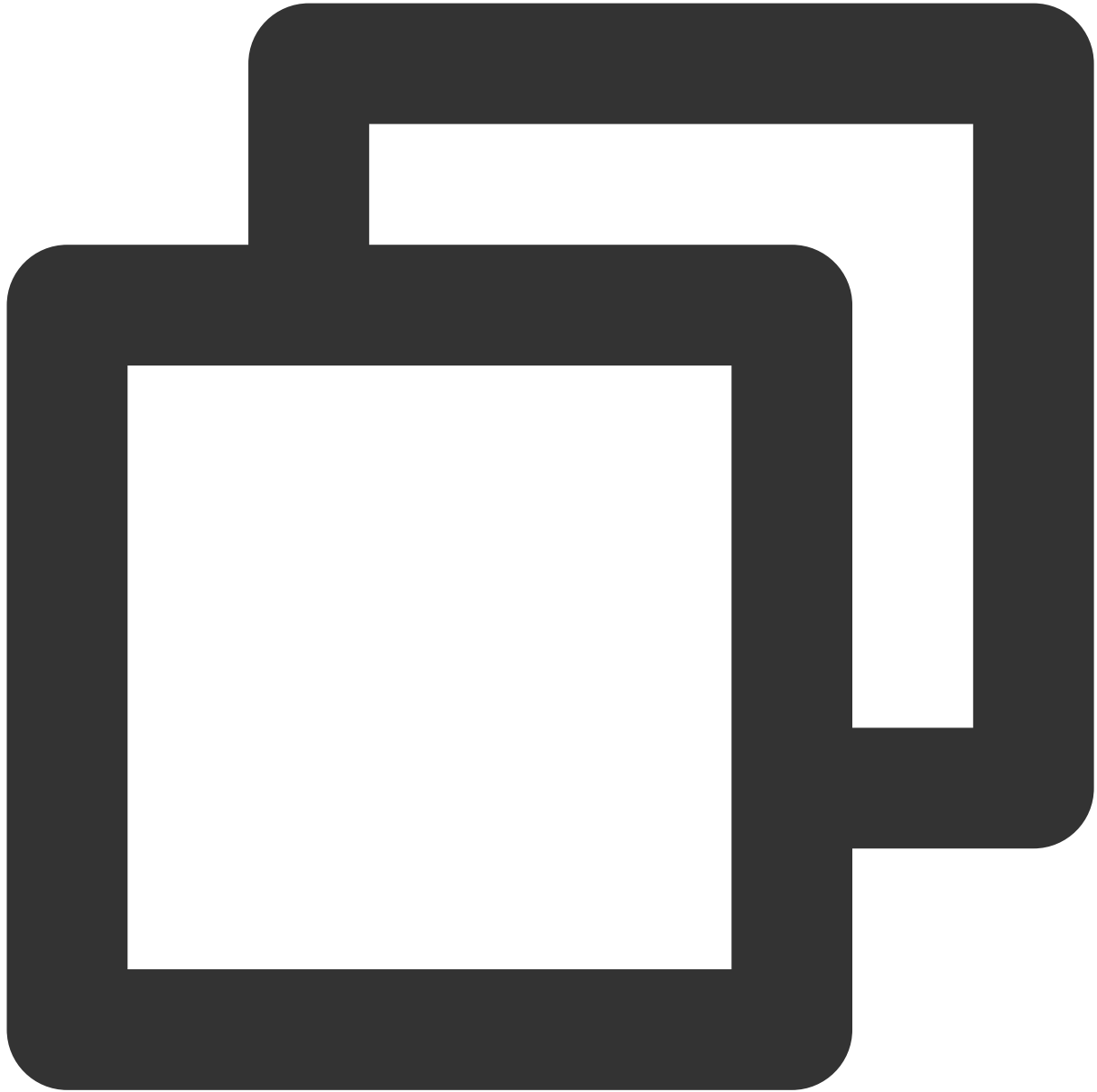
이 API는 `TUICallEvent.INVITED` 콜백을 수신한 후 통화를 수락하는 데 사용됩니다.



```
tuiCallEngine.on(TUICallEvent.INVITED, () => {  
  tuiCallEngine.accept().promise.then(() => {  
    //success  
  }).catch(error => {  
    console.warn('accept error:', error);  
  });  
});
```

reject

이 API는 `TUICallEvent.INVITED` 콜백을 수신한 후 통화를 거부하는 데 사용됩니다.



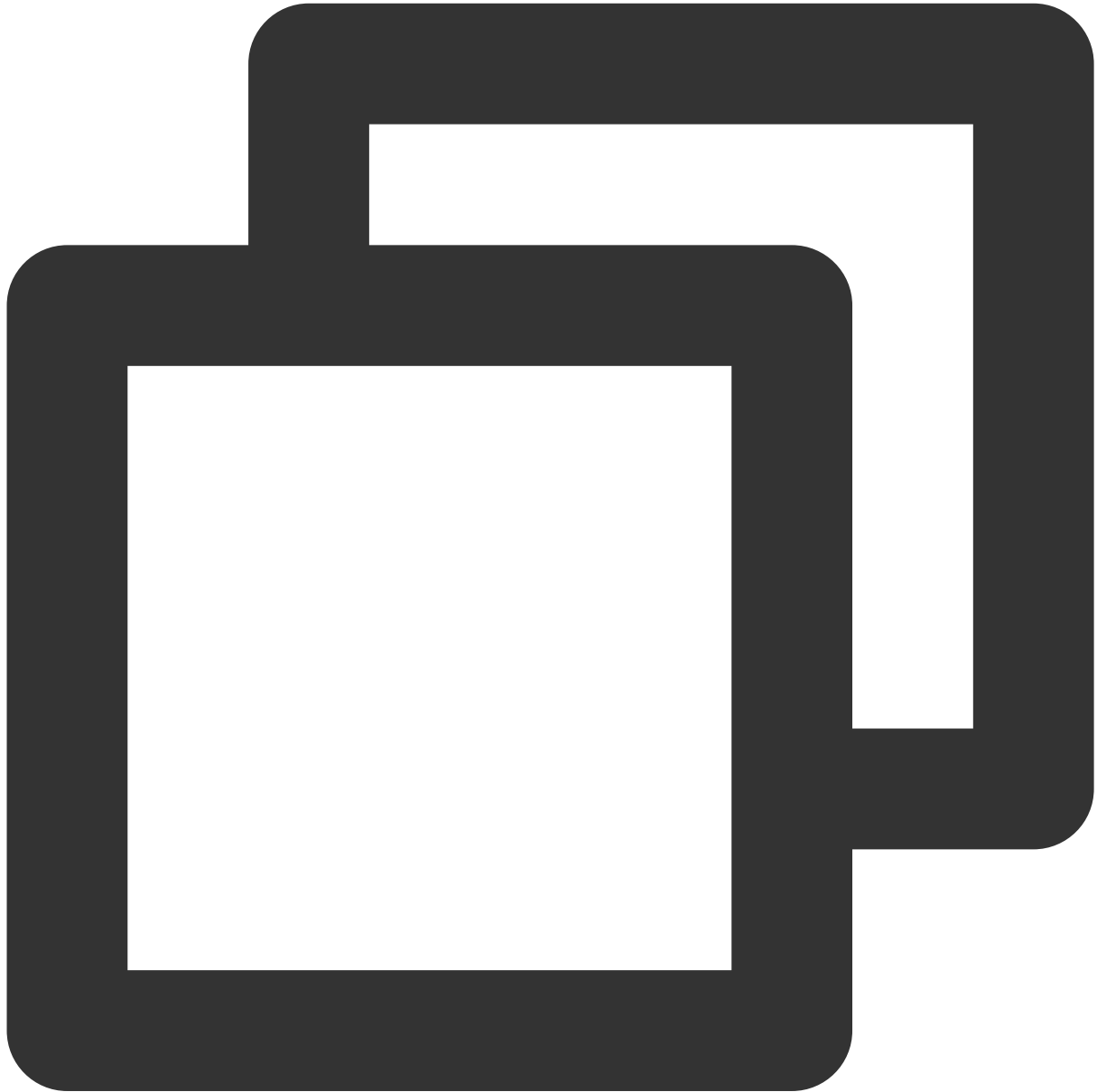
```
tuiCallEngine.on(TUICallEvent.INVITED, () => {
  tuiCallEngine.reject().then(() => {
    //success
  }).catch(error => {
    console.warn('reject error:', error);
  });
});
```

hangup

이 API는 통화를 종료하는 데 사용됩니다.

통화 중인 경우 이 API로 통화를 종료할 수 있습니다.

통화가 아직 응답이 없는 경우, 이 API는 통화를 취소합니다

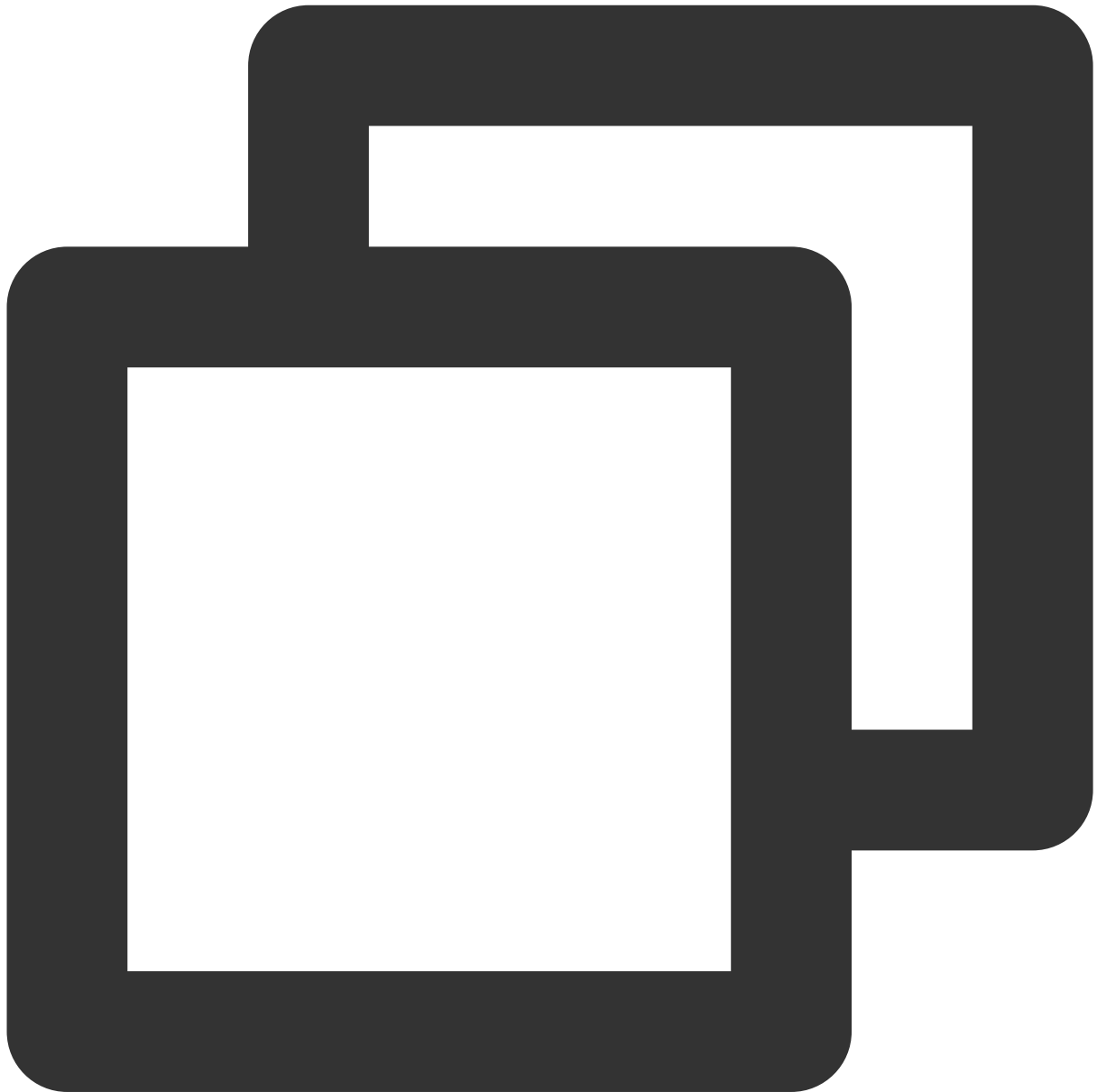


```
tuiCallEngine.hangup().then(() => {  
  //success  
}).catch(error => {  
  console.warn('hangup error:', error);  
});
```

inviteUser

이 API는 현재 그룹 통화에 사용자를 초대하는 데 사용됩니다.

이 API는 그룹 통화의 참가자가 새 사용자를 초대하기 위해 호출합니다.



```
const userIDList = ['jack', 'john'];
const params = {
  userIDList
};
tuiCallEngine.inviteUser(params).then(() => {
  // success
});
```

```

}).catch(error => {
    console.error('inviteUser error:', error);
});

```

매개변수는 다음과 같습니다:

매개변수	유형	설명
userIDList	Array	초청자의 userID
offlinePushInfo	Object	사용자 지정 오프라인 알림(선택 사항). 이 매개변수는 tsignaling 버전이 0.8.0 이상인 경우에만 유효합니다.

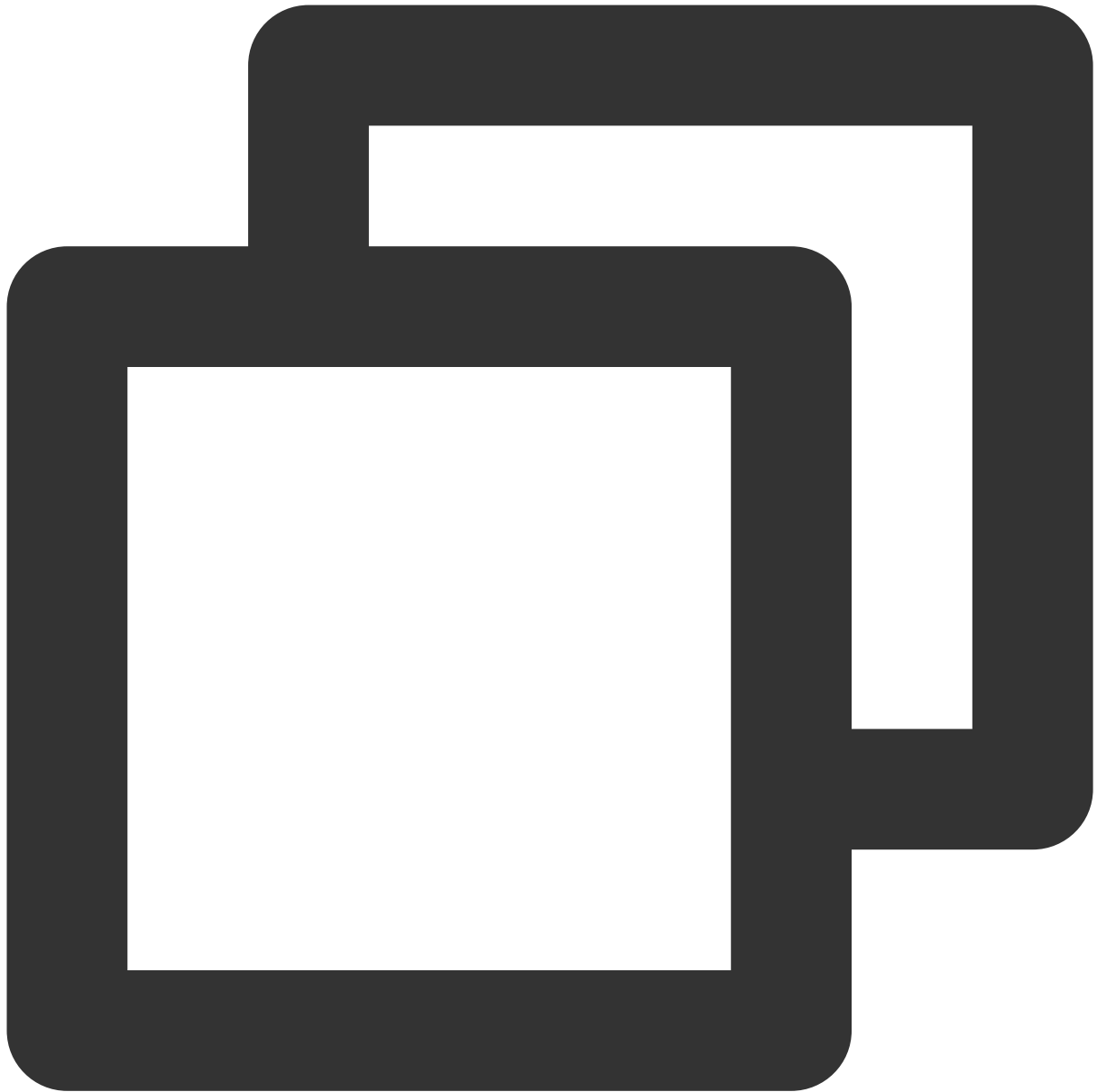
offlinePushInfo

매개변수	유형	설명
title	string	오프라인 알림 제목(선택 사항)
description	string	오프라인 알림 내용(선택 사항)
androidOPPOChannelID	string	OPPO 8.0 이상에서 오프라인 알림을 위한 채널 ID(선택 사항)
extension	string	통과 콘텐츠(이 매개변수는 선택 사항이며 tsignaling 0.9.0 이상에서만 유효)

joinInGroupCall

이 API는 그룹 통화에 참여하는 데 사용됩니다.

이 API는 그룹 구성원이 그룹 통화에 참여하기 위해 호출합니다.



```
const params = {
  roomID: 123,
  type: 1,
  groupID: 111
};
tuiCallEngine.joinInGroupCall(params).then(() => {
  // success
}).catch(error => {
  console.error('joinInGroupCall error:', error);
});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명
roomID	Number	방 ID. 현재는 숫자로 된 방 ID만 사용할 수 있으며, 문자열 형식의 방 ID는 향후 지원될 예정입니다
groupID	String	그룹 ID
type	Number	통화 유형. 1 - 음성 통화, 2 - 영상 통화

switchCallMediaType

통화 유형을 변경합니다.

이 API는 1v1 통화에만 작동합니다.

ERROR 콜백 전환 실패. code: 60001



```
// 1 음성 통화; 2 영상 통화
tuiCallEngine.switchCallMediaType(2).then(() => {
  //success
}).catch(error => {
  console.warn('switchCallMediaType error:', error);
});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명

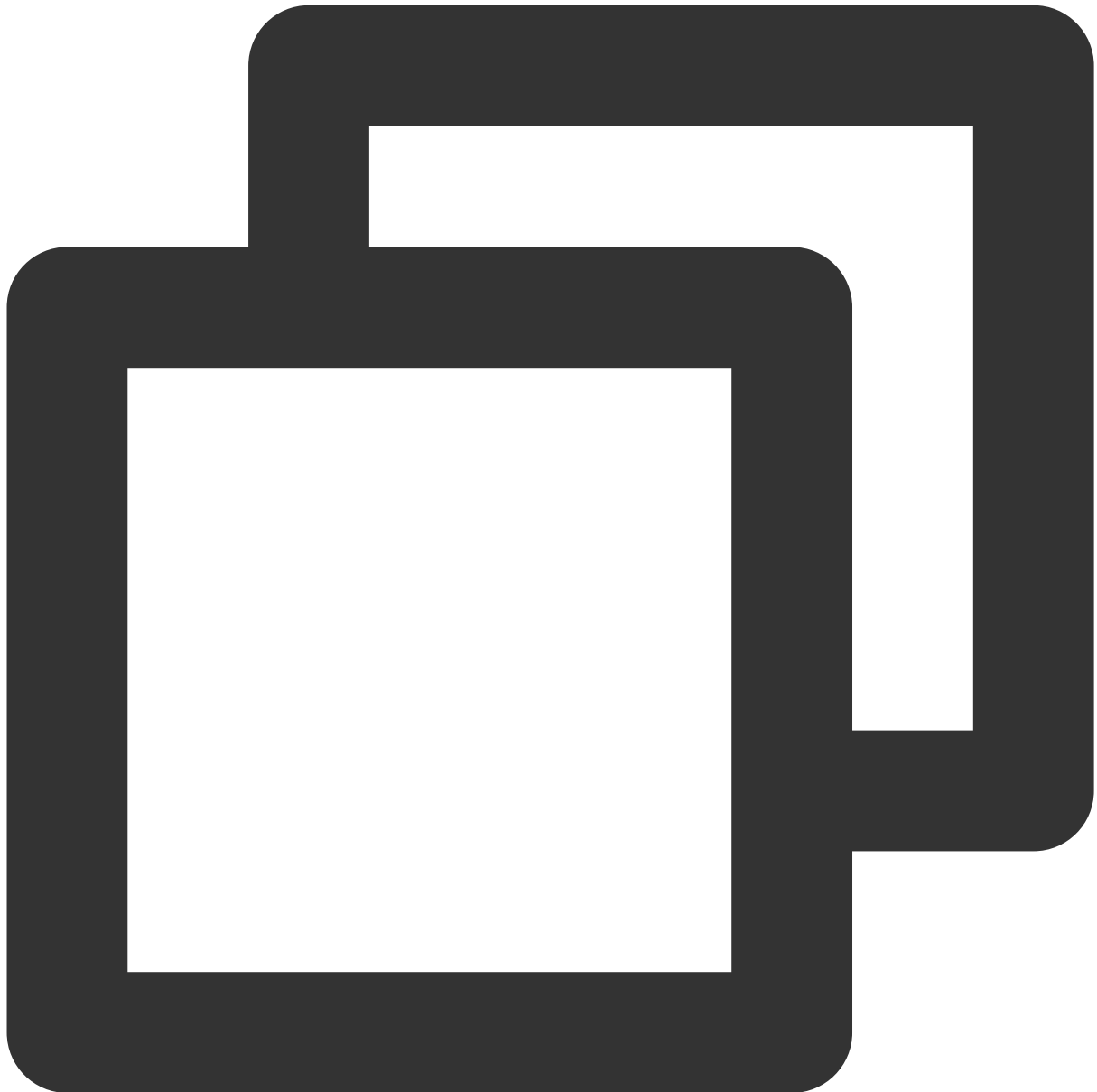
newMediaType

Number

1-음성 통화, 2-영상 통화

startRemoteView

이 API는 원격 비디오 렌더링을 시작하는 데 사용됩니다. [USER_VIDEO_AVAILABLE](#) 이벤트를 수신한 후 해당 원격 비디오 렌더링을 시작합니다.



```
let promise = tuiCallEngine.startRemoteView({
  userID: 'user1',
  videoViewDomID: 'video_1',
});
```

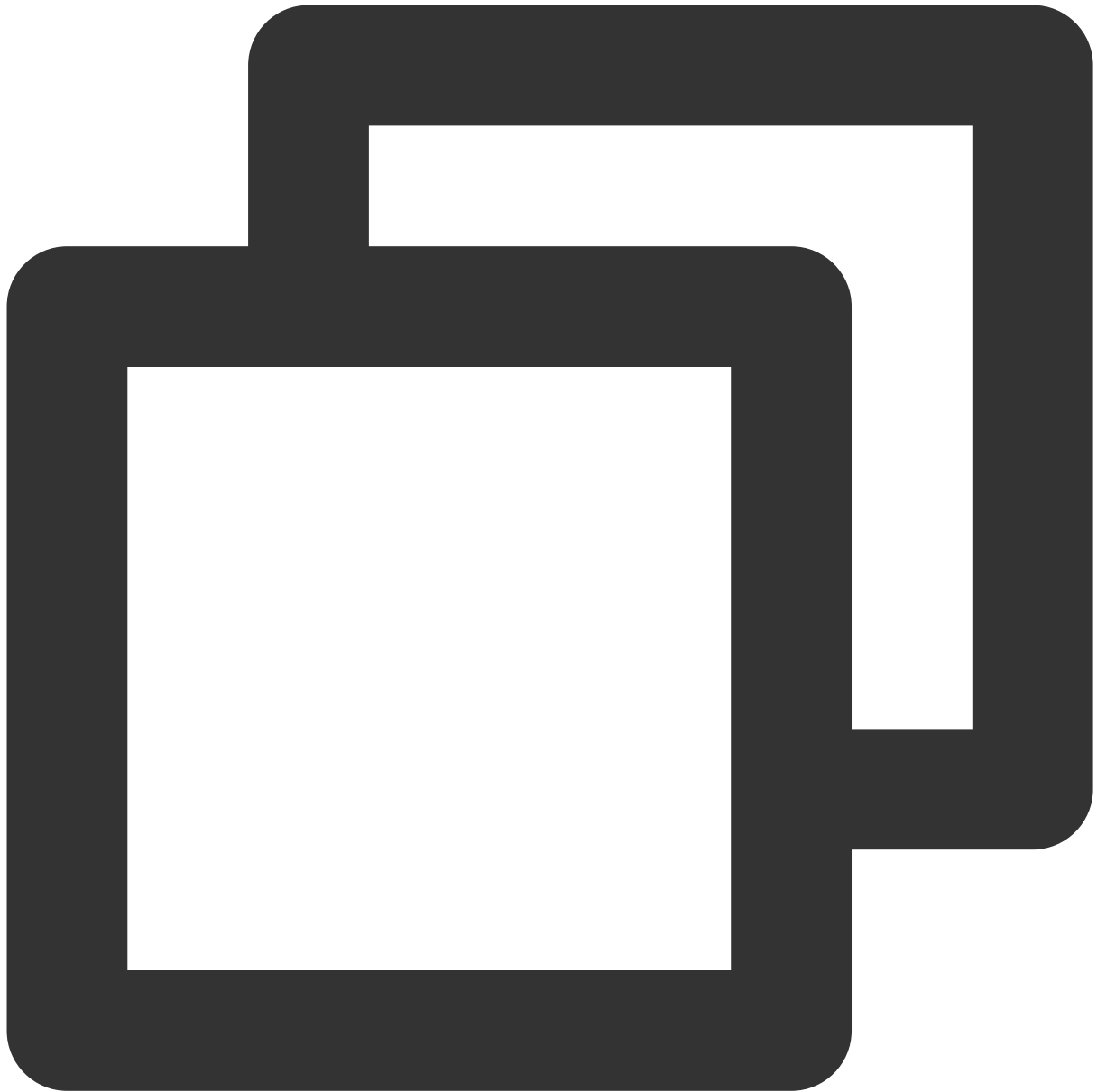
```
promise.then(() => {
  //success
}).catch(error => {
  console.warn('startRemoteView error:', error);
});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명
userID	String	사용자 ID
videoViewDomID	String	사용자의 데이터는 이 dom id 노드에서 렌더링됩니다

stopRemoteView

이 API는 원격 비디오 렌더링을 중지하는 데 사용됩니다



```
tuiCallEngine.stopRemoteView({userID: 'user1'});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명
userID	String	사용자 id

startLocalView

이 API는 로컬 비디오 렌더링을 시작하는 데 사용됩니다



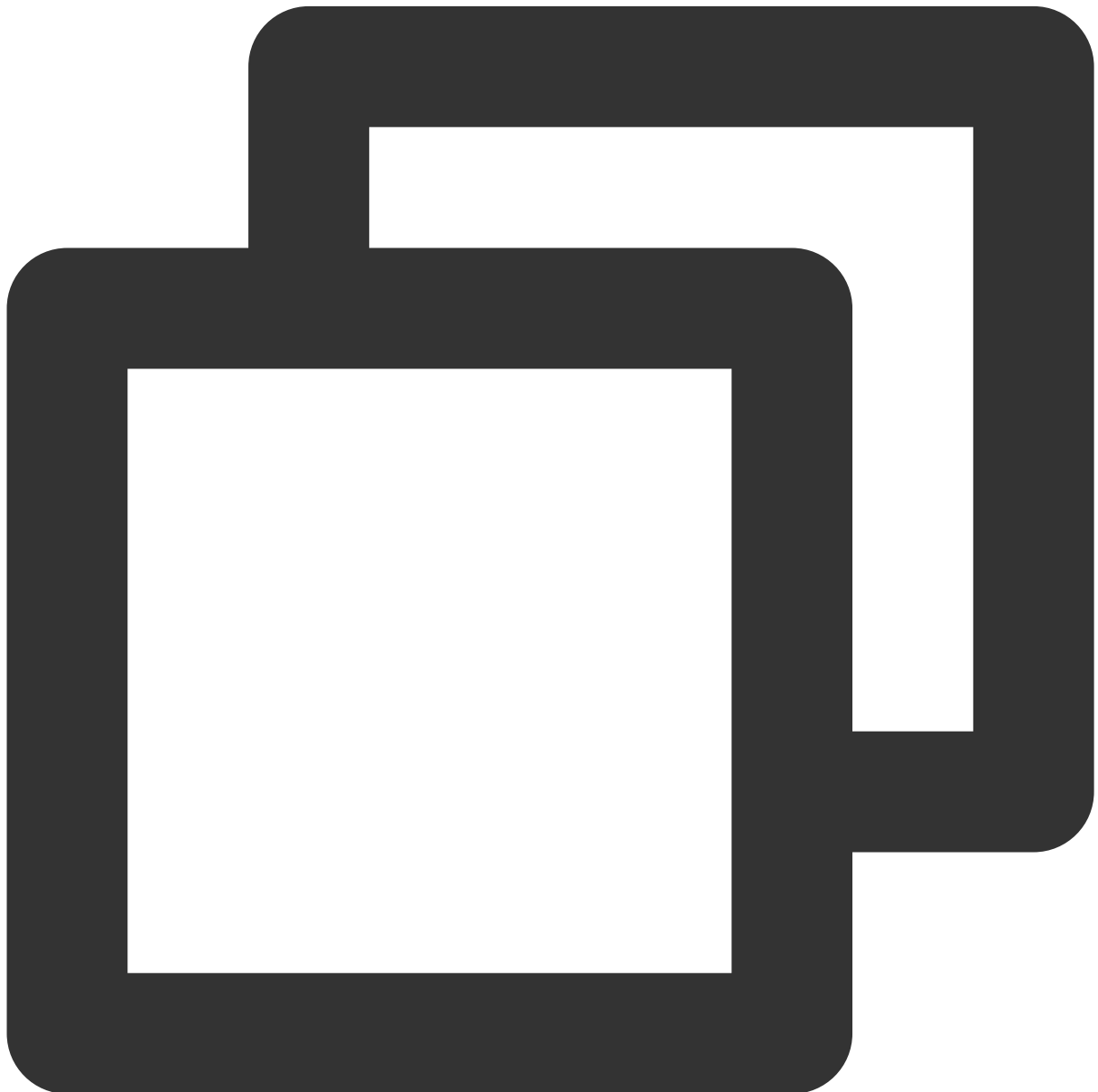
```
let promise = tuiCallEngine.startLocalView({
  userID: 'user1',
  videoViewDomID: 'video_1'
});
promise.then(() => {
  //success
}).catch(error => {
  console.warn('startLocalView error:', error);
});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명
userID	String	사용자 ID
videoViewDomID	String	사용자의 데이터는 이 dom id 노드에서 렌더링됩니다

stopLocalView

이 API는 로컬 비디오 렌더링을 중지하는 데 사용됩니다



```
let promise = tuiCallEngine.stopLocalView({userID: 'user1'});
```

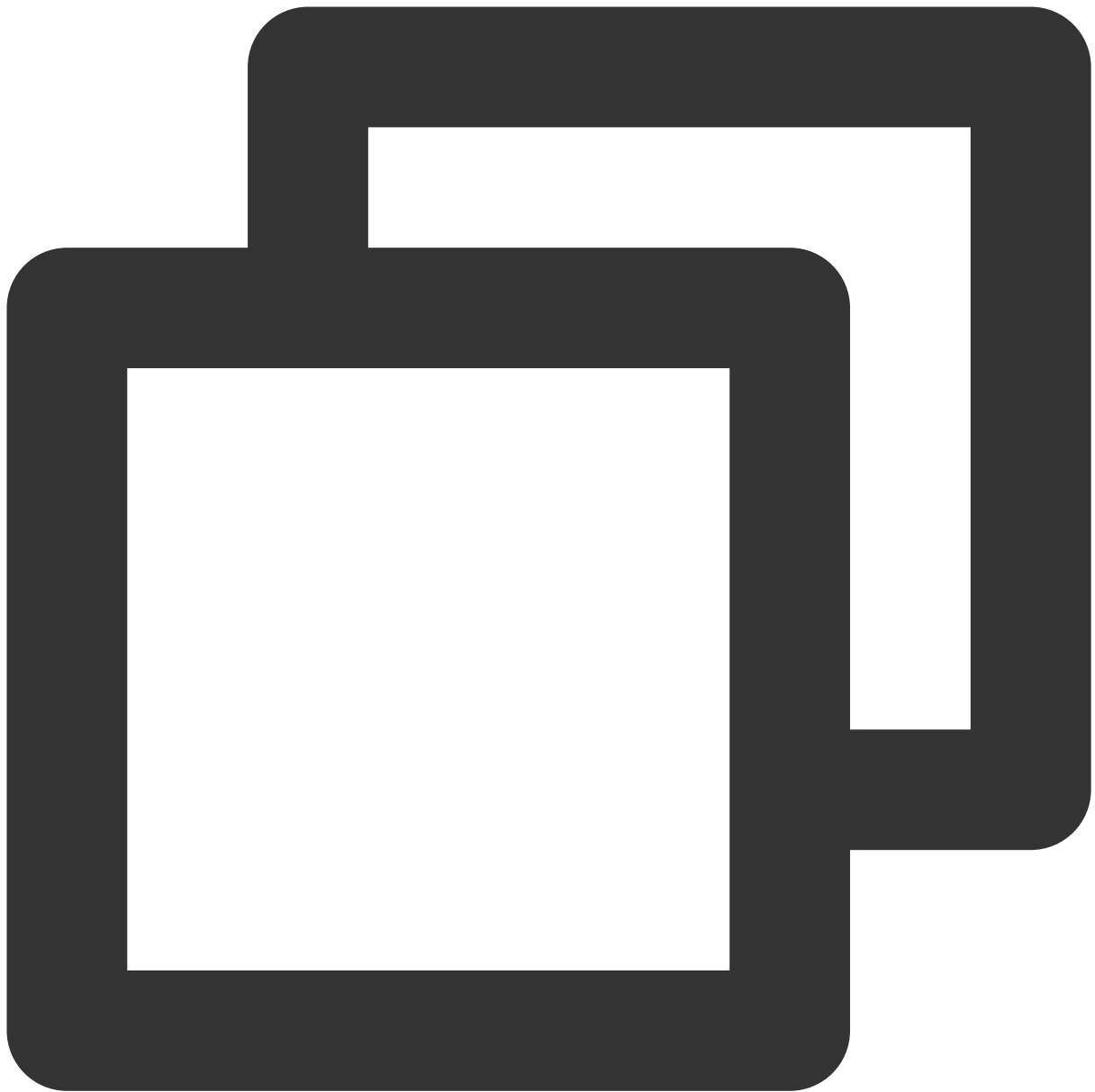
```
promise.then(() => {  
    //success  
}).catch(error => {  
    console.warn('stopLocalView error:', error)  
});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명
userID	String	사용자 ID

openCamera

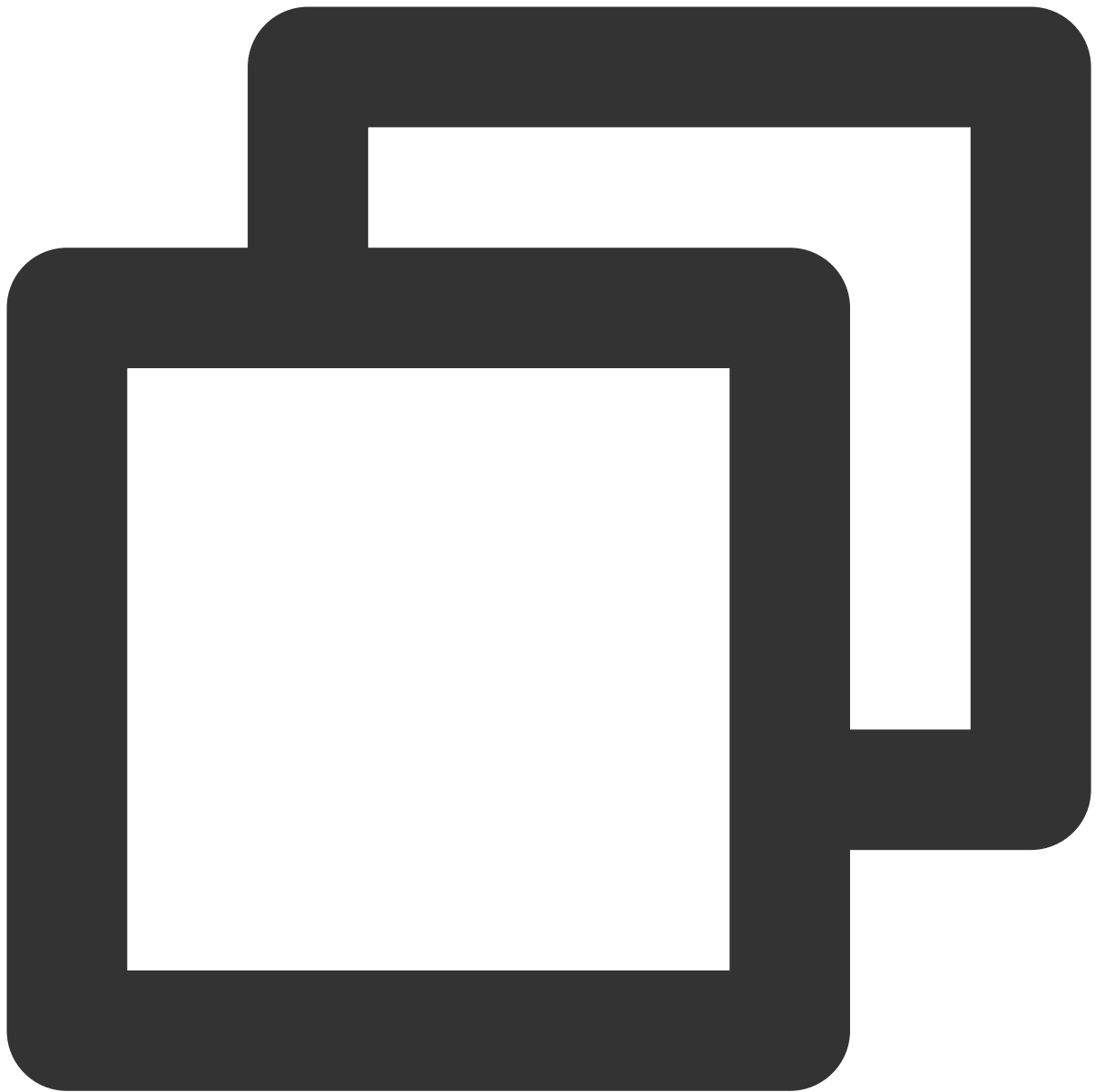
이 API는 카메라를 켜는 데 사용됩니다.



```
tuiCallEngine.openCamera().then(() => {  
  //success  
}).catch(error => {  
  console.warn('openCamera error:', error);  
});
```

closeCamara

이 API는 카메라를 끄는 데 사용됩니다



```
tuiCallEngine.closeCamera().then(() => {  
  //success  
}).catch(error => {  
  console.warn('closeCamara error:', error);  
});
```

openMicrophone

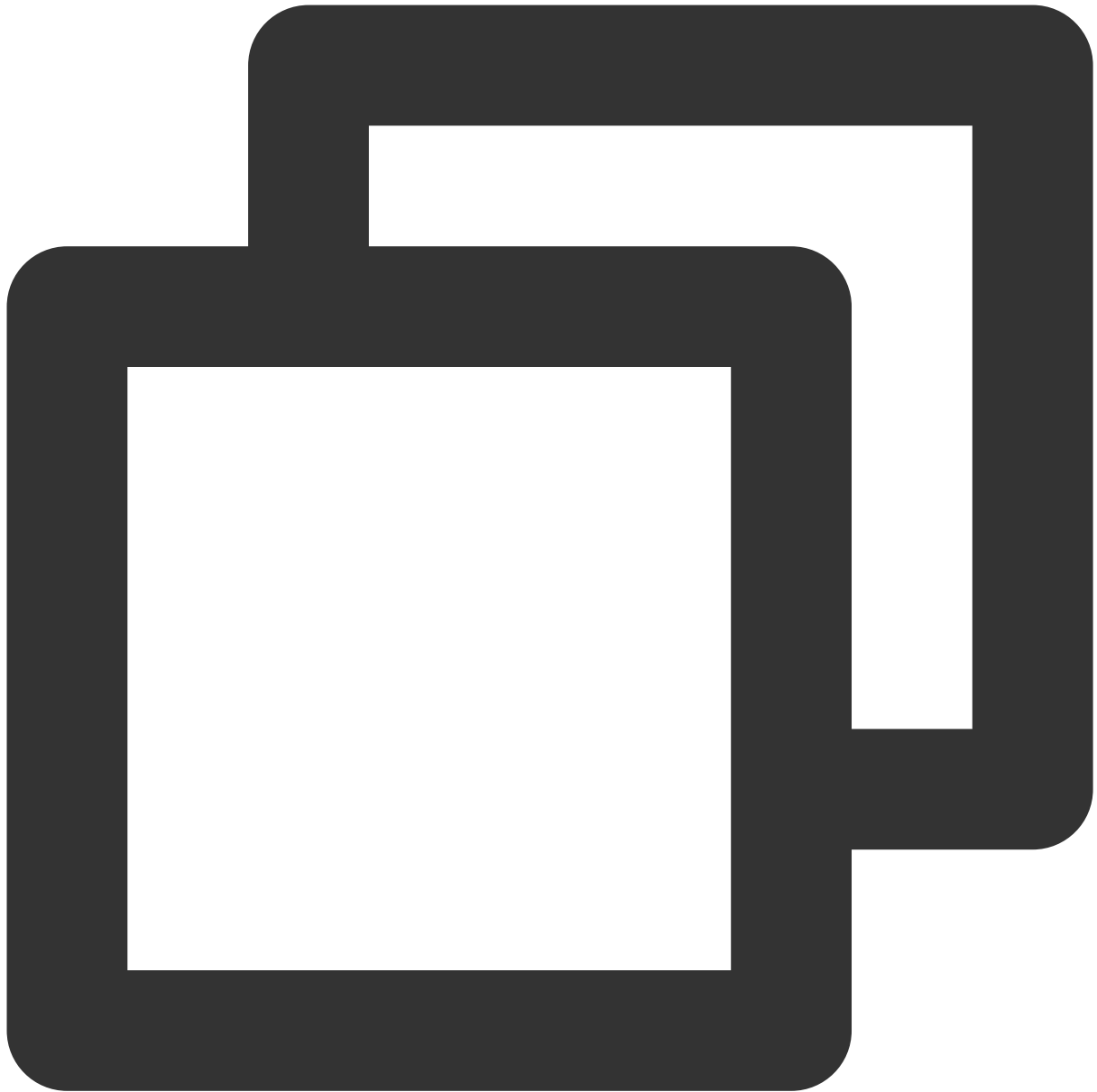
이 API는 마이크를 켜는 데 사용됩니다.



```
tuiCallEngine.openMicrophone().then(() => {  
  //success  
}).catch(error => {  
  console.warn('openMicrophone error:', error);  
});
```

closeMicrophone

이 API는 마이크를 끄는 데 사용됩니다.



```
tuiCallEngine.closeMicrophone().then(() => {  
  //success  
}).catch(error => {  
  console.warn('closeMicrophone error:', error);  
});
```

setVideoQuality

이 API는 비디오 품질을 설정하는 데 사용됩니다.



```
const profile = '720p';
tuiCallEngine.setVideoQuality(profile).then(() => {
  //success
}).catch(error => {
  console.warn('setVideoQuality error:', error)
}); // 비디오 품질을 720p로 설정
```

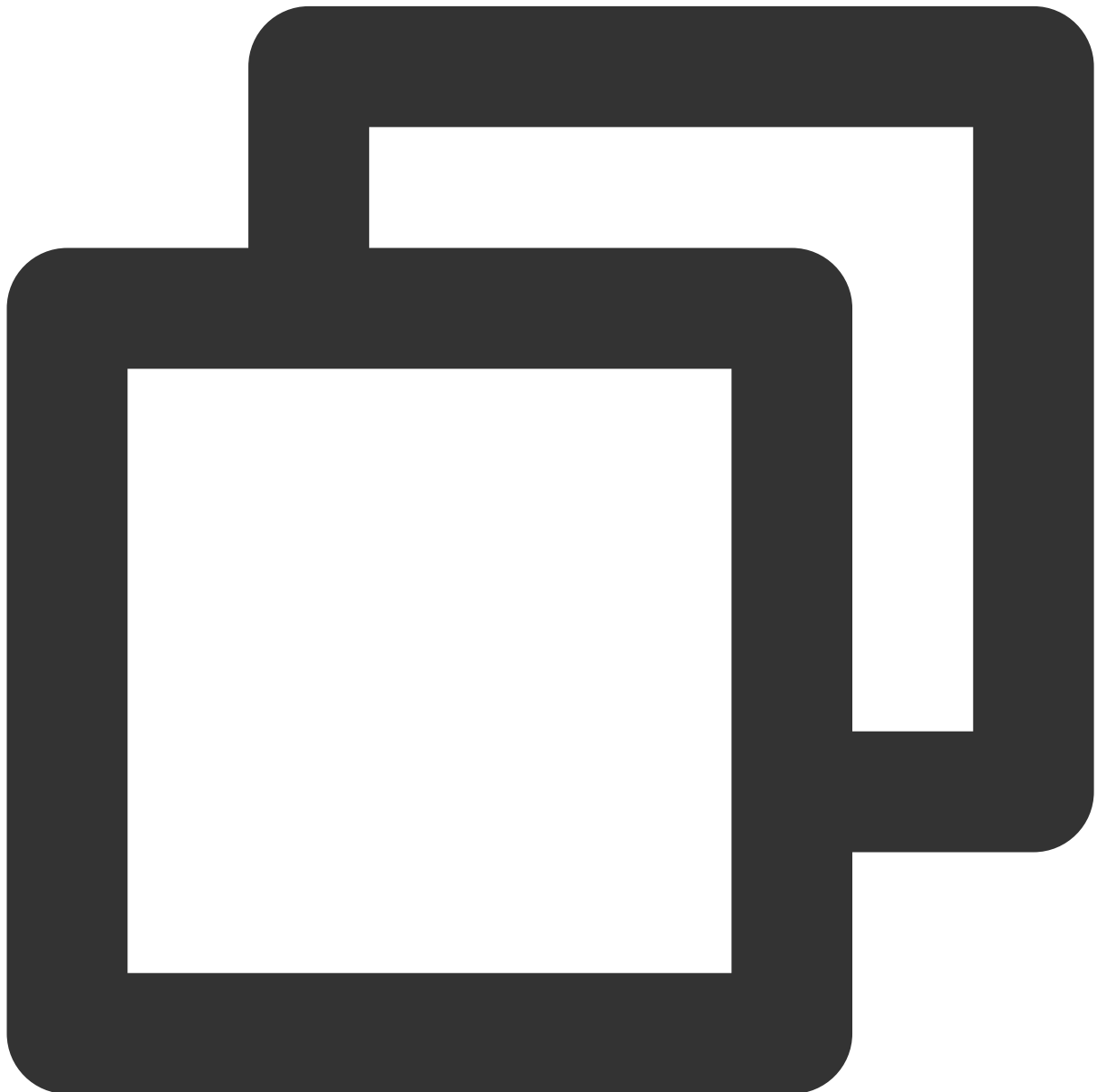
매개변수는 다음과 같습니다:

비디오 Profile	해상도(너비 x 높이)

480p	640 × 480
720p	1280 × 720
1080p	1920 × 1080

getDeviceList

이 API는 장치 목록을 가져오는 데 사용됩니다.



```
tuiCallEngine.getDeviceList("camera").then((devices) => {
```

```
console.log(devices);
}).catch(error => {
  console.warn('getDeviceList error:', error);
});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명
deviceType	String	장치 유형. 유효한 값: 'camera'-카메라, 'microphones'-마이크

switchDevice

이 API는 다른 카메라나 마이크로 변경하는 데 사용됩니다.



```
let promise = tuiCallEngine.switchDevice({
  deviceType: 'video',
  deviceId: cameras[0].deviceId
});
promise.then(() => {
  //success
}).catch(error => {
  console.warn('switchDevice error:', error)
});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명
deviceType	String	변경할 장치의 유형 'video' 카메라 'audio' 마이크
deviceId	String	변경할 장치의 ID getCameras()를 사용하여 카메라 ID 가져오기 getMicrophones()를 사용하여 마이크 ID 가져오기

enableAIVoice

AI 노이즈 캔슬링을 활성화/비활성화합니다.

참고

AI 노이즈 캔슬링은 v4.12.1 이상에서 지원됩니다. 자세한 내용은 [튜토리얼: AI 노이즈 캔슬링 활성화](#)를 참고하십시오.



```
let promise = tuiCallEngine.enableAIVoice(true);
promise.then(() => {
  // success
}).catch(error => {
  console.warn('enableAIVoice error:', error)
});
```

매개변수는 다음과 같습니다:

매개변수	유형	설명

enable	Boolean	AI 노이즈 캔슬링 활성화 또는 비활성화 여부입니다. enable=true(기본값)는 AI 노이즈 캔슬링이 활성화됨 표시
--------	---------	---

TUICallEvent

최종 업데이트 날짜: : 2022-11-21 16:34:37

TUICallEvent API

TUICallEvent API는 음성/영상 통화 컴포넌트의 **콜백 API**입니다.

이벤트 목록

EVENT	설명
TUICallEvent.ERROR	SDK 내부 오류 발생
TUICallEvent.SDK_READY	SDK가 ready 상태
TUICallEvent.KICKED_OUT	반복되는 로그인으로 인해 현재 사용자가 방에서 제거되었음
TUICallEvent.USER_ACCEPT	사용자가 통화를 수락
TUICallEvent.USER_ENTER	사용자가 통화에 참여
TUICallEvent.USER_LEAVE	사용자가 통화를 종료
TUICallEvent.REJECT	사용자가 통화를 거절
TUICallEvent.NO_RESP	초대된 사용자가 응답하지 않았음
TUICallEvent.LINE_BUSY	초대된 사용자가 통화 중
TUICallEvent.CALLING_TIMEOUT	통화 시간 초과(초대 대상자가 수신)
TUICallEvent.USER_VIDEO_AVAILABLE	원격 사용자가 카메라를 켜거나 끄
TUICallEvent.USER_AUDIO_AVAILABLE	원격 사용자가 마이크를 켜거나 끄
TUICallEvent.USER_VOICE_VOLUME	원격 사용자가 통화 볼륨을 조정
TUICallEvent.GROUP_CALL_INVITEE_LIST_UPDATE	그룹 통화 초대 목록이 업데이트되었음
TUICallEvent.INVITED	통화에 초대되었음
TUICallEvent.CALLING_CANCEL	통화가 취소되었음(초대 대상자가 수신함)

EVENT	설명
<code>TUICallEvent.CALLING_END</code>	통화가 종료되었음
<code>TUICallEvent.DEVICED_UPDATED</code>	장치 목록이 업데이트되었음
<code>TUICallEvent.CALL_TYPE_CHANGED</code>	통화 유형이 변경되었음

ERROR

SDK 내부에서 오류가 발생했습니다.

```
let onError = function(error) {
  console.log(error)
};
tuiCallEngine.on(TUICallEvent.ERROR, onError);
```

SDK_READY

SDK ready 상태 진입 시, 이 콜백을 수신합니다.

```
let onSDKReady = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.SDK_READY, onSDKReady);
```

KICKED_OUT

반복 로그인으로 인해 방에서 퇴장되었습니다.

```
let handleOnKickedOut = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.KICKED_OUT, handleOnKickedOut);
```

USER_ACCEPT

사용자가 전화를 받았습니다.

```
let handleUserAccept = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.USER_ACCEPT, handleUserAccept);
```

USER_ENTER

사용자가 통화 참여에 동의했습니다.

```
let handleUserEnter = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.USER_ENTER, handleUserEnter);
```

USER_LEAVE

사용자가 통화를 종료하는 데 동의했습니다.

```
let handleUserLeave = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.USER_LEAVE, handleUserLeave);
```

REJECT

사용자가 통화를 거절했습니다.

```
let handleInviteeReject = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.REJECT, handleInviteeReject);
```

NO_RESP

초대 받은 사람이 응답하지 않았습니다.

C2C 통화에서 초대받은 사람이 응답하지 않으면 초대된 사람이 이 콜백을 받게 됩니다.

그룹 통화에서 모든 초대 대상자는 이 콜백을 받을 수 있습니다. 예를 들어 사용자 A가 사용자 B와 사용자 C를 그룹 통화에 초대했지만 B가 응답하지 않으면 A와 C 모두 이 콜백을 받습니다.

```
let handleNoResponse = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.NO_RESP, handleNoResponse);
```

LINE_BUSY

초대받은 사람이 통화 중입니다.

```
let handleLineBusy = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.LINE_BUSY, handleLineBusy);
```

CALLING_TIMEOUT

통화 시간이 초과되었습니다. 이 콜백은 초대받은 사람이 받았습니다.

```
let handleCallingTimeout = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.CALLING_TIMEOUT, handleCallingTimeout);
```

USER_VIDEO_AVAILABLE

원격 사용자가 카메라를 켜거나 끕니다.

```
let handleUserVideoChange = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.USER_VIDEO_AVAILABLE, handleUserVideoChange);
```

USER_AUDIO_AVAILABLE

원격 사용자가 마이크를 켜거나 끕니다.

```
let handleUserAudioChange = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.USER_AUDIO_AVAILABLE, handleUserAudioChange);
```

USER_VOICE_VOLUME

원격 사용자가 통화 볼륨을 조정했습니다.

```
let handleUserVoiceVolumeChange = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.USER_VOICE_VOLUME, handleUserVoiceVolumeChange);
```

GROUP_CALL_INVITEE_LIST_UPDATE

그룹 통화의 초대 대상자 목록이 업데이트되었습니다.

```
let handleGroupInviteeListUpdate = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.GROUP_CALL_INVITEE_LIST_UPDATE, handleGroupInviteeListUpdate);
```

INVITED

통화에 초대되었습니다.

```
let handleNewInvitationReceived = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.INVITED, handleNewInvitationReceived);
```

CALLING_CANCEL

통화가 취소되었습니다. 이 콜백은 초대받은 사람이 받았습니다.

```
let handleCallingCancel = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.CALLING_CANCEL, handleCallingCancel);
```

CALLING_END

통화가 종료되었습니다.

```
let handleCallingEnd = function(event) {
  console.log(event)
};
tuiCallEngine.on(TUICallEvent.CALLING_END, handleCallingEnd);
```

DEVICED_UPDATED

장치 목록이 업데이트되었습니다.

```
let handleDeviceUpdated = function({ microphoneList, cameraList, currentMicrophoneID, currentCameraID }) {
  console.log(microphoneList, cameraList, currentMicrophoneID, currentCameraID)
}
```

```
};  
tuiCallEngine.on(TUICallEvent.DEVICED_UPDATED, handleDeviceUpdated);
```

CALL_TYPE_CHANGED

통화 유형이 변경되었습니다.

```
let handleCallTypeChanged = function({ oldCallType, newCallType }) {  
  console.log(oldCallType, newCallType)  
};  
tuiCallEngine.on(TUICallEvent.CALL_TYPE_CHANGED, handleDeviceUpdated);
```


릴리스 노트(TUICallKit)

Web

최종 업데이트 날짜: : 2023-02-17 16:34:22

Version 1.3.1 @2022.11.29

참고

새 버전은 `tuicall-engine-webrtc@1.2.1` 에 종속합니다. 이 버전을 사용하려면 먼저 `tuicall-engine-webrtc` 를 업데이트하십시오.

새로운 기능

양식을 최적화했습니다.

전화를 받기 전에 발신자의 통화 유형 변경을 듣기 위한 지원이 추가되었습니다.

basic demo에서 장치 테스트에 대한 지원을 추가했습니다.

오류 수정

사용자가 전화를 끊을 때 발생하는 내부 로직 오류를 수정했습니다.

Version 1.3.0 @2022.11.14

참고

이 버전으로 업데이트하기 전에 [업데이트 가이드](#)를 읽어보시기 바랍니다.

새로운 기능

이제 통화 보기가 모바일 웹페이지의 세로 모드에 자동으로 적응할 수 있습니다.

전화를 걸 때 로컬 카메라 미리보기 지원이 추가되었습니다.

basic demo에서 통화하기 전에 장치 테스트에 대한 지원을 추가했습니다.

오류 수정

`TUICallKitServer.destroy()` 이후 `tim` 인스턴스가 완전히 종료되지 않는 문제를 수정했습니다.

발신자가 통화 중일 때 발신자가 응답 없음 알림을 받는 문제를 수정했습니다.

vite 컨텍스트에서 `typescript` 유형 패키징 실패를 수정했습니다.

API 변경 사항

`TUICallKitServer.call()` 또는 `TUICallKitServer.groupCall()` 호출 후 오류가 발생하면

`beforeCalling` 콜백이 더 이상 반환되지 않습니다. `try catch`를 사용하여 오류를 캐치할 수 있습니다.

Version 1.2.0 @2022.11.03

새로운 기능

TUICallEngine SDK의 새 버전에 맞게 조정되었습니다.

Version 1.1.0 @2022.10.21

새로운 기능

통화 중 전체 화면 지원이 추가되었습니다.

`<TUICallKitMini/>` 를 사용하여 통화 창 최소화에 대한 지원을 추가했습니다.

오류 수정

알려진 문제를 수정하고 안정성을 개선했습니다.

Version 1.0.3 @2022.10.14

새로운 기능

basic demo에서 빠른 UserID 복사 및 원클릭 창 열기에 대한 지원이 추가되었습니다.

Version 1.0.2 @2022.09.30

새로운 기능

통합 가이드에 데모 및 자세한 지침이 추가되었습니다.

오류 수정

사용자가 방에 처음 입장할 때 디바이스 상태가 표시되지 않던 문제가 수정되었습니다.

webpack을 패키징에 사용할 때 가끔 아이콘을 로딩하지 못하는 문제를 수정했습니다.

몇 가지 알려진 양식 문제를 수정했습니다.

Version 1.0.1 @2022.09.26

새로운 기능

전화를 걸 때 수신자의 마이크 아이콘 숨기기 지원이 추가되었습니다.

오류 수정

basic demo의 SDKAppID 입력 제한을 숫자로 변경했습니다.

Version 1.0.0 @2022.09.23

[TUICallKit 기본 demo](#)

[통합\(TUICallKit\)](#)

[API 문서\(TUICallKit\)](#)

[UI 사용자 정의\(TUICallKit\)](#)

[TUICallKit \(Web\) FAQs](#)

Android & iOS

최종 업데이트 날짜: : 2022-11-21 17:43:59

설명 :

최신 버전의 SDK(v1.1.0.103)의 TUICallEngine API가 일부 변경되었습니다.

- maven latest 업데이트 메커니즘으로 인해 Android 프로젝트 빌드 오류가 발생하는 경우 다음 두 가지 방법을 사용하여 문제를 해결해 보십시오.
 - TUICallKit을 최신 버전으로 업데이트합니다.
 - tuicallkit/build.gradle에서 tuicallengine 종속성을 1.0의 고정 버전 `com.tencent.liteav.tuikit:tuicallengine:1.0.0.53` 으로 지정합니다.
- pod update를 실행할 때 iOS 프로젝트 빌드 오류가 발생하면 다음 두 가지 방법을 사용하여 문제를 해결해 보십시오.
 - TUICallKit을 최신 버전으로 업데이트합니다.
 - Podfile에 `pod 'TUICallEngine', '1.0.0.53'` 을 추가합니다.

Version 1.1.0.103 @ 2022.09.30

- Android&iOS: 그룹 통화에 새로운 구성원을 초대하는 기능을 최적화했습니다.
- Android&iOS: 통화를 하기 전에 녹화, 조정 및 기타 요금이 부과되지 않도록 통화 프로세스를 최적화했습니다.
- Android&iOS: 사용자 정의 오프라인 알림에 대한 지원이 추가되었습니다.
- Android&iOS: 일부 **TUICallEngine** API의 매개변수가 변경되었습니다. 자세한 내용은 [call\(\)](#), [groupCall\(\)](#), [inviteUser\(\)](#), [onCallReceived\(\)](#)를 참고하십시오.
- Android&iOS: 그룹 통화 중 간헐적으로 발생하는 콜백 오류를 수정했습니다.
- Android&iOS: 반복 로그인 또는 UserSig 만료로 인한 상태 비정상 문제를 수정했습니다.
- iOS: Objective-C와 Swift가 혼합된 언어 TUICallKit 프로젝트 빌드 시, 'init' 호출 시 오류가 발생하는 현상이 수정되었습니다.
- Android: Kotlin 프로젝트에 플로팅 창 기능 통합 시 오류가 발생하는 현상이 수정되었습니다.

Version 1.0.0.53 @ 2022.08.15

첫 번째 릴리스:

- Android&iOS: 1v1 및 그룹 음성/영상 통화를 지원합니다.
- Android&iOS: 주요 장치에 대한 오프라인 통화 푸시를 지원합니다.
- Android&iOS: 사용자 정의 프로필 사진 및 대화명을 지원합니다.
- Android&iOS: 플로팅 통화 창을 지원합니다.
- Android&iOS: 사용자 지정 벨소리를 지원합니다.

- Android&iOS: 사용자가 여러 플랫폼에서 로그인한 상태에서의 전화 수신을 지원합니다.