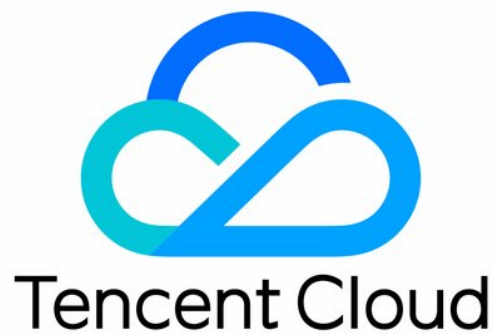


Tencent Real-Time Communication

Voice Chat Room (with 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:

Voice Chat Room (with UI)

TUIVoiceRoom (Android) 통합

TUIVoiceRoom (iOS) 통합

TUICalling API 쿼리

TRTCVoiceRoom(iOS)

TRTCVoiceRoom(Android)

Voice Chat Room (with UI)

TUIVoiceRoom (Android) 통합

최종 업데이트 날짜: : 2022-09-06 14:19:54

컴포넌트 개요

TUIVoiceRoom은 오픈 소스 오디오/비디오 UI 컴포넌트입니다. 프로젝트에 통합한 후 몇 줄의 코드 작성만으로 App에서 '그룹 오디오 채팅' 시나리오를 지원하도록 할 수 있습니다. iOS 플랫폼도 지원합니다. 기본 기능은 다음과 같습니다.

설명 :

TUIKit 시리즈 컴포넌트는 Tencent Cloud의 두 가지 기본 PaaS 서비스, 즉 [Tencent Real-Time Communication](#) 및 [Instant Messaging](#)을 사용합니다. TRTC를 활성화하면 IM과 IM SDK 평가판(100 DAU만 지원)이 자동으로 활성화됩니다. IM 과금 내역은 [요금 안내](#)를 참고하십시오.



컴포넌트 통합

1단계: TUIVoiceRoom 컴포넌트 다운로드 및 가져오기

[Github](#)로 이동하여 코드를 복제하거나 다운로드하고 Android/Source 디렉터리를 프로젝트에 복사하고 다음 가져오기 작업을 완료합니다.

- 아래와 같이 `setting.gradle` 에서 가져오기 완료:

```
include ':Source'
```

- app의 `build.gradle` 파일에 Source에 대한 종속성을 추가합니다.

```
api project(':Source')
```

- 루트 디렉터리의 `build.gradle` 파일에 TRTC SDK 및 IM SDK 에 대한 종속성 추가:

```
ext {  
    liteavSdk = "com.tencent.liteav:LiteAVSDK_TRTC:latest.release"  
    imSdk = "com.tencent.imsdk:imsdk-plus:latest.release"  
}
```

2단계: 권한 요청 및 난독화 규칙 구성

AndroidManifest.xml에서 App 권한을 설정합니다. SDK에는 다음 권한이 필요합니다(Android 6.0 이상에서는 런타임 시 마이크 액세스를 요청해야 함).

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

`proguard-rules.pro` 파일에서 SDK 클래스를 난독화 금지 목록에 추가합니다.

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

3단계: 컴포넌트 초기화 및 로그인

```
// 1. 초기화,  
TRTCVoiceRoom mTRTCVoiceRoom = TRTCVoiceRoom.sharedInstance(this);
```

```
mTRTCVoiceRoom.setDelegate(new TRTCVoiceRoomDelegate() {
});
// 2. 로그인,
mTRTCVoiceRoom.login(SDKAppID, userId, userSig, new TRTCVoiceRoomCallback.ActionCallback() {
@Override
public void onCallback(int code, String msg) {
if (code == 0) {
//로그인 성공
}
}
});
```

매개변수 설명:

- **SDKAppID:** TRTC 애플리케이션 ID입니다. TRTC 서비스를 활성화하지 않은 경우 [TRTC 콘솔](#)에 로그인하여 TRTC 애플리케이션을 생성하고 [애플리케이션 정보](#)를 클릭합니다. SDKAppID는 아래와 같습니다.

- **Secretkey:** SDKAppID에 해당하는 TRTC 애플리케이션 키. TRTC 콘솔의 [애플리케이션 관리](#) 페이지에서 SecretKey는 아래와 같습니다.
- **userId:** 현재 사용자의 ID로, 문자(a-z 및 A-Z), 숫자(0-9), 하이픈(-) 및 언더바(_)만 포함할 수 있는 문자열입니다. 사용자 계정 시스템과 일관성을 유지하는 것이 좋습니다.
- **userSig:** SDKAppID, userId 및 Secretkey를 기반으로 계산된 보안 보호 서명입니다. [여기](#)를 클릭하여 디버깅 userSig를 온라인으로. 자세한 내용은 [UserSig](#)를 참고하십시오.

4단계: 음성 채팅방 구현

1. 방 주인은 `TRTCVoiceRoom#createRoom`을 통해 음성 대화방 생성

```
// 1. 방주인은 API를 호출하여 방 생성
int roomId = 12345; //방 id
final TRTCVoiceRoomDef.RoomParam roomParam = new TRTCVoiceRoomDef.RoomParam();
roomParam.roomName = "방 이름";
roomParam.needRequest = false; // 마이크 연결 시 방주인 확인 필요 여부
roomParam.seatCount = 7; // 방의 자리 수, 총 7개로 설정하고 방주인이 한 개를 점유한 후
시청자가 남은 6개 자리 점유
roomParam.coverUrl = "방 표지 이미지의 URL";
mTRTCVoiceRoom.createRoom(roomId, roomParam, new TRTCVoiceRoomCallback.ActionCa
llback() {
@Override
public void onCallback(int code, String msg) {
if (code == 0) {
//방 생성 성공
}
}
});
```

2. 청취자는 `TRTCVoiceRoom#enterRoom`을 통해 오디오 대화방에 입장

```
// 1. 청취자가 API를 호출하여 방에 입장
mTRTCVoiceRoom.enterRoom(roomId, new TRTCVoiceRoomCallback.ActionCallback() {
@Override
public void onCallback(int code, String msg) {
if (code == 0) {
//방 들어가기 성공
}
}
});
```

3. 청취자는 `TRTCVoiceRoom#enterSeat`를 통해 마이크 켜

```
// 1: 청취자가 마이크를 켜기 위해 API 호출
int seatIndex = 2; //좌석 index
mTRTCVoiceRoom.enterSeat(seatIndex, new TRTCVoiceRoomCallback.ActionCallback()
{
@Override
public void onCallback(int code, String msg) {
if (code == 0) {
```

```
//작업 완료
}
}
});
// 2.onSeatListChange 콜백 수신, 마이크 위치 리스트 새로고침
@Override
public void onSeatListChange(final List<TRTCVoiceRoomDef.SeatInfo> seatInfoList) {
}
}
```

4. 방 주인이 `TRTCVoiceRoom#pickSeat`를 통해 청취자 마이크 활성화

```
// 1: 방 주인이 청취자를 초대
int seatIndex = 2; //좌석 index
String userId = "123"; //발언할 사용자의 id
mTRTCVoiceRoom.pickSeat(1, userId, new TRTCVoiceRoomCallback.ActionCallback() {
@Override
public void onCallback(int code, String msg) {
if (code == 0) {
//작업 완료
}
}
});
// 2.onSeatListChange 콜백 수신, 마이크 위치 리스트 새로고침
@Override
public void onSeatListChange(final List<TRTCVoiceRoomDef.SeatInfo> seatInfoList) {
}
}
```

5. 청취자는 `TRTCVoiceRoom#sendInvitation`을 통해 발언 요청

```
// 청취자 앵글
// 1.청취자가 API를 호출하여 말하기 요청
String seatIndex = "1"; //좌석 index
String userId = "123"; //사용자 id
String inviteId = mTRTCVoiceRoom.sendInvitation("takeSeat", userId, seatIndex, null);
// 2.초대가 수락된 후 사용자를 자리에 앉힘
@Override
public void onInviteeAccepted(String id, String invitee) {
if(id.equals(inviteId)) {
mTRTCVoiceRoom.enterSeat(index, null);
}
}
}
```



```
// 방 주인 앵글
// 1. 방 주인이 요청을 수신함
@Override
public void onReceiveNewInvitation(final String id, String inviter, String cmd,
final String content) {
    if (cmd.equals("takeSeat")) {
        // 2. 방 주인이 청취자 요청에 동의
        mTRTCVoiceRoom.acceptInvitation(id, null);
    }
}
```

6. 방 주인은 `TRTCVoiceRoom#sendInvitation`을 통해 청취자 초대

```
// 방 주인 앵글
// 1. 방 주인은 API를 호출하여 청취자에게 발언 요청
String seatIndex = "1"; //좌석 index
String userId = "123"; //사용자 id
String inviteId = mTRTCVoiceRoom.sendInvitation("pickSeat", userId, seatIndex,
null);
// 2. 초대가 수락된 후 사용자를 자리에 앉힘
@Override
public void onInviteeAccepted(String id, String invitee) {
    if(id.equals(inviteId)) {
        mTRTCVoiceRoom.pickSeat(index, null);
    }
}
// 청취자 앵글
// 1. 청취자가 요청을 수신함
@Override
public void onReceiveNewInvitation(final String id, String inviter, String cmd,
final String content) {
    if (cmd.equals("pickSeat")) {
        // 2. 청취자의 방 주인 요청 수락
        mTRTCVoiceRoom.acceptInvitation(id, null);
    }
}
```

7. `TRTCVoiceRoom#sendRoomTextMsg`를 통한 문자 채팅 구현

```
// 발신측: 텍스트 메시지 발송
mTRTCVoiceRoom.sendRoomTextMsg("Hello Word!", null);
// 수신측: 텍스트 메시지 수신
mTRTCVoiceRoom.setDelegate(new TRTCVoiceRoomDelegate() {
    @Override
```

```
public void onRecvRoomTextMsg(String message, TRTCVoiceRoomDef.UserInfo userInfo) {
    Log.d(TAG, userInfo.userName + "님이 발송한 메시지:" + message);
}
});
```

8. TRTCVoiceRoom#sendRoomCustomMsg를 통한 화면 댓글 구현

```
// 발신 측: 사용자 정의 Cmd를 통해 댓글 자막과 '좋아요' 메시지 구분 가능
// eg: "CMD_DANMU": 댓글 자막 메시지, "CMD_LIKE": '좋아요' 메시지
mTRTCVoiceRoom.sendRoomCustomMsg("CMD_DANMU", "Hello world", null);
mTRTCVoiceRoom.sendRoomCustomMsg("CMD_LIKE", "", null);
// 수신측: 사용자 정의 메시지 수신
mTRTCVoiceRoom.setDelegate(new TRTCVoiceRoomDelegate() {
    @Override
    public void onRecvRoomCustomMsg(String cmd, String message, TRTCVoiceRoomDef.UserInfo userInfo) {
        if ("CMD_DANMU".equals(cmd)) {
            // 댓글 자막 메시지 수신
            Log.d(TAG, userInfo.userName + "님이 발송한 댓글 자막 메시지:" + message);
        } else if ("CMD_LIKE".equals(cmd)) {
            // '좋아요' 메시지 수신
            Log.d(TAG, userInfo.userName + "좋아요를 눌렀습니다!");
        }
    }
});
```

FAQ

요구 사항이나 피드백은 colleenyu@tencent.com으로 문의하십시오.

TUIVoiceRoom (iOS) 통합

최종 업데이트 날짜: : 2022-09-06 14:16:52

컴포넌트 개요

TUIVoiceRoom은 오픈 소스 오디오/비디오 UI 컴포넌트입니다. 프로젝트에 통합한 후 몇 줄의 코드 작성만으로 App에서 '그룹 오디오 채팅' 시나리오를 추가할 수 있습니다. 동시에 [Android](#) 플랫폼도 지원합니다. 기본 기능은 다음과 같습니다.

설명 :

TUIKit 시리즈 컴포넌트는 Tencent Cloud의 두 가지 기본 PaaS 서비스, 즉 [Tencent Real-Time Communication](#) 및 [Instant Messaging](#)을 사용합니다. TRTC를 활성화하면 IM과 IM SDK 평가판(100 DAU만 지원)이 자동으로 활성화됩니다. IM 과금 내역은 [요금 안내](#)를 참고하십시오.



컴포넌트 통합

1단계: TUIVoiceRoom 컴포넌트 다운로드 및 가져오기

xcode 프로젝트의 Podfile 과 동일한 수준에 TUIVoiceRoom 폴더를 만들고 [GitHub 리포지토리의 iOS 디렉터리](#)에서 TXAppBasic, Resources, Source 및 TUIVoiceRoom.podspec 파일을 폴더로 복사하고 다음 가져오기 작업을 완료합니다.

- 프로젝트의 Podfile을 열고 다음과 같이 TUIVocieRoom.podspec을 가져옵니다.

```
# path는 Podfile에 상대적인 TXAppBasic.podspec의 경로
pod 'TXAppBasic', :path => "TUIVoiceRoom/TXAppBasic/"
# path는 Podfile에 대한 TUIVoiceRoom.podspec의 경로
pod 'TUIVoiceRoom', :path => "TUIVoiceRoom/", :subspecs => ["TRTC"]
```

- 터미널을 열고 Podfile 디렉터리로 들어가 pod install 을 실행합니다.

```
pod install
```

2단계: 권한 요청 및 난독화 규칙 구성

info.plist 파일에 Privacy > Microphone Usage Description 을 추가해 마이크 권한을 신청해야 합니다.

```
<key>NSMicrophoneUsageDescription</key>
<string>VoiceRoomApp은 오디오가 포함된 비디오를 녹화하려면 마이크에 액세스해야 함</string>
```

3단계: 컴포넌트 초기화 및 로그인

```
// 초기화
let mTRTCVoiceRoom = TRTCVoiceRoom.shared()
// 로그인
mTRTCVoiceRoom.login(sdkAppID: SDKAppID, userId: userId, userSig: userSig) { code, message in
if code == 0 {
//로그인 성공
}
}
```

매개변수 설명:

- **SDKAppID**: TRTC 애플리케이션 ID입니다. TRTC 서비스를 활성화하지 않은 경우 [TRTC 콘솔](#)에 로그인하여 TRTC 애플리케이션을 생성하고 **애플리케이션 정보**를 클릭합니다. SDKAppID는 아래와 같습니다.

The screenshot shows the 'Application Management' page in the Tencent Real-Time Communication console. The left sidebar contains navigation options like Overview, Usage Statistics, Monitoring Dashboard, Development Assistance, Application Management (selected), and Relevant Cloud Services. The main content area has tabs for Application Info, Function Configuration, Callback Configuration, and Quick Start. Under 'Quick Start', there are two steps: 'Step 1: download SDK + auxiliary demo source code' with links for iOS, Mac, Android, Web, Windows, and Flutter; and 'Step 2: obtain the secret key to issue UserSig'. A warning states 'The secret key is sensitive information. Please do not disclose it.' Below this is a 'Secret Key (Key)' field, which is highlighted with a red box and labeled 'Secretkey'. A 'Copy Secret Key' button is present, with the text 'HMAC-SHA256 encrypted' below it.

- **Secretkey**: SDKAppID에 해당하는 TRTC 애플리케이션 키. TRTC 콘솔의 [애플리케이션 관리](#) 페이지에서 SecretKey는 아래와 같습니다.
- **userId**: 현재 사용자의 ID로, 문자(a-z 및 A-Z), 숫자(0-9), 하이픈(-) 및 언더바(_)만 포함할 수 있는 문자열입니다. 사용자 계정 시스템과 일관성을 유지하는 것이 좋습니다.
- **userSig**: SDKAppId, userId 및 Secretkey를 기반으로 계산된 보안 보호 서명입니다. [여기](#)를 클릭하여 디버깅 userSig를 온라인으로. 자세한 내용은 [UserSig](#)를 참고하십시오.

4단계: 음성 채팅방 구현

1. 방 주인은 `TRTCVoiceRoom#createRoom`을 통해 음성 대화방 생성

```
// 오디오 채팅방 매개변수 초기화
let roomParam = VoiceRoomParam()
roomParam.roomName = "방 이름"
roomParam.needRequest = false // 청취자 마이크 활성화 시 방 주인 동의 필요 여부
roomParam.coverUrl = "방 커버 이미지의 URL"
roomParam.seatCount = 7 // 방의 자리 수, 총 7개로 설정하고 방 주인이 한 개를 점유한 후
시청자가 남은 6개 자리 점유
roomParam.seatInfoList = []
```

```

// 좌석 정보 초기화
for _ in 0..  
param.seatCount {
let seatInfo = VoiceRoomSeatInfo()
param.seatInfoList.append(seatInfo)
}
// 방 생성
mTRTCVoiceRoom.createRoom(roomID: yourRoomID, roomParam: roomParam) { (code, message) in
if code == 0 {
// 생성 성공
}
}
}

```

2. 청취자는 `TRTCVoiceRoom#enterRoom`을 통해 오디오 대화방에 입장

```

// 1. 청취자가 API를 호출하여 방에 입장
mTRTCVoiceRoom.enterRoom(roomID: roomID) { (code, message) in
// 방 입장 결과 콜백
if code == 0 {
// 방 입장 성공
}
}
}

```

3. 청취자는 `TRTCVoiceRoom#enterSeat`를 통해 마이크를 켜

```

// 1: 청취자가 마이크를 켜기 위해 API 호출
let seatIndex = 2; //좌석 index
mTRTCVoiceRoom.enterSeat(seatIndex: 2) { (code, message) in
if code == 0 {
// 마이크가 성공적으로 켜짐
}
}
// 2.onSeatListChange 콜백 수신, 마이크 위치 리스트 새로고침
@Override
func onSeatListChange(seatInfoList: [VoiceRoomSeatInfo]) {
// 새로 고친 좌석 리스트
}
}

```

4. 방 주인은 `TRTCVoiceRoom#pickSeat`를 통해 청취자 마이크 활성화

```

// 1: 방 주인이 청취자를 초대
let seatIndex = 2; //좌석 index

```

```

let userId = "123"; //발언할 사용자의 id
mTRTCVoiceRoom.pickSeat(seatIndex: 1, userId: "123") { (code, message) in
if code == 0 {
}
}
// 2.onSeatListChange 콜백 수신, 마이크 위치 리스트 새로고침
func onSeatListChange(seatInfoList: [VoiceRoomSeatInfo]) {
// 새로 고친 좌석 리스트
}

```

5. 청취자가 `TRTCVoiceRoom#sendInvitation`을 통해 발언 요청

```

// 청취자 앵글
// 1.청취자가 API를 호출하여 말하기 요청
let seatIndex = "1"; //좌석 index
let userId = "123"; //사용자 id
let inviteId = mTRTCVoiceRoom.sendInvitation(cmd: "takeSeat", userId: ownerUserId, content: "1") { (code, message) in
// 발송 결과 콜백
}
// 2.초대가 수락된 후 사용자를 자리에 앉힘
func onInviteeAccepted(identifier: String, invitee: String) {
if identifier == selfID {
self.mTRTCVoiceRoom.enterSeat(seatIndex: ) { (code, message) in
// 결과 콜백
}
}
}
// 방 주인 앵글
// 1. 방 주인이 요청을 수신함
func onReceiveNewInvitation(identifier: String, inviter: String, cmd: String, content: String) {
if cmd == "takeSeat" {
// 2.방 주인이 청취자 요청에 동의
self.mTRTCVoiceRoom.acceptInvitation(identifier: identifier, callback: nil)
}
}
}

```

6. 방 주인은 `TRTCVoiceRoom#sendInvitation`을 통해 청취자 초대

```

// 방 주인 앵글
// 1.sendInvitation을 호출하여 사용자 '123' 을 좌석 2로 초대
let inviteId = self.mTRTCVoiceRoom.sendInvitation(cmd: "pickSeat", userId: ownerUserId, content: "2") { (code, message) in

```

```

// 발송 결과 콜백
}
// 2.초대가 수락된 후 사용자를 자리에 앉힘
func onInviteeAccepted(identifier: String, invitee: String) {
if identifier == selfID {
self.mTRTCVoiceRoom.pickSeat(seatIndex: ) { (code, message) in
// 결과 콜백
}
}
}
// 청취자 앵글
// 1.청취자가 요청을 수신함
func onReceiveNewInvitation(identifier: String, inviter: String, cmd: String, content: String) {
if cmd == "pickSeat" {
// 2.청취자의 방 주인 요청 수락
self.mTRTCVoiceRoom.acceptInvitation(identifier: identifier, callback: nil)
}
}
}

```

7. TRTCVoiceRoom#sendRoomTextMsg를 통한 문자 채팅 구현

```

// 발신측: 텍스트 메시지 발송
self.mTRTCVoiceRoom.sendRoomTextMsg(message: message) { (code, message) in
}
// 수신측: 텍스트 메시지 수신
func onRecvRoomTextMsg(message: String, userInfo: VoiceRoomUserInfo) {
//수신한 message 정보 처리 방법
}

```

8. TRTCVoiceRoom#sendRoomCustomMsg를 통한 화면 댓글 구현

```

// 예시: 발신측: 사용자 정의 Cmd를 통해 댓글 자막과 '좋아요' 메시지 구분 가능
// eg: "CMD_DANMU": 댓글 자막 메시지, "CMD_LIKE": '좋아요' 메시지
self.mTRTCVoiceRoom.sendRoomCustomMsg(cmd: "CMD_DANMU", message: "hello world",
callback: nil)
self.mTRTCVoiceRoom.sendRoomCustomMsg(cmd: "CMD_LIKE", message: "", callback: nil)
// 수신측: 사용자 정의 메시지 수신
func onRecvRoomCustomMsg(cmd: String, message: String, userInfo: VoiceRoomUserInfo) {
if cmd == "CMD_DANMU" {
// 댓글 자막 메시지 수신
}
}

```



```
}  
if cmd == "CMD_LIKE" {  
  // '좋아요' 메시지 수신  
}  
}
```

FAQ

요구 사항이나 피드백은 colleenyu@tencent.com으로 문의하십시오.

TUICalling API 쿼리

TRTCVoiceRoom(iOS)

최종 업데이트 날짜: : 2022-07-19 16:21:32

TRTCVoiceRoom은 Tencent Cloud의 Real-Time Communication(TRTC)과 Instant Messaging(IM) 서비스를 기반으로 구성되며, 다음 기능을 지원합니다.

- 방 주인이 신규 음성 채팅방을 생성하면 청취자가 음성 채팅방에 입장하여 청취 및 인터랙션.
- 방 주인이 청취자에게 마이크 켜기 요청 또는 마이크가 켜진 사용자의 마이크 강제 끄기.
- 방 주인의 자리 차단 및 청취자 마이크 연결 신청 차단.
- 청취자가 마이크 켜기를 신청하여 마이크가 켜진 호스트가 될 수 있고, 다른 사람들과 음성으로 인터랙션할 수 있으며, 언제든지 마이크를 끄고 일반 청취자가 될 수 있습니다.
- 다양한 텍스트 메시지 및 사용자 정의 메시지를 지원합니다. 사용자 정의 메시지를 이용해 댓글 자막, 좋아요, 선물 기능 등을 구현할 수 있습니다.

설명 :

TUIKit 시리즈 컴포넌트는 Tencent Cloud의 두 가지 기본 PaaS 서비스, 즉 [Tencent Real-Time Communication](#) 및 [Instant Messaging](#)을 사용합니다. TRTC를 활성화하면 IM과 IM SDK 평가판(100 DAU만 지원)이 자동으로 활성화됩니다. IM 과금 내역은 [요금 안내](#)를 참고하십시오.

TRTCVoiceRoom은 오픈 소스 Class로, Tencent Cloud의 두 가지 클로즈드 소스 SDK에 종속됩니다. 자세한 구현 방법은 [음성 채팅방\(iOS\)](#)을 참고하십시오.

- TRTC SDK: [TRTC SDK](#)를 저지연 음성 채팅 컴포넌트로 사용합니다.
- IM SDK: [IM SDK](#)의 AVChatroom을 사용해 채팅방 기능을 구현하며, IM 속성 인터페이스를 통해 마이크 위치 리스트 등 방 정보를 저장하고 초대 신호를 마이크 켜기 신청/마이크 넘기기에 사용할 수 있습니다.

TRTCVoiceRoom API 개요

SDK 기본 함수

API	설명
sharedInstance	컴포넌트 싱글톤 가져오기.
destroySharedInstance	컴포넌트 싱글톤 폐기.

API	설명
setDelegate	이벤트 콜백 설정.
setDelegateHandler	이벤트 콜백이 있는 스레드 설정.
login	로그인.
logout	로그아웃.
setSelfProfile	개인 정보 수정.

방 관련 API

API	설명
createRoom	방 생성(방 주인 호출), 방이 없는 경우 시스템에서 자동으로 새로운 방 생성.
destroyRoom	방 폐기(방 주인 호출).
enterRoom	방 입장(청취자 호출).
exitRoom	방 퇴장(청취자 호출).
getRoomInfoList	방 리스트의 세부 정보 획득.
getUserInfoList	지정 <code>userId</code> 의 사용자 정보 획득, nil인 경우 방 안에 있는 모든 사용자 정보 획득.

마이크 위치 관리 API

API	설명
enterSeat	마이크 연결(청취자와 방 주인 모두 호출 가능).
moveSeat	마이크 위치 이동 (마이크 연결된 호스트 호출 가능).
leaveSeat	마이크 연결 해제(호스트 호출).
pickSeat	마이크 넘기기(방 주인 호출).
kickSeat	마이크 강제 끄기(방 주인 호출).
muteSeat	특정 마이크 위치 음소거/음소거 해제(방 주인 호출).
closeSeat	특정 마이크 위치 차단/차단 해제(방 주인 호출).

로컬 오디오 작업 API

API	설명
startMicrophone	마이크 수집 시작.
stopMicrophone	마이크 수집 중지.
setAudioQuality	오디오 품질 설정.
muteLocalAudio	로컬 음소거 활성화/비활성화.
setSpeaker	스피커 활성화 설정.
setAudioCaptureVolume	마이크 수집 음량 설정.
setAudioPlayOutVolume	재생 볼륨 설정.
setVoiceEarMonitorEnable	인이어 모니터링 활성화/비활성화.

원격 사용자 오디오 작업 API

API	설명
muteRemoteAudio	특정 사용자 음소거/음소거 해제.
muteAllRemoteAudio	모든 사용자 음소거/음소거 해제.

배경 음악 음향 효과 관련 API

API	설명
getAudioEffectManager	배경 음악 음향 효과 관리 객체 TXAudioEffectManager 가져오기.

메시지 발송 관련 API

API	설명
sendRoomTextMsg	방 안에서 텍스트 메시지 발송, 일반적으로 댓글 자막 채팅에 사용.
sendRoomCustomMsg	사용자 정의 텍스트 메시지 발송.

초대 신호 관련 API

API	설명
sendInvitation	사용자에게 초대 발송.

API	설명
acceptInvitation	초대 수락.
rejectInvitation	초대 거부.
cancelInvitation	초대 취소.

TRTCVoiceRoomDelegate API 개요

일반적인 이벤트 콜백

API	설명
onError	오류 콜백.
onWarning	경고 콜백.
onDebugLog	Log 콜백.

방 이벤트 콜백 API

API	설명
onRoomDestroy	방 폐기 콜백.
onRoomInfoChange	음성 채팅방 정보 변경 콜백.
onUserVolumeUpdate	사용자 통화 볼륨 콜백.

마이크 위치 변경 콜백

API	설명
onSeatListChange	전체 마이크 위치 리스트 변경.
onAnchorEnterSeat	사용자 마이크 켜짐(직접 마이크 켜/방 주인 특정 사용자 마이크 켜).
onAnchorLeaveSeat	사용자 마이크 꺼짐(직접 마이크 끄/방 주인이 특정 사용자 마이크 끄).
onSeatMute	방 주인 마이크 음소거.
onUserMicrophoneMute	사용자 마이크 음소거 여부.

API	설명
onSeatClose	방 주인 마이크 차단.

청취자 입장/퇴장 이벤트 콜백

API	설명
onAudienceEnter	청취자 입장 알림 수신.
onAudienceExit	청취자 퇴장 알림 수신.

메시지 이벤트 콜백

API	설명
onRecvRoomTextMsg	텍스트 메시지 수신.
onRecvRoomCustomMsg	사용자 정의 메시지 수신.

신호 이벤트 콜백

API	설명
onReceiveNewInvitation	새로운 초대 요청 수신.
onInviteeAccepted	초대된 사용자가 초대 수락.
onInviteeRejected	초대된 사용자가 초대 거절.
onInvitationCancelled	초대한 사용자가 초대 취소.

기본 SDK API

sharedInstance

[TRTCVoiceRoom](#) 싱글톤 객체를 가져옵니다.

```
/**
 * TRTCVoiceRoom 싱글톤 객체 획득
 *
 * - returns: TRTCVoiceRoom 인스턴스
 * - note: {@link TRTCVoiceRoom#destroySharedInstance()}를 호출하여 싱글톤 객체 폐기
```

```
*/
+ (instancetype) sharedInstance NS_SWIFT_NAME(shared());
```

destroySharedInstance

TRTCVoiceRoom 싱글톤 객체를 폐기합니다.

설명 :

인스턴스 폐기 후에는 외부에 캐시된 TRTCVoiceRoom 인스턴스를 다시 사용할 수 없으며, 다시 [sharedInstance](#)를 호출해 새로운 인스턴스를 획득해야 합니다.

```
/**
 * TRTCVoiceRoom 싱글톤 객체 폐기
 *
 * - note: 인스턴스 폐기 후에는 외부에 캐시된 TRTCVoiceRoom 인스턴스를 다시 사용할 수 없으며, 다시 {@link TRTCVoiceRoom#sharedInstance()}를 호출해 새로운 인스턴스 획득
 */
+ (void) destroySharedInstance NS_SWIFT_NAME(destroyShared());
```

setDelegate

TRTCVoiceRoom 이벤트 콜백은 TRTCVoiceRoomDelegate를 통해 TRTCVoiceRoom의 다양한 상태 알림을 받아볼 수 있습니다.

```
/**
 * 컴포넌트 콜백 인터페이스 설정
 *
 * TRTCVoiceRoomDelegate를 통해 TRTCVoiceRoom의 다양한 상태 알림 가져오기
 *
 * - parameter delegate 콜백 인터페이스
 * - note: TRTCVoiceRoom의 이벤트 콜백은 기본적으로 Main Queue에서 귀하에게 콜백합니다. 이벤트 콜백이 존재하는 큐를 지정할 경우 {@link TRTCVoiceRoom#setDelegateQueue(queue)} 이용 가능
 */
- (void) setDelegate:(id<TRTCVoiceRoomDelegate>) delegate NS_SWIFT_NAME(setDelegate(delegate:));
```

설명 :

setDelegate는 TRTCVoiceRoom의 프록시의 콜백입니다.

setDelegateQueue

이벤트 콜백이 존재하는 스레드 큐를 설정합니다. 기본적으로 메인 스레드 MainQueue로 발송합니다.

```
/**
 * 이벤트 콜백이 있는 큐 설정
 *
 * - parameter queue 큐입니다. TRTCVoiceRoom의 각종 상태 알림을 콜백하며 지정한 queue로
   배포합니다.
 */
- (void) setDelegateQueue: (dispatch_queue_t) queue NS_SWIFT_NAME (setDelegateQueue (q
ueue:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
queue	dispatch_queue_t	TRTCVoiceRoom의 각종 상태를 통지하며, 지정한 스레드 큐로 배포합니다.

login

로그인

```
- (void) login: (int) sdkAppID
  userId: (NSString *) userId
  userSig: (NSString *) userSig
  callback: (ActionCallback _Nullable) callback NS_SWIFT_NAME (login (sdkAppID:userId:u
serSig:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
sdkAppId	int	TRTC 콘솔 >애플리케이션 관리> 애플리케이션 정보에서 SDKAppID를 확인 할 수 있습니다.
userId	NSString	현재 사용자 ID, 문자열 유형은 영어 알파벳(a-z, A-Z), 숫자(0-9), 대시부호(-), 언더바(_)만 허용됩니다.
userSig	NSString	Tencent Cloud가 설계한 일종의 보안 서명입니다. 획득 방식은 UserSig 계산 및 사용 방법 을 참고하십시오.
callback	ActionCallback	로그인 콜백이며, 성공 시 code는 0입니다.

logout

로그아웃합니다.

```
- (void)logout:(ActionCallback _Nullable)callback NS_SWIFT_NAME(logout(callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callback	ActionCallback	로그아웃 콜백이며, 성공 시 code는 0입니다.

setSelfProfile

개인 정보를 수정합니다.

```
- (void)setSelfProfile:(NSString *)userName avatarURL:(NSString *)avatarURL callback:(ActionCallback _Nullable)callback NS_SWIFT_NAME(setSelfProfile(userName:avatarURL:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userName	NSString	대화명.
avatarURL	NSString	프로필 사진 주소.
callback	ActionCallback	개인 프로필 정보 설정 콜백이며, 성공 시 code는 0입니다.

방 관련 API

createRoom

방 생성(방 주인 호출).

```
- (void)createRoom:(int)roomId roomParam:(VoiceRoomParam *)roomParam callback:(ActionCallback _Nullable)callback NS_SWIFT_NAME(createRoom(roomID:roomParam:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
------	----	----

매개변수	유형	의미
roomId	int	방 식별 번호이며, 귀하가 할당하고 통합 관리합니다. 여러 개의 roomId를 1개의 음성 채팅방 리스트로 통합할 수 있으며, Tencent Cloud는 현재 음성 채팅방 리스트 관리 서비스를 제공하지 않으므로 직접 관리하시기 바랍니다.
roomParam	VoiceRoomParam	방 정보입니다. 방 이름, 마이크 위치 정보, 썸네일 정보 등과 같이 방을 설명하는 데 사용됩니다. 마이크 위치 관리가 필요한 경우 방의 마이크 위치 개수를 설정해야 합니다.
callback	ActionCallback	방 생성 결과 콜백이며, 성공 시 code는 0입니다.

방 주인의 정상적인 방송 시작 호출 프로세스는 다음과 같습니다.

- 방 주인이 `createRoom` 을 호출하여 새로운 음성 채팅방을 생성합니다. 이 때 방 ID, 마이크 연결 시 방 주인 확인 필요 여부, 마이크 위치 개수 등 방 속성 정보를 전송합니다.
- 방 주인이 방 생성 후 `enterSeat` 을 호출하여 자리에 입장합니다.
- 방 주인이 컴포넌트의 `onSeatListChange` 마이크 위치 리스트 변경 이벤트 알림을 수신합니다. 이때 마이크 위치 리스트의 변경 내용을 UI 인터페이스에 새로고침할 수 있습니다.
- 방 주인은 마이크 위치 리스트에 사용자가 입장할 때 `onAnchorEnterSeat` 이벤트 알림 또한 수신하며, 이때 자동으로 마이크 수집이 활성화됩니다.

destroyRoom

방 폐기(방 주인 호출). 방 주인은 방 생성 후 해당 함수를 호출해 방을 폐기할 수 있습니다.

```
- (void)destroyRoom: (ActionCallback _Nullable) callback NS_SWIFT_NAME (destroyRoom(callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callback	ActionCallback	방 폐기 결과 콜백이며, 성공 시 code는 0입니다.

enterRoom

방 입장(청취자 호출).

```
- (void)enterRoom: (NSInteger) roomId callback: (ActionCallback _Nullable) callback NS_SWIFT_NAME (enterRoom(roomID:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	NSInteger	방 식별 번호.
callback	ActionCallback	방 입장 결과 콜백이며, 성공 시 code는 0입니다.

청취자가 방에 입장하여 청취하는 정상적인 호출 프로세스는 다음과 같습니다.

1. 청취자가 귀하의 서버에서 최신 음성 채팅방 리스트를 획득하며, 여기에는 여러 음성 채팅방의 roomId 및 방 정보가 포함될 수 있습니다.
2. 청취자가 음성 채팅방 1개를 선택하고 `enterRoom` 을 호출하여 해당 방으로 입장합니다.
3. 방 입장 후 컴포넌트의 `onRoomInfoChange` 방 속성 변경 이벤트 공지를 수신합니다. 이 때 UI에 방 이름 표시, 마이크를 켤 때 방 주인에게 동의 요청 필요 여부 기록 등 방의 속성을 기록할 수 있으며 그에 해당하는 변경이 가능합니다.
4. 방 입장 후 컴포넌트의 `onSeatListChange` 마이크 위치 리스트 변경 이벤트 알림을 수신합니다. 이 때 마이크 위치 리스트의 변경 내용을 UI 인터페이스에 새로고침할 수 있습니다.
5. 방 입장 후 마이크 위치 리스트에 호스트 입장 `onAnchorEnterSeat` 이벤트 공지도 수신합니다.

exitRoom

방 퇴장.

```
- (void)exitRoom:(ActionCallback _Nullable)callback NS_SWIFT_NAME(exitRoom(callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callback	ActionCallback	방 퇴장 결과 콜백이며, 성공 시 code는 0입니다.

getRoomInfoList

방 리스트의 세부 정보를 획득합니다. 방 이름, 방 썸네일은 방 주인이 `createRoom()` 생성 시 `roomInfo`를 통해 설정할 수 있습니다.

설명 :

방 리스트 및 방 정보를 모두 직접 관리하는 경우 해당 함수는 생략할 수 있습니다.

```
- (void) getRoomInfoList: (NSArray<NSNumber * > *) roomIdList callback: (VoiceRoomInfoCallback _Nullable) callback NS_SWIFT_NAME(getRoomInfoList (roomIdList:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomIdList	NSArray<NSNumber>	방 번호 리스트.
callback	RoomInfoCallback	방 세부 정보 콜백.

getUserInfoList

지정 userId의 사용자 정보 획득.

```
- (void) getUserInfoList: (NSArray<NSString * > * _Nullable) userIDList callback: (VoiceRoomUserListCallback _Nullable) callback NS_SWIFT_NAME(getUserInfoList (userIDList:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userIdList	NSArray<NSString>	획득할 사용자 ID 리스트입니다. null인 경우 방 안에 있는 모든 사용자 정보를 획득합니다.
userlistcallback	UserListCallback	사용자 세부 정보 콜백.

마이크 위치 관리 API

enterSeat

직접 마이크 켜기(청취자 및 방 주인 모두 호출 가능).

설명 :

마이크 연결 완료 후, 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onAnchorEnterSeat` 이벤트를 알림을 수신합니다.

```
- (void) enterSeat: (NSInteger) seatIndex callback: (ActionCallback _Nullable) callback NS_SWIFT_NAME(enterSeat (seatIndex:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	NSInteger	마이크를 연결할 마이크 위치 번호.
callback	ActionCallback	작업 콜백.

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다. 청취자의 마이크 연결에 방 주인의 동의가 필요한 시나리오의 경우, 먼저 `sendInvitation` 을 호출하여 방 주인에게 신청하고 `onInvitationAccept` 수신 후 다시 해당 함수를 호출합니다.

moveSeat

마이크 위치 이동(마이크 연결된 호스트 호출 가능).

설명 :

마이크 위치 이동 완료 후 방 안의 모든 구성원은 'onSeatListChange', 'onAnchorLeaveSeat' 및 'onAnchorEnterSeat'의 이벤트 알림을 받게 됩니다. (호스트 호출 후 마이크 좌석 번호 정보만 수정되며, 사용자의 호스트 신분은 변경되지 않습니다.)

```
- (NSInteger)moveSeat:(NSInteger)seatIndex callback:(ActionCallback _Nullable)callback
NS_SWIFT_NAME(moveSeat(seatIndex:callback:))
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	NSInteger	이동할 마이크 위치 번호.
callback	ActionCallback	작업 콜백.

반환값:

반환값	유형	의미
code	NSInteger	마이크 위치 이동 결과(0은 성공, 그 외는 실패, 10001은 인터페이스 호출 빈도 제한).

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다. 청취자의 마이크 연결에 방 주인의 동의가 필요한 시나리오의 경우, 먼저 `sendInvitation` 을 호출하여 방 주인에게 신청하고 `onInvitationAccept` 수신 후 다시 해당 함수를 호출합니다.

leaveSeat

직접 마이크 끄(호스트 호출).

설명 :

마이크 연결 해제 완료 후, 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onAnchorLeaveSeat` 이벤트 알림을 수신합니다.

```
- (void)leaveSeat:(ActionCallback _Nullable)callback NS_SWIFT_NAME(leaveSeat(callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callback	ActionCallback	작업 콜백.

pickSeat

특정 사용자 마이크 켜기(방 주인 호출).

설명 :

방 주인이 마이크를 연결할 사용자를 지정하면 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onAnchorEnterSeat` 이벤트 알림을 수신합니다.

```
- (void)pickSeat:(NSInteger)seatIndex userId:(NSString *)userId callback:(ActionCallback _Nullable)callback NS_SWIFT_NAME(pickSeat(seatIndex:userId:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	NSInteger	마이크를 연결할 마이크 위치 번호.
userId	NSString	사용자 ID.

매개변수	유형	의미
callback	ActionCallback	작업 콜백.

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다. 방 주인이 청취자의 동의를 얻어야만 청취자가 마이크를 연결할 수 있는 시나리오의 경우 먼저 `sendInvitation` 을 호출하여 청취자에게 신청하고 `onInvitationAccept` 수신 후 다시 해당 함수를 호출합니다.

kickSeat

특정 사용자 마이크 끄기(방 주인 호출).

설명 :

방 주인이 특정 사용자의 마이크를 끄면 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onAnchorLeaveSeat` 이벤트를 알릴 수 있습니다.

```
- (void)kickSeat:(NSInteger)seatIndex callback:(ActionCallback _Nullable)callback
NS_SWIFT_NAME(kickSeat(seatIndex:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	NSInteger	마이크 연결을 해제할 마이크 위치 번호.
callback	ActionCallback	작업 콜백.

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다.

muteSeat

특정 마이크 위치 음소거/음소거 해제(방 주인 호출).

설명 :

특정 마이크 위치를 음소거/음소거 해제하면 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onSeatMute` 이벤트를 알릴 수 있습니다.

```
- (void)muteSeat:(NSInteger)seatIndex isMute:(BOOL)isMute callback:(ActionCallbac
k _Nullable)callback NS_SWIFT_NAME(muteSeat(seatIndex:isMute:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	NSInteger	작업 진행할 마이크 위치 번호.
isMute	BOOL	YES: 음소거, NO: 음소거 해제.
callback	ActionCallback	작업 콜백.

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다. 해당 seatIndex 자리에 있는 호스트는 자동으로 muteAudio가 호출되어 음소거/음소거 해제됩니다.

closeSeat

특정 마이크 위치 차단/차단 해제(방 주인 호출).

설명 :

방 주인이 해당 마이크 위치를 차단/차단 해제하면 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onSeatClose` 이벤트를 알릴 수 있습니다.

```
- (void)closeSeat:(NSInteger)seatIndex isClose:(BOOL)isClose callback:(ActionCall
back _Nullable)callback NS_SWIFT_NAME(closeSeat(seatIndex:isClose:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	NSInteger	작업 진행할 마이크 위치 번호.
isClose	BOOL	YES: 차단, NO: 해제.
callback	ActionCallback	작업 콜백.

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다. 해당 seatIndex 자리가 차단되고 자동으로 마이크 연결이 해제됩니다.

로컬 오디오 작업 API

startMicrophone

마이크 수집을 시작합니다.

```
- (void)startMicrophone;
```

stopMicrophone

마이크 수집 중지

```
- (void)stopMicrophone;
```

setAudioQuality

오디오 품질을 설정합니다.

```
- (void)setAudioQuality:(NSInteger)quality NS_SWIFT_NAME(setAudioQuality(quality:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
quality	NSInteger	오디오의 품질입니다. 자세한 내용은 TRTC SDK 를 참고하십시오.

muteLocalAudio

로컬 오디오를 음소거/음소거 취소합니다.

```
- (void)muteLocalAudio:(BOOL)mute NS_SWIFT_NAME(muteLocalAudio(mute:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
mute	BOOL	오디오를 음소거/음소거 취소합니다. 자세한 내용은 TRTC SDK 를 참고하십시오.

setSpeaker

스피커를 활성화합니다.

```
- (void) setSpeaker: (BOOL) userSpeaker NS_SWIFT_NAME (setSpeaker (userSpeaker:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
useSpeaker	BOOL	YES: 스피커, NO: 헤드셋.

setAudioCaptureVolume

마이크 수집 볼륨을 설정합니다.

```
- (void) setAudioCaptureVolume: (NSInteger) volume NS_SWIFT_NAME (setAudioCaptureVolume (volume:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
volume	NSInteger	수집 볼륨, 0 - 100으로 설정할 수 있으며 기본 값은 100입니다.

setAudioPlayOutVolume

재생 볼륨을 설정합니다.

```
- (void) setAudioPlayOutVolume: (NSInteger) volume NS_SWIFT_NAME (setAudioPlayOutVolume (volume:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
volume	NSInteger	재생 볼륨, 0 - 100으로 설정할 수 있으며 기본 값은 100입니다.

muteRemoteAudio

지정 사용자 음소거/음소거 해제.

```
- (void) muteRemoteAudio: (NSString *) userId mute: (BOOL) mute NS_SWIFT_NAME (muteRemoteAudio (userId:mute:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
------	----	----

매개변수	유형	의미
userId	NSString	지정된 사용자 ID.
mute	BOOL	YES: 음소거, NO: 음소거 해제.

muteAllRemoteAudio

모든 사용자 음소거/음소거 해제.

```
- (void)muteAllRemoteAudio:(BOOL)isMute NS_SWIFT_NAME(muteAllRemoteAudio(isMute:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
mute	BOOL	YES: 음소거, NO: 음소거 해제.

setVoiceEarMonitorEnable

인이어 모니터링 활성화/비활성화.

```
- (void)setVoiceEarMonitorEnable:(BOOL)enable NS_SWIFT_NAME(setVoiceEarMonitor(enable:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
enable	BOOL	YES: 인이어 모니터링 활성화, NO: 인이어 모니터링 비활성화.

배경 음악 음향 효과 관련 API

getAudioEffectManager

배경 음악 음향 효과 관리 객체 [TXAudioEffectManager](#) 획득.

```
- (TXAudioEffectManager * _Nullable)getAudioEffectManager;
```

메시지 발송 관련 API

sendRoomTextMsg

방 안에서 텍스트 메시지 발송, 일반적으로 닷글 자막 채팅에 사용.

```
- (void)sendRoomTextMsg:(NSString *)message callback:(ActionCallback _Nullable) callback NS_SWIFT_NAME (sendRoomTextMsg(message:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
message	NSString	텍스트 메시지.
callback	ActionCallback	발송 결과 콜백.

sendRoomCustomMsg

사용자 정의 텍스트 메시지를 발송합니다.

```
- (void)sendRoomCustomMsg:(NSString *)cmd message:(NSString *)message callback:(ActionCallback _Nullable) callback NS_SWIFT_NAME (sendRoomCustomMsg(cmd:message:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
cmd	NSString	명령어, 개발자가 사용자 정의할 수 있으며 주로 서로 다른 메시지 유형을 구분 하는데 사용합니다.
message	NSString	텍스트 메시지.
callback	ActionCallback	발송 결과 콜백.

초대 신호 관련 API

sendInvitation

사용자에게 초대 발송.

```
- (NSString *)sendInvitation:(NSString *)cmd
userId:(NSString *)userId
content:(NSString *)content
callback:(ActionCallback _Nullable) callback NS_SWIFT_NAME (sendInvitation(cmd:userId:content:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
cmd	NSString	서비스의 사용자 정의 명령.
userId	NSString	초대한 사용자 ID.
content	NSString	초대 내용.
callback	ActionCallback	발송 결과 콜백.

반환값:

반환값	유형	의미
inviteId	NSString	해당 초대 ID를 식별하는 데 사용.

acceptInvitation

초대 수락.

```
- (void)acceptInvitation:(NSString *)identifier callback:(ActionCallback _Nullable) callback NS_SWIFT_NAME(acceptInvitation(id:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	NSString	초대 ID.
callback	ActionCallback	발송 결과 콜백.

rejectInvitation

초대 거부.

```
- (void)rejectInvitation:(NSString *)identifier callback:(ActionCallback _Nullable) callback NS_SWIFT_NAME(rejectInvitation(id:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
------	----	----

매개변수	유형	의미
id	NSString	초대 ID.
callback	ActionCallback	발송 결과 콜백.

cancelInvitation

초대 취소.

```
- (void)cancelInvitation:(NSString *)identifier callback:(ActionCallback _Nullable)callback NS_SWIFT_NAME(cancelInvitation(id:callback:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	NSString	초대 ID.
callback	ActionCallback	발송 결과 콜백.

TRTCVoiceRoomDelegate 이벤트 콜백

일반적인 이벤트 콜백

onError

오류 콜백.

설명 :

SDK가 복구할 수 없는 오류는 반드시 수신하고 상황에 따라 인터페이스를 통해 사용자에게 적절히 안내해야 합니다.

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

매개변수는 다음과 같습니다.

매개변수	유형	의미
code	int	오류 코드.
message	NSString	오류 정보.

onWarning

경고 콜백.

```
- (void)onWarning:(int)code
message:(NSString *)message
NS_SWIFT_NAME(onWarning(code:message:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
code	int	오류 코드.
message	NSString	경고 정보.

onDebugLog

Log 콜백.

```
- (void)onDebugLog:(NSString *)message
NS_SWIFT_NAME(onDebugLog(message:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
message	NSString	로그 정보.

방 이벤트 콜백

onRoomDestroy

방 폐기 콜백. 방 주인이 방을 종료하면 방 안에 있는 모든 사용자는 해당 알림을 받게 됩니다.

```
- (void)onRoomDestroy:(NSString *)roomId
NS_SWIFT_NAME(onRoomDestroy(roomId:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	NSString	방 ID.

onRoomInfoChange

방 입장 후 해당 인터페이스를 콜백합니다. roomInfo의 정보는 방 주인이 방 생성 시 입력한 정보입니다.

```
- (void)onRoomInfoChange:(VoiceRoomInfo *)roomInfo
NS_SWIFT_NAME(onRoomInfoChange(roomInfo:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomInfo	VoiceRoomInfo	방 정보.

onUserMicrophoneMute

사용자 마이크의 음소거 여부 콜백으로 사용자가 muteLocalAudio 호출하면 방의 모든 사용자는 해당 알림을 받게 됩니다.

```
- (void)onUserMicrophoneMute:(NSString *)userId mute:(BOOL)mute
NS_SWIFT_NAME(onUserMicrophoneMute(userId:mute:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	NSString	사용자 ID.
mute	BOOL	YES: 음소거, NO: 음소거 해제.

onUserVolumeUpdate

음량 크기 알림을 활성화하여 모든 참석자의 음량 크기를 통지합니다.

```
- (void)onUserVolumeUpdate:(NSArray<TRTCVolumeInfo *> *)userVolumes totalVolume:
(NSInteger)totalVolume
NS_SWIFT_NAME(onUserVolumeUpdate(userVolumes:totalVolume:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userVolumes	NSArray	사용자 리스트.
totalVolume	NSInteger	볼륨 크기, 값: 0 - 100.

마이크 위치 콜백

onSeatListChange

모든 마이크 위치 리스트를 포함한 전체 마이크 위치 리스트를 변경합니다.

```
- (void) onSeatInfoChange: (NSArray<VoiceRoomSeatInfo *> *) seatInfoList
NS_SWIFT_NAME (onSeatListChange (seatInfoList:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatInfoList	NSArray<VoiceRoomSeatInfo>	전체 마이크 위치 리스트.

onAnchorEnterSeat

사용자 마이크 켜짐(직접 마이크 켜/방 주인이 특정 사용자 마이크 켜).

```
- (void) onAnchorEnterSeat: (NSInteger) index
user: (VoiceRoomUserInfo *) user
NS_SWIFT_NAME (onAnchorEnterSeat (index:user:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
index	NSInteger	마이크가 연결된 마이크 위치.
user	VoiceRoomUserInfo	마이크가 연결된 사용자의 세부 정보.

onAnchorLeaveSeat

사용자 마이크 꺼짐(직접 마이크 끄/방 주인이 특정 사용자 마이크 끄).

```
- (void) onAnchorLeaveSeat: (NSInteger) index
user: (VoiceRoomUserInfo *) user
```

```
NS_SWIFT_NAME (onAnchorLeaveSeat (index:user:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
index	NSInteger	연결을 해제할 마이크 위치.
user	VoiceRoomUserInfo	마이크가 연결된 사용자의 세부 정보.

onSeatMute

방 주인 마이크 비활성화.

```
- (void) onSeatMute: (NSInteger) index
isMute: (BOOL) isMute
NS_SWIFT_NAME (onSeatMute (index:isMute:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
index	NSInteger	작업 진행할 마이크 위치.
isMute	BOOL	YES: 음소거, NO: 음소거 해제.

onSeatClose

방 주인 마이크 차단.

```
- (void) onSeatClose: (NSInteger) index
isClose: (BOOL) isClose
NS_SWIFT_NAME (onSeatClose (index:isClose:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
index	NSInteger	작업 진행할 마이크 위치.
isClose	BOOL	YES: 차단, NO: 차단 해제.

청취자 입장/퇴장 이벤트 콜백

onAudienceEnter

청취자 방 입장 알림 수신.

```
- (void)onAudienceEnter:(VoiceRoomUserInfo *)userInfo
NS_SWIFT_NAME(onAudienceEnter(userInfo:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userInfo	VoiceRoomUserInfo	입장한 청취자 정보.

onAudienceExit

청취자 방 퇴장 알림 수신.

```
- (void)onAudienceExit:(VoiceRoomUserInfo *)userInfo
NS_SWIFT_NAME(onAudienceExit(userInfo:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userInfo	VoiceRoomUserInfo	퇴장한 청취자 정보.

메시지 이벤트 콜백

onRecvRoomTextMsg

텍스트 메시지를 수신합니다.

```
- (void)onRecvRoomTextMsg:(NSString *)message
userInfo:(VoiceRoomUserInfo *)userInfo
NS_SWIFT_NAME(onRecvRoomTextMsg(message:userInfo:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
message	NSString	텍스트 메시지.
userInfo	VoiceRoomUserInfo	발신자 정보.

onRecvRoomCustomMsg

사용자 정의 메시지를 수신합니다.

```
- (void)onRecvRoomCustomMsg:(NSString *)command
message:(NSString *)message
userInfo:(VoiceRoomUserInfo *)userInfo
NS_SWIFT_NAME(onRecvRoomCustomMsg(command:message:userInfo:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
command	NSString	명령어로, 개발자가 사용자 정의할 수 있으며 주로 서로 다른 메시지 유형을 구분하는데 사용합니다.
message	NSString	텍스트 메시지.
userInfo	VoiceRoomUserInfo	발신자 정보.

초대 신호 이벤트 콜백

onReceiveNewInvitation

새로운 초대 요청 수신.

```
- (void)onReceiveNewInvitation:(NSString *)identifier
inviter:(NSString *)inviter
cmd:(NSString *)cmd
content:(NSString *)content
NS_SWIFT_NAME(onReceiveNewInvitation(id:inviter:cmd:content:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	NSString	초대 ID.
inviter	NSString	초대한 사용자 ID.
cmd	NSString	서비스 지정 명령어. 개발자가 사용자 정의함.
content	NSString	서비스에서 지정한 내용.

onInviteeAccepted

초대된 사용자가 초대 수락.

```
- (void)onInviteeAccepted:(NSString *)identifier
invitee:(NSString *)invitee
NS_SWIFT_NAME(onInviteeAccepted(id:invitee:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	NSString	초대 ID.
invitee	NSString	초대된 사용자 ID.

onInviteeRejected

초대된 사용자가 초대 거부.

```
- (void)onInviteeRejected:(NSString *)identifier
invitee:(NSString *)invitee
NS_SWIFT_NAME(onInviteeRejected(id:invitee:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	NSString	초대 ID.
invitee	NSString	초대된 사용자 ID.

onInvitationCancelled

초대한 사용자가 초대 취소.

```
- (void)onInvitationCancelled:(NSString *)identifier
invitee:(NSString *)invitee NS_SWIFT_NAME(onInvitationCancelled(id:invitee:));
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	NSString	초대 ID.
inviter	NSString	초대한 사용자 ID.

TRTCVoiceRoom(Android)

최종 업데이트 날짜: : 2022-07-19 16:21:32

TRTCVoiceRoom은 Tencent Cloud의 Real-Time Communication(TRTC)과 Instant Messaging(IM) 서비스를 기반으로 구성되며, 다음 기능을 지원합니다.

- 방 주인이 신규 음성 채팅방을 생성하면 청취자가 음성 채팅방에 입장하여 청취 및 인터랙션.
- 방 주인이 청취자에게 마이크 켜기 요청 또는 마이크가 켜진 사용자의 마이크 강제 끄기.
- 방 주인의 자리 차단 및 청취자 마이크 연결 신청 차단.
- 청취자가 마이크 켜기를 신청하여 마이크가 켜진 호스트가 될 수 있고, 다른 사람들과 음성으로 인터랙션할 수 있으며, 언제든지 마이크를 끄고 일반 청취자가 될 수 있습니다.
- 다양한 텍스트 메시지 및 사용자 정의 메시지를 지원합니다. 사용자 정의 메시지를 이용해 댓글 자막, 좋아요, 선물 기능 등을 구현할 수 있습니다.

설명 :

TUIKit 시리즈 컴포넌트는 Tencent Cloud의 두 가지 기본 PaaS 서비스, 즉 [Tencent Real-Time Communication](#) 및 [Instant Messaging](#)을 사용합니다. TRTC를 활성화하면 IM과 IM SDK 평가판(100 DAU만 지원)이 자동으로 활성화됩니다. IM 과금 내역은 [요금 안내](#)를 참고하십시오.

TRTCVoiceRoom은 오픈 소스 Class로, Tencent Cloud의 두 가지 클로즈드 소스 SDK에 종속됩니다. 자세한 구현 방법은 [음성 채팅방\(Android\)](#)을 참고하십시오.

- TRTC SDK: [TRTC SDK](#)를 저지연 음성 채팅 컴포넌트로 사용합니다.
- IM SDK: [IM SDK](#)의 AVChatroom을 사용해 채팅방 기능을 구현하며, IM 속성 인터페이스를 통해 마이크 위치 리스트 등 방 정보를 저장하고 초대 신호를 마이크 켜기 신청/마이크 넘기기에 사용할 수 있습니다.

TRTCVoiceRoom API 개요

SDK 기본 함수

API	설명
sharedInstance	컴포넌트 싱글톤 가져오기.
destroySharedInstance	컴포넌트 싱글톤 폐기.
setDelegate	이벤트 콜백 설정.

API	설명
setDelegateHandler	이벤트 콜백이 있는 스레드 설정.
login	로그인.
logout	로그아웃.
setSelfProfile	개인 정보 수정.

방 관련 API

API	설명
createRoom	방 생성(방 주인 호출), 방이 없는 경우 시스템에서 자동으로 새로운 방 생성.
destroyRoom	방 폐기(방 주인 호출).
enterRoom	방 입장(청취자 호출).
exitRoom	방 퇴장(청취자 호출).
getRoomInfoList	방 리스트의 세부 정보 획득.
getUserInfoList	지정 <code>userId</code> 의 사용자 정보 획득. null인 경우 방 안에 있는 모든 사용자 정보 획득.

마이크 위치 관리 API

API	설명
enterSeat	마이크 연결(청취자와 방 주인 모두 호출 가능).
moveSeat	마이크 위치 이동(마이크 연결된 호스트 호출 가능).
leaveSeat	마이크 연결 해제(호스트 호출).
pickSeat	마이크 넘기기(방 주인 호출).
kickSeat	마이크 강제 끄기(방 주인 호출).
muteSeat	특정 마이크 위치 음소거/음소거 해제(방 주인 호출).
closeSeat	특정 마이크 위치 차단/차단 해제(방 주인 호출).

로컬 오디오 작업 API

API	설명
startMicrophone	마이크 수집 시작.
stopMicrophone	마이크 수집 중지.
setAudioQuality	오디오 품질 설정.
muteLocalAudio	로컬 음소거 활성화/비활성화.
setSpeaker	스피커 활성화 설정.
setAudioCaptureVolume	마이크 수집 음량 설정.
setAudioPlayoutVolume	재생 볼륨 설정.
setVoiceEarMonitorEnable	인이어 모니터링 활성화/비활성화.

원격 사용자 오디오 작업 API

API	설명
muteRemoteAudio	특정 사용자 음소거/음소거 해제.
muteAllRemoteAudio	모든 사용자 음소거/음소거 해제.

배경 음악 음향 효과 관련 API

API	설명
getAudioEffectManager	배경 음악 음향 효과 관리 객체 TXAudioEffectManager 가져오기.

메시지 발송 관련 API

API	설명
sendRoomTextMsg	방 안에서 텍스트 메시지 발송, 일반적으로 댓글 자막 채팅에 사용.
sendRoomCustomMsg	사용자 정의 텍스트 메시지 발송.

초대 신호 관련 API

API	설명
sendInvitation	사용자에게 초대 발송.

API	설명
acceptInvitation	초대 수락.
rejectInvitation	초대 거부.
cancelInvitation	초대 취소.

TRTCVoiceRoomDelegate API 개요

일반적인 이벤트 콜백

API	설명
onError	오류 콜백.
onWarning	경고 콜백.
onDebugLog	Log 콜백.

방 이벤트 콜백 API

API	설명
onRoomDestroy	방 폐기 콜백.
onRoomInfoChange	음성 채팅방 정보 변경 콜백.
onUserVolumeUpdate	사용자 통화 볼륨 콜백.

마이크 위치 변경 콜백

API	설명
onSeatListChange	전체 마이크 위치 리스트 변경.
onAnchorEnterSeat	사용자 마이크 켜짐(직접 마이크 켜/방 주인 특정 사용자 마이크 켜).
onAnchorLeaveSeat	사용자 마이크 꺼짐(직접 마이크 끄/방 주인이 특정 사용자 마이크 끄).
onSeatMute	방 주인 마이크 음소거.
onUserMicrophoneMute	사용자 마이크 음소거 여부.

API	설명
onSeatClose	방 주인 마이크 차단.

청취자 입장/퇴장 이벤트 콜백

API	설명
onAudienceEnter	청취자 입장 알림 수신.
onAudienceExit	청취자 퇴장 알림 수신.

메시지 이벤트 콜백

API	설명
onRecvRoomTextMsg	텍스트 메시지 수신.
onRecvRoomCustomMsg	사용자 정의 메시지 수신.

신호 이벤트 콜백

API	설명
onReceiveNewInvitation	새로운 초대 요청 수신.
onInviteeAccepted	초대된 사용자가 초대 수락.
onInviteeRejected	초대된 사용자가 초대 거절.
onInvitationCancelled	초대한 사용자가 초대 취소.

기본 SDK API

sharedInstance

[TRTCVoiceRoom](#) 싱글톤 객체를 가져옵니다.

```
public static synchronized TRTCVoiceRoom sharedInstance(Context context);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
context	Context	Android 컨텍스트로, 내부가 ApplicationContext로 전환되어 시스템 API 호출에 사용됩니다.

destroySharedInstance

TRTCVoiceRoom 싱글톤 객체를 폐기합니다.

설명 :

인스턴스 폐기 후에는 외부에 캐시된 TRTCVoiceRoom 인스턴스를 다시 사용할 수 없으며, 다시 [sharedInstance](#)를 호출해 새로운 인스턴스를 획득해야 합니다.

```
public static void destroySharedInstance ();
```

setDelegate

TRTCVoiceRoom 이벤트 콜백은 TRTCVoiceRoomDelegate를 통해 TRTCVoiceRoom의 다양한 상태 알림을 받아볼 수 있습니다.

```
public abstract void setDelegate (TRTCVoiceRoomDelegate delegate);
```

설명 :

setDelegate는 TRTCVoiceRoom의 프록시의 콜백입니다.

setDelegateHandler

이벤트 콜백이 존재하는 스레드를 설정합니다.

```
public abstract void setDelegateHandler (Handler handler);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
handler	Handler	TRTCVoiceRoom의 각종 상태를 통지하며, 지정한 handler 스레드로 배포합니다.

login

로그인

```
public abstract void login(int sdkAppId,
String userId, String userSig,
TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
sdkAppId	int	TRTC 콘솔 >애플리케이션 관리> 애플리케이션 정보에서 SDKAppID를 확인 할 수 있습니다.
userId	String	현재 사용자 ID입니다. 문자열 유형은 영어 알파벳(a-z, A-Z), 숫자(0-9), 대시 부호(-), 언더바(_)만 허용됩니다.
userSig	String	Tencent Cloud가 설계한 일종의 보안 서명입니다. 획득 방식은 UserSig 계산 및 사용 방법 을 참고하십시오.
callback	ActionCallback	로그인 콜백이며, 성공 시 code는 0입니다.

logout

로그아웃합니다.

```
public abstract void logout(TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callback	ActionCallback	로그아웃 콜백이며, 성공 시 code는 0입니다.

setSelfProfile

개인 정보를 수정합니다.

```
public abstract void setSelfProfile(String userName, String avatarURL, TRTCVoiceR
oomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
------	----	----

매개변수	유형	의미
userName	String	닉네임.
avatarURL	String	프로필 사진 주소.
callback	ActionCallback	개인 프로필 정보 설정 콜백이며, 성공 시 code는 0입니다.

방 관련 API

createRoom

방 생성(방 주인 호출).

```
public abstract void createRoom(int roomId, TRTCVoiceRoomDef.RoomParam roomParam,
    TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	int	방 식별 번호이며, 귀하가 할당하고 통합 관리합니다. 여러 개의 roomId를 1개의 음성 채팅방 리스트로 통합할 수 있으며, Tencent Cloud는 현재 음성 채팅방 리스트 관리 서비스를 제공하지 않으므로 직접 관리하시기 바랍니다.
roomParam	TRTCCreateRoomParam	방 정보입니다. 방 이름, 마이크 위치 정보, 썸네일 정보 등과 같이 방을 설명하는 데 사용됩니다. 마이크 위치 관리가 필요한 경우 방의 마이크 위치 개수를 설정해야 합니다.
callback	ActionCallback	방 생성 결과 콜백이며, 성공 시 code는 0입니다.

방 주인의 정상적인 방송 시작 호출 프로세스는 다음과 같습니다.

- 방 주인이 `createRoom` 을 호출하여 새로운 음성 채팅방을 생성합니다. 이 때 방 ID, 마이크 연결 시 방장 확인 필요 여부, 마이크 위치 개수 등 방 속성 정보를 전송합니다.
- 방 주인이 방 생성 후 `enterSeat` 을 호출하여 자리에 입장합니다.
- 방 주인이 컴포넌트의 `onSeatListChange` 마이크 위치 리스트 변경 이벤트 알림을 수신합니다. 이때 마이크 위치 리스트의 변경 내용을 UI 인터페이스에 새로고침할 수 있습니다.
- 방 주인이 마이크 위치 리스트에 사용자가 입장할 때 `onAnchorEnterSeat` 이벤트 알림 또한 수신하며, 이때 자동으로 마이크 수집이 활성화됩니다.

destroyRoom

방 폐기(방 주인 호출). 방 주인은 방 생성 후 해당 함수를 호출해 방을 폐기할 수 있습니다.

```
public abstract void destroyRoom(TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callback	ActionCallback	방 폐기 결과 콜백이며, 성공 시 code는 0입니다.

enterRoom

방 입장(청취자 호출).

```
public abstract void enterRoom(int roomId, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	int	방 식별 번호.
callback	ActionCallback	방 입장 결과 콜백이며, 성공 시 code는 0입니다.

청취자가 방에 입장하여 청취하는 정상적인 호출 프로세스는 다음과 같습니다.

1. 청취자가 귀하의 서버에서 최신 음성 채팅방 리스트를 획득하며, 여기에는 여러 음성 채팅방의 roomId 및 방 정보가 포함될 수 있습니다.
2. 청취자가 음성 채팅방 1개를 선택하고 `enterRoom` 을 호출하여 해당 방으로 입장합니다.
3. 방 입장 후 컴포넌트의 `onRoomInfoChange` 방 속성 변경 이벤트 알림을 수신합니다. 이때 UI에 방 이름 표시, 마이크를 켤 때 방 주인에게 동의 요청 필요 여부 기록 등 방의 속성을 기록할 수 있으며 그에 해당하는 변경이 가능합니다.
4. 방 입장 후 컴포넌트의 `onSeatListChange` 마이크 위치 리스트 변경 이벤트 알림을 수신합니다. 이때 마이크 위치 리스트의 변경 내용을 UI 인터페이스에 새로고침할 수 있습니다.
5. 방 입장 후 마이크 위치 리스트에 호스트 입장 `onAnchorEnterSeat` 이벤트 알림도 수신합니다.

exitRoom

방 퇴장.

```
public abstract void exitRoom(TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callback	ActionCallback	방 퇴장 결과 콜백이며, 성공 시 code는 0입니다.

getRoomInfoList

방 리스트의 세부 정보를 획득합니다. 방 이름, 방 썸네일은 방 주인이 `createRoom()` 생성 시 `roomInfo`를 통해 설정할 수 있습니다.

설명 :

방 리스트 및 방 정보를 모두 직접 관리하는 경우 해당 함수는 생략할 수 있습니다.

```
public abstract void getRoomInfoList(List<Integer> roomIdList, TRTCVoiceRoomCallback.RoomInfoCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomIdList	List<Integer>	방 번호 리스트.
callback	RoomInfoCallback	방 세부 정보 콜백.

getUserInfoList

지정 `userId`의 사용자 정보 획득.

```
public abstract void getUserInfoList(List<String> userIdList, TRTCVoiceRoomCallback.UserListCallback userlistcallback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userIdList	List<String>	획득해야 할 사용자 ID 리스트입니다. null인 경우 방 안에 있는 모든 사용자 정보를 획득합니다.
userlistcallback	UserListCallback	사용자 세부 정보 콜백.

마이크 위치 관리 API

enterSeat

직접 마이크 켜기(청취자 및 방 주인 모두 호출 가능).

설명 :

마이크 연결 완료 후, 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onAnchorEnterSeat` 이벤트 알림을 수신합니다.

```
public abstract void enterSeat(int seatIndex, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	int	마이크를 연결할 마이크 위치 번호.
callback	ActionCallback	작업 콜백.

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다. 청취자의 마이크 연결에 방 주인의 동의가 필요한 시나리오의 경우, 먼저 `sendInvitation` 을 호출하여 방 주인에게 신청하고 `onInvitationAccept` 수신 후 다시 해당 함수를 호출합니다.

moveSeat

마이크 위치 이동(마이크 연결된 호스트 호출 가능).

설명 :

마이크 위치 이동 완료 후 방 안의 모든 구성원은 'onSeatListChange', 'onAnchorLeaveSeat' 및 'onAnchorEnterSeat'의 이벤트 알림을 받게 됩니다. (호스트 호출 후 마이크 좌석 번호 정보만 수정되며, 사용자의 호스트 신분은 변경되지 않습니다.)

```
public abstract int moveSeat(int seatIndex, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	int	마이크를 이동할 마이크 위치 번호.
callback	ActionCallback	작업 콜백.

반환값:

반환값	유형	의미
code	int	마이크 이동 작동 결과(0은 성공, 그 외는 실패, 10001은 인터페이스 호출 빈도 제한).

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다. 청취자의 마이크 연결에 방 주인의 동의가 필요한 시나리오의 경우, 먼저 `sendInvitation` 을 호출하여 방 주인에게 신청하고 `onInvitationAccept` 수신 후 다시 해당 함수를 호출합니다.

leaveSeat

직접 마이크 끄(호스트 호출).

설명 :

마이크 연결 해제 완료 후, 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onAnchorLeaveSeat` 이벤트 알림을 수신합니다.

```
public abstract void leaveSeat (TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
callback	ActionCallback	작업 콜백.

pickSeat

특정 사용자 마이크 켜기(방 주인 호출).

설명 :

방 주인이 마이크를 연결할 사용자를 지정하면 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onAnchorEnterSeat` 이벤트 알림을 수신합니다.

```
public abstract void pickSeat(int seatIndex, String userId, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	int	마이크를 연결할 마이크 위치 번호.
userId	String	사용자 ID.
callback	ActionCallback	작업 콜백.

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다. 방 주인이 청취자의 동의를 얻어야만 청취자가 마이크를 연결할 수 있는 시나리오의 경우, 먼저 `sendInvitation` 을 호출하여 청취자에게 신청하고 `onInvitationAccept` 수신 후 다시 해당 함수를 호출합니다.

kickSeat

특정 사용자 마이크 끄기(방 주인 호출).

설명 :

방 주인이 특정 사용자의 마이크를 끄면 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onAnchorLeaveSeat` 이벤트 알림을 수신합니다.

```
public abstract void kickSeat(int seatIndex, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	int	마이크 연결을 해제할 마이크 위치 번호.
callback	ActionCallback	작업 콜백.

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다.

muteSeat

특정 마이크 위치 음소거/음소거 해제(방 주인 호출).

설명 :

특정 마이크 위치를 음소거/음소거 해제하면 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onSeatMute` 이벤트를 알릴 수 있습니다.

```
public abstract void muteSeat(int seatIndex, boolean isMute, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	int	작업을 진행할 마이크 위치 번호.
isMute	boolean	true: 음소거, false: 음소거 해제.
callback	ActionCallback	작업 콜백.

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다. 해당 seatIndex 자리에 있는 호스트는 자동으로 muteAudio가 호출되어 음소거/음소거 해제됩니다.

closeSeat

특정 마이크 위치 차단/차단 해제(방 주인 호출).

설명 :

방 주인이 해당 마이크 위치를 차단/차단 해제하면 방 안에 있는 모든 사용자가 `onSeatListChange` 및 `onSeatClose` 이벤트를 알릴 수 있습니다.

```
public abstract void closeSeat(int seatIndex, boolean isClose, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatIndex	int	작업을 진행할 마이크 위치 번호.

매개변수	유형	의미
isClose	boolean	true: 차단, false: 차단 해제.
callback	ActionCallback	작업 콜백.

해당 인터페이스를 호출하면 마이크 위치 리스트가 즉시 수정됩니다. 해당 `seatIndex` 자리가 차단되고 자동으로 마이크 연결이 해제됩니다.

로컬 오디오 작업 API

startMicrophone

마이크 수집을 시작합니다.

```
public abstract void startMicrophone();
```

stopMicrophone

마이크 수집 중지

```
public abstract void stopMicrophone();
```

setAudioQuality

오디오 품질을 설정합니다.

```
public abstract void setAudioQuality(int quality);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
quality	int	오디오의 품질입니다. 자세한 내용은 TRTC SDK 를 참고하십시오.

muteLocalAudio

로컬 오디오를 음소거/음소거 취소합니다.

```
public abstract void muteLocalAudio(boolean mute);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
mute	boolean	오디오를 음소거/음소거 취소합니다. 자세한 내용은 TRTC SDK 를 참고하십시오.

setSpeaker

스피커를 활성화합니다.

```
public abstract void setSpeaker(boolean useSpeaker);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
useSpeaker	boolean	true: 스피커, false: 핸드셋.

setAudioCaptureVolume

마이크 수집 볼륨을 설정합니다.

```
public abstract void setAudioCaptureVolume(int volume);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
volume	int	수집 볼륨, 범위: 0 - 100, 기본값: 100.

setAudioPlayoutVolume

재생 볼륨을 설정합니다.

```
public abstract void setAudioPlayoutVolume(int volume);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
volume	int	재생 볼륨, 범위: 0 - 100, 기본값: 100.

muteRemoteAudio

지정 사용자 음소거/음소거 해제.

```
public abstract void muteRemoteAudio(String userId, boolean mute);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	String	지정 사용자 ID.
mute	boolean	true: 음소거 켜기, false: 음소거 끄기.

muteAllRemoteAudio

모든 사용자 음소거/음소거 해제.

```
public abstract void muteAllRemoteAudio(boolean mute);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
mute	boolean	true: 음소거 켜기, false: 음소거 끄기.

setVoiceEarMonitorEnable

인이어 모니터링 활성화/비활성화.

```
public abstract void setVoiceEarMonitorEnable(boolean enable);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
enable	boolean	true: 인이어 모니터링 활성화; false: 인이어 모니터링 비활성화.

배경 음악 음향 효과 관련 API

getAudioEffectManager

배경 음악 음향 효과 관리 객체 [TXAudioEffectManager](#) 획득.

```
public abstract TXAudioEffectManager getAudioEffectManager();
```

메시지 발송 관련 API

sendRoomTextMsg

방 안에서 텍스트 메시지 발송, 일반적으로 댓글 자막 채팅에 사용.

```
public abstract void sendRoomTextMsg(String message, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
message	String	텍스트 메시지.
callback	ActionCallback	발송 결과 콜백.

sendRoomCustomMsg

사용자 정의 텍스트 메시지를 발송합니다.

```
public abstract void sendRoomCustomMsg(String cmd, String message, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
cmd	String	명령어, 개발자가 사용자 정의할 수 있으며 주로 서로 다른 메시지 유형을 구분하는 데 사용됩니다.
message	String	텍스트 메시지.
callback	ActionCallback	발송 결과 콜백.

초대 신호 관련 API

sendInvitation

사용자에게 초대 발송.

```
public abstract String sendInvitation(String cmd, String userId, String content, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
cmd	String	서비스의 사용자 정의 명령.
userId	String	초대한 사용자 ID.
content	String	초대 내용.
callback	ActionCallback	발송 결과 콜백.

반환값:

반환값	유형	의미
inviteId	String	해당 초대 ID를 식별하는 데 사용.

acceptInvitation

초대 수락.

```
public abstract void acceptInvitation(String id, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	String	초대 ID.
callback	ActionCallback	발송 결과 콜백.

rejectInvitation

초대 거부.

```
public abstract void rejectInvitation(String id, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	String	초대 ID.

매개변수	유형	의미
callback	ActionCallback	발송 결과 콜백.

cancelInvitation

초대 취소.

```
public abstract void cancelInvitation(String id, TRTCVoiceRoomCallback.ActionCallback callback);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	String	초대 ID.
callback	ActionCallback	발송 결과 콜백.

TRTCVoiceRoomDelegate 이벤트 콜백

일반적인 이벤트 콜백

onError

오류 콜백.

설명 :

SDK가 복구할 수 없는 오류는 반드시 수신하고 상황에 따라 인터페이스를 통해 사용자에게 적절히 안내해야 합니다.

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

매개변수는 다음과 같습니다.

매개변수	유형	의미
code	int	오류 코드.

매개변수	유형	의미
message	String	오류 정보.

onWarning

경고 콜백.

```
void onWarning(int code, String message);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
code	int	오류 코드.
message	String	경고 정보.

onDebugLog

Log 콜백.

```
void onDebugLog(String message);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
message	String	로그 정보.

방 이벤트 콜백

onRoomDestroy

방 폐기 콜백. 방 주인이 방을 종료하면 방 안에 있는 모든 사용자는 해당 알림을 받게 됩니다.

```
void onRoomDestroy(String roomId);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomId	String	방 ID.

onRoomInfoChange

방 입장 후 해당 인터페이스를 콜백합니다. roomInfo의 정보는 방 주인이 방 생성 시 입력한 정보입니다.

```
void onRoomInfoChange (TRTCVoiceRoomDef.RoomInfo roomInfo);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
roomInfo	RoomInfo	방 정보.

onUserMicrophoneMute

사용자 마이크의 음소거 여부 콜백으로 사용자가 muteLocalAudio 호출하면 방의 모든 사용자는 해당 알림을 받게 됩니다.

```
void onUserMicrophoneMute (String userId, boolean mute);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userId	String	사용자 ID.
mute	boolean	true: 음소거, false: 음소거 해제.

onUserVolumeUpdate

음량 크기 알림을 활성화하여 모든 참석자의 음량 크기를 통지합니다.

```
void onUserVolumeUpdate (List<TRTCCLoudDef.TRTCVolumeInfo> userVolumes, int totalVolume);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userVolumes	ListList<trtcclouddef.trtcvolumeinfo>	사용자 리스트.
totalVolume	int	볼륨 크기, 값: 0 - 100.

마이크 위치 콜백

onSeatListChange

모든 마이크 위치 리스트를 포함한 전체 마이크 위치 리스트를 변경합니다.

```
void onSeatListChange (List<SeatInfo> seatInfoList);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
seatInfoList	List<SeatInfo>	전체 마이크 위치 리스트.

onAnchorEnterSeat

사용자 마이크 켜짐(직접 마이크 켜/방 주인이 특정 사용자 마이크 켜).

```
void onAnchorEnterSeat (int index, TRTCVoiceRoomDef.UserInfo user);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
index	int	마이크가 연결된 마이크 위치.
user	UserInfo	마이크가 연결된 사용자의 세부 정보.

onAnchorLeaveSeat

사용자 마이크 꺼짐(직접 마이크 끄/방 주인이 특정 사용자 마이크 끄).

```
void onAnchorLeaveSeat (int index, TRTCVoiceRoomDef.UserInfo user);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
index	int	연결을 해제할 마이크 위치.
user	UserInfo	마이크가 연결된 사용자의 세부 정보.

onSeatMute

방 주인 마이크 비활성화.

```
void onSeatMute (int index, boolean isMute);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
index	int	작업 진행할 마이크 위치.
isMute	boolean	true: 음소거, false: 음소거 해제.

onSeatClose

방 주인 마이크 차단.

```
void onSeatClose(int index, boolean isClose);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
index	int	작업 마이크 위치.
isClose	boolean	true: 차단, false: 차단 해제.

청취자 입장/퇴장 이벤트 콜백

onAudienceEnter

청취자 방 입장 알림 수신.

```
void onAudienceEnter(TRTCVoiceRoomDef.UserInfo userInfo);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userInfo	UserInfo	입장한 청취자 정보.

onAudienceExit

청취자 방 퇴장 알림 수신.

```
void onAudienceExit(TRTCVoiceRoomDef.UserInfo userInfo);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
userInfo	UserInfo	퇴장한 청취자 정보.

메시지 이벤트 콜백

onRecvRoomTextMsg

텍스트 메시지를 수신합니다.

```
void onRecvRoomTextMsg(String message, TRTCVoiceRoomDef.UserInfo userInfo);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
message	String	텍스트 메시지.
userInfo	UserInfo	발신자 정보.

onRecvRoomCustomMsg

사용자 정의 메시지를 수신합니다.

```
void onRecvRoomCustomMsg(String cmd, String message, TRTCVoiceRoomDef.UserInfo userInfo);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
cmd	String	명령어. 개발자가 사용자 정의할 수 있으며, 주로 서로 다른 메시지 유형을 구분하는데 사용됩니다.
message	String	텍스트 메시지.
userInfo	UserInfo	발신자 정보.

초대 신호 이벤트 콜백

onReceiveNewInvitation

새로운 초대 요청 수신.

```
void onReceiveNewInvitation(String id, String inviter, String cmd, String content);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	String	초대 ID.
inviter	String	초대한 사용자 ID.
cmd	String	서비스 지정 명령어. 개발자가 사용자 정의함.
content	String	서비스에서 지정한 내용.

onInviteeAccepted

초대된 사용자가 초대 수락.

```
void onInviteeAccepted(String id, String invitee);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	String	초대 ID.
invitee	String	초대된 사용자 ID.

onInviteeRejected

초대된 사용자가 초대 거부.

```
void onInviteeRejected(String id, String invitee);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	String	초대 ID.
invitee	String	초대된 사용자 ID.

onInvitationCancelled

초대한 사용자가 초대 취소.

```
void onInvitationCancelled(String id, String inviter);
```

매개변수는 다음과 같습니다.

매개변수	유형	의미
id	String	초대 ID.
inviter	String	초대한 사용자 ID.
</trtcclouddef.trtcvolumeinfo>		