

Game Multimedia Engine

고급 기능 개발 가이드

제품 문서



Tencent Cloud

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:

고급 기능 개발 가이드

서버 녹음

전체 녹음

사용자 지정 녹음

녹음 콜백 설명

클라이언트 녹음

실시간 음성 채팅 녹음

레인지 보이스

3D 음향 효과

음향 효과 및 반주

음성 변조

실시간 음성 반주

실시간 사운드 이퀄라이저

실시간 노래방 기능

네트워크 오디오 스트림 포워딩 라우팅

사용자 지정 메시지 채널

회사 방화벽 제한을 해결하는 방법

Language Parameter Reference List

GME 채팅방 관리 통합

고급 기능 개발 가이드

서버 녹음

전체 녹음

최종 업데이트 날짜: : 2024-01-18 16:00:04

본문은 전체 녹음 방식으로 **GME 서버 녹음** 기능을 빠르게 통합할 수 있도록 도와줍니다.

시나리오

GME는 실시간 음성 채팅 스트림을 위한 **서버 녹음** 기능을 제공하여 개발자가 콘텐츠 보관, 콘텐츠 관리 및 콘텐츠 재생산과 같은 시나리오를 구현할 수 있도록 지원합니다. 전체 녹음: 애플리케이션 내 전체 음성 녹음과 방별 혼합 스트림, 사용자별 단일 스트림을 지원합니다. 사용자 지정 녹음: 지정된 방의 방별 혼합 스트림, 사용자별 단일 스트림을 지원합니다. 녹음된 오디오 파일은 귀하의 계정에 있는 **COS** 서비스에 저장됩니다.

본문은 **전체 녹음**의 개발 및 통합 방법에 대해서만 설명합니다. 애플리케이션에 사용자 지정 녹음을 활성화하려면 [개발 가이드 - 사용자 지정 녹음](#)을 참고하십시오.

주의사항 :

GME 서버 녹음 기능을 사용하면 녹음 과정에서 GME에서 녹음 서비스 요금이 발생합니다. GME 글로벌 포털 녹음 서비스는 2023년 4월 1일부터 정식 과금될 예정이며, 자세한 내용은 [GME 구매 가이드](#)에 공지됩니다.

녹음된 파일은 Tencent Cloud 계정의 **COS** 서비스에 저장되며 저장 용량, 저장 기간 및 액세스 빈도 등 고객의 구체적인 사용 방식에 따라 **COS** 청구서가 발행됩니다. 자세한 내용은 [COS 과금 개요](#)를 참고하십시오.

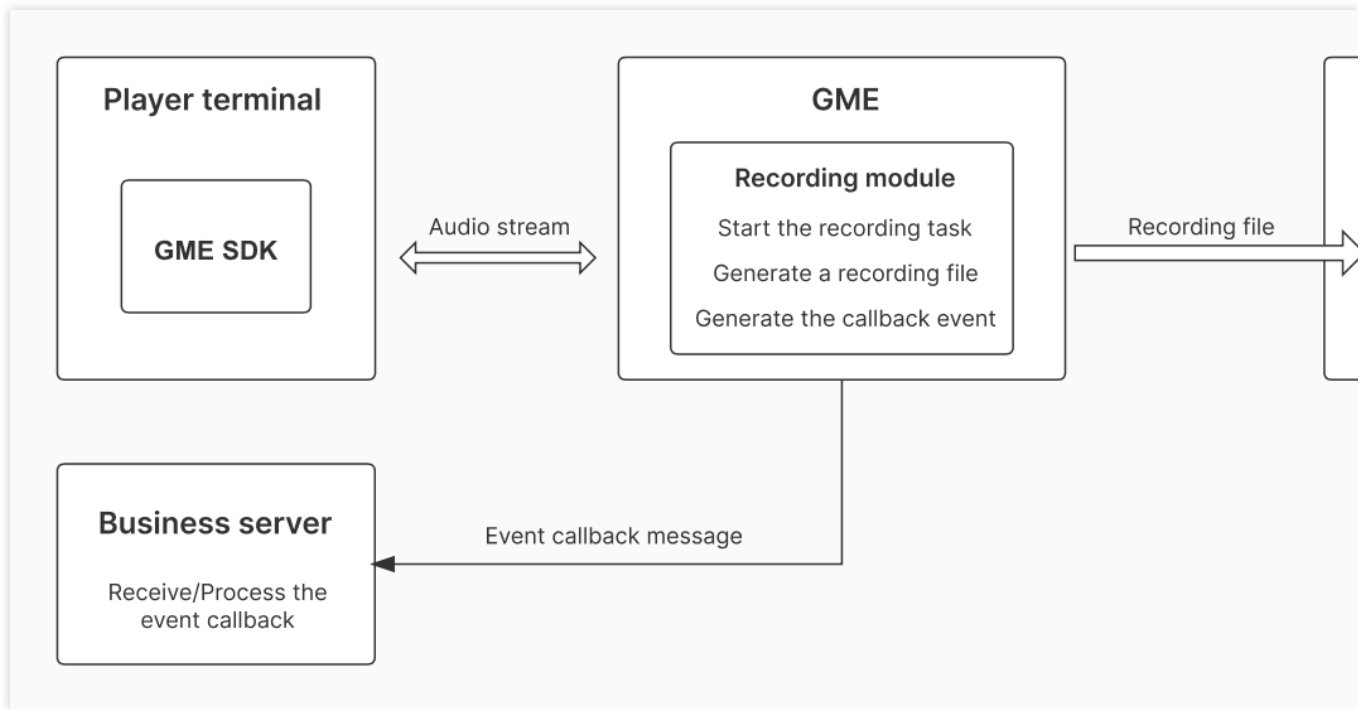
전제 조건

실시간 음성 채팅 서비스 활성화 완료: [서비스 활성화](#)를 참고하십시오.

서버 녹음 서비스 활성화 완료: 현재 서버 녹음 기능은 얼로우리스트에 있는 사용자에게 제공됩니다. 얼로우리스트를 활성화하려면 당사에 문의하십시오.

GME SDK 통합 완료: 핵심 API 및 실시간 음성 채팅 API 통합을 포함합니다. 자세한 내용은 [Native SDK 빠른 통합](#), [Unity SDK 빠른 통합](#), [Unreal SDK 빠른 통합](#)을 참고하십시오.

서비스 아키텍처



기능 설명

1. 녹음 범위

전체 녹음을 활성화하면 모든 실시간 음성 채팅방이 녹음됩니다. 방 혼합 스트림만 녹음, 사용자 단일 스트림만 녹음 또는 단일 스트림과 혼합 스트림 동시 녹음을 구성할 수 있습니다.

2. 녹음 메커니즘

녹음 작업 실행 메커니즘

첫 번째 사용자가 방에 입장한 후 녹음 작업 실행을 트리거합니다.

녹음 작업 종료 메커니즘

마지막 사용자가 방에서 퇴장한 후 녹음 작업 종료를 트리거합니다.

녹음 작업 분할 메커니즘

단일 오디오 파일 길이가 2시간이 되면 자동으로 오디오 파일을 분할합니다

사용자가 마이크를 끄면 사용자의 단일 스트림 녹음이 자동으로 분할되고 사용자가 마이크를 켜면 새로운 세그먼트가 시작됩니다

녹음 작업이 비정상적으로 중단된 경우 작업이 자동으로 재연결된 후 새로운 파일이 시작됩니다

녹음 작업 이벤트 알림 메커니즘

녹음 작업 이벤트는 콜백 메커니즘을 통해 설정한 콜백 주소로 알림됩니다. **녹음 시작**, **녹음 중지**, **녹음 파일 업로드 완료** 이벤트가 발생하면 콜백 알림을 받게 됩니다.

콜백 정보에 대한 자세한 내용은 [녹음 콜백](#)을 참고하십시오.

3. 저장 위치

GME 서버 녹음 파일은 귀하의 계정에 있는 **COS** 서비스의 지정된 버킷에 저장됩니다. 버킷의 리전은 귀하가 지정합니다. 선택 가능한 리전 목록은 [리전 및 액세스 도메인](#)을 참고하십시오.

4. 녹음 파일 형식

.mp3

5. 녹음 파일 이름 생성 규칙

사용자 단일 스트림 녹음 파일: bizid_roomid_userid/\${작업 시작 시간}_\${id}_audio.mp3

방 혼합 스트림 녹음 파일: bizid_roomid/\${taskid}_\${작업 시작 시간}_\${id}_audio.mp3

bizid: GME 애플리케이션 ID입니다. [GME 콘솔](#)에서 가져올 수 있습니다.

roomid: 음성 방 ID입니다. 실시간 음성 채팅 서비스 이용 시 정의되어 GME SDK로 전달됩니다.

userid: 플레이어 ID입니다. 실시간 음성 채팅 서비스 이용 시 정의되어 GME SDK로 전달됩니다.

taskid: GME 녹음 서비스에서 생성한 녹음 작업 ID입니다. 각 녹음 작업에는 고유한 작업 ID가 있습니다.

id: 녹음 작업을 위한 세그먼트의 일련 번호입니다. 초기 일련 번호는 0입니다.

통합 단계

1. 콘솔에서 녹음 서비스 구성 완료비즈니스
2. 녹음 작업 콜백 수신(선택 사항)비즈니스
3. 녹음 파일 조회/관리비즈니스

1단계: 콘솔에서 녹음 서비스 구성 완료

녹음 서비스 활성화/비활성화

[GME 콘솔](#)에 로그인하여 [서비스 관리] 메뉴로 들어가 녹음 서비스를 활성화할 애플리케이션의 [설정]을 클릭하여 애플리케이션 상세 페이지로 이동합니다. 페이지의 [음성 녹음 서비스]에서 **수정**을 클릭합니다.

Voice Recording Service

Recording enable/disable Enable Disable

Store recording to [Bind](#)

Callback URL [Modify](#)

Recording mode Custom recording Full recording

Single-stream recording by user Mixed-stream recording by room

I have read and agree to [Billing Description for Voice Recording Service](#).

Save

Cancel

녹음 스위치를 **ON**으로 설정합니다.

녹음 서비스를 처음 시작할 때 GME는 귀하의 **COS 서비스**에 액세스하기 위해 귀하에게 서비스 승인을 신청합니다.

팝업 창 페이지에서 권한 부여에 동의해주셔야 정상적으로 서버 녹음 서비스를 이용하실 수 있습니다.

Service Authorization

Other cloud service features will be used when performing operations related to this service.

Please create Service-Linked roles for **GME** and authorize the it to use other cloud services. The relevant information is as

Role Name	GME_QCSLinkedRoleInGameMedia (Service-Linked roles)
Role Description	The current role is the GME service linked role, which will access your other service resources with scope of the permissions of the associated policy.
(Preset) Policy	QcloudAccessForGMELinkedRoleInGameMedia ⓘ

Authorize

Cancel

녹음 파일 저장 구성

애플리케이션 상세 페이지로 들어가 해당 페이지에서 [음성 녹음 서비스] 수정을 클릭하고 **녹음 파일 스토리지 버킷**에서 **바인딩**을 클릭합니다.

스토리지 버킷 바인딩 팝업 창에서 기존 COS 스토리지 버킷(COS 콘솔에서 사전에 생성한 버킷)을 바인딩하거나 새로운 스토리지 버킷을 생성할 수 있습니다.

Bind bucket

File storage is supported by COS. The created buckets will be synced to COS.

Current bucket

Bucket type Existing COS bucket Create

Bucket name

녹음 이벤트 콜백 설정(선택 사항)

녹음 서비스의 이벤트 콜백을 받으려면 콜백 주소를 설정합니다. 작업 경로: 애플리케이션 세부 정보 페이지 이동 후 해당 페이지의 [음성 녹음 서비스] 수정을 클릭하고, **콜백 주소**에서 수정 클릭한 후 팝업 창에 콜백을 받을 url 주소를 입력합니다. 현재 녹음 작업 완료 상태 이벤트 콜백 메시지만 푸시됩니다.

Modify callback address

Callback URL

녹음 범위 구성

녹음 범위는 사용자 지정 녹음 또는 전체 녹음 중에서 선택할 수 있습니다. 전체 녹음을 선택하고 체크 박스에 단일 스트림 또는 혼합 스트림 녹음을 지정해야 합니다.

상기 구성을 완료한 후 **저장**을 클릭하면 녹음 서비스가 시작됩니다. 더 이상 녹음 서비스를 사용할 필요가 없으면 예기치 않은 요금이 발생되지 않도록 콘솔에서 녹음 서비스 스위치를 **OFF**로 설정하십시오.

2단계: 녹음 작업 콜백 수신(선택 사항)

1단계에서 콜백 주소를 설정한 경우 녹음 작업 이벤트 콜백을 수신하게 됩니다.

3단계: 녹음 파일 조회/관리

녹음 작업이 끝난 후 몇 분 안에 오디오 파일이 생성됩니다. [COS 콘솔](#)에 로그인하여 녹음된 파일을 조회 및 관리할 수 있습니다.

사용자 지정 녹음

최종 업데이트 날짜: : 2024-01-18 16:00:04

본문은 사용자 지정 녹음 방식으로 **GME 서버 녹음** 기능을 빠르게 통합할 수 있도록 도와줍니다.

시나리오

GME는 실시간 음성 채팅 스트림을 위한 **서버 녹음** 기능을 제공하여 개발자가 콘텐츠 보관, 콘텐츠 관리 및 콘텐츠 재생산과 같은 시나리오를 구현할 수 있도록 지원합니다. 전체 녹음: 애플리케이션 내 전체 음성 녹음과 방별 혼합 스트림, 사용자별 단일 스트림을 지원합니다. 사용자 지정 녹음: 지정된 방의 방별 혼합 스트림, 사용자별 단일 스트림을 지원합니다. 녹음된 오디오 파일은 귀하의 계정에 있는 **COS** 서비스에 저장됩니다.

본문은 **사용자 지정 녹음**의 개발 및 통합 방법에 대해서만 설명합니다. 애플리케이션에 전체 녹음을 활성화하려면 [개발 가이드 - 전체 녹음](#)을 참고하십시오.

주의사항 :

GME 서버 녹음 기능을 사용하면 녹음 과정에서 GME에서 녹음 서비스 요금이 발생합니다. GME 글로벌 포털 녹음 서비스는 2023년 4월 1일부터 정식 과금될 예정이며, 자세한 내용은 [GME 구매 가이드](#)에 공지됩니다.

녹음된 파일은 Tencent Cloud 계정의 **COS** 서비스에 저장되며 저장 용량, 저장 기간 및 액세스 빈도 등 고객의 구체적인 사용 방식에 따라 **COS** 청구서가 발행됩니다. 자세한 내용은 [COS 과금 개요](#)를 참고하십시오.

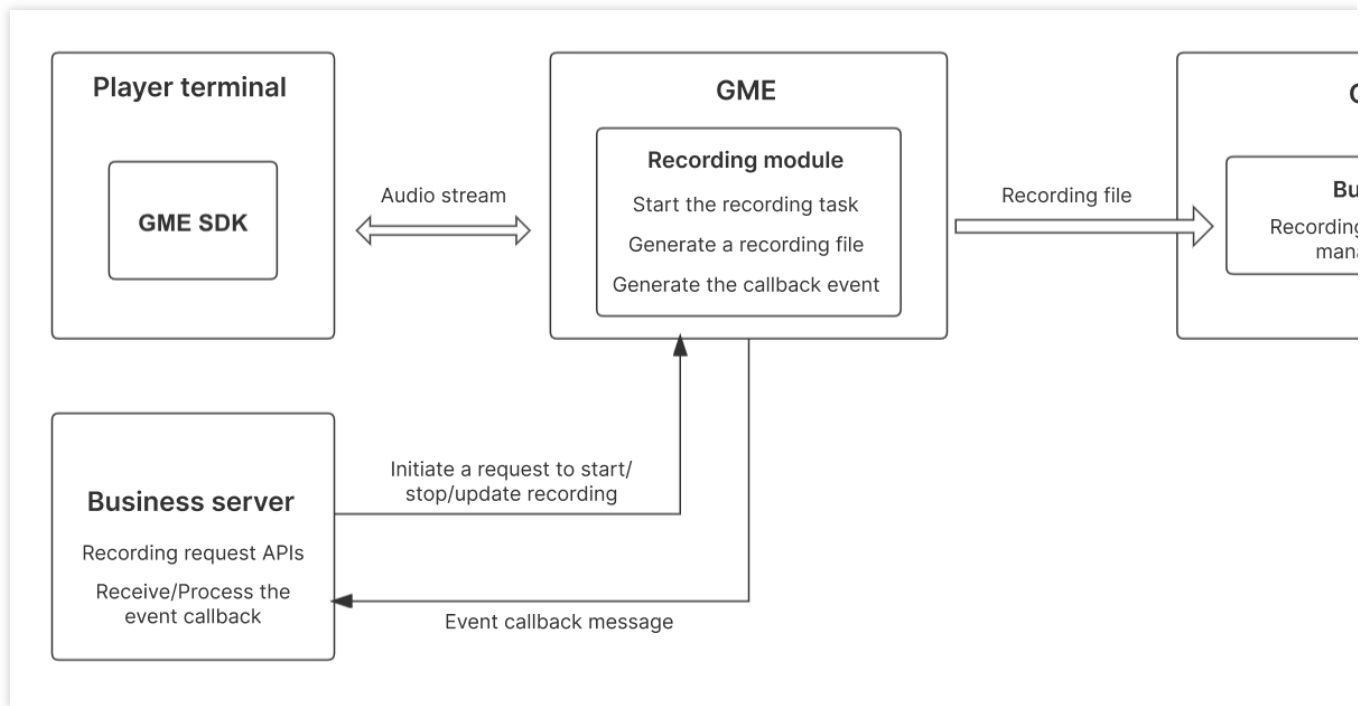
전제 조건

실시간 음성 채팅 서비스 활성화 완료: [서비스 활성화](#)를 참고하십시오.

서버 녹음 서비스 활성화 완료: 현재 서버 녹음 기능은 얼로우리스트에 있는 사용자에게 제공됩니다. 얼로우리스트를 활성화하려면 당사에 문의하십시오.

GME SDK 통합 완료: 핵심 API 및 실시간 음성 채팅 API 통합을 포함합니다. 자세한 내용은 [Native SDK 빠른 통합](#), [Unity SDK 빠른 통합](#), [Unreal SDK 빠른 통합](#)을 참고하십시오.

서비스 아키텍처



기능 설명

1. 녹음 범위

서버 API를 통해 녹음할 방 ID 혼합 스트림 또는 단일 사용자 스트림을 지정할 수 있습니다.

녹음할 지정된 방 ID에 대해 서버 API 매개변수를 통해 녹음할 플레이어를 얼로우리스트로 또는 녹음하지 않을 플레이어를 블록리스트로 지정할 수 있습니다.

2. 관련 API

StartRecord() 녹음 시작

이 API 매개변수에서 단일 스트림 또는 혼합 스트림 녹음 정의, 녹음할 RoomID 지정, 얼로우리스트 또는 블록리스트의 사용자 ID를 지정할 수 있습니다.

StopRecord() 녹음 종료

이 API를 통해 taskid에 대한 녹음을 중지할 수 있습니다. taskid를 기록하지 않은 경우 DescribeTaskInfo()를 통해 지정된 방에서 진행 중인 녹음 작업의 taskid를 가져와야 합니다.

ModifyRecordInfo() 녹음 정보 업데이트

이 API를 통해 taskid의 녹음 유형 업데이트, 얼로우리스트 사용자 ID 또는 블록리스트 사용자 ID 업데이트 등 녹음 정보를 업데이트할 수 있습니다. taskid를 기록하지 않은 경우 DescribeTaskInfo()를 통해 지정된 방에서 진행 중인 녹음 작업의 taskid를 가져와야 합니다.

DescribeTaskInfo() 방 녹음 정보 쿼리

이 API를 통해 지정된 방의 진행 중인 녹음 작업의 taskid, 얼로우리스트/블록리스트 사용자의 ID 등 녹음 정보를 쿼리할 수 있습니다.

[DescribeRecordInfo\(\)](#) 녹음 작업 정보 쿼리

이 API를 통해 taskid의 녹음 유형, 녹음 방 ID, 녹음 사용자 ID, 녹음 파일 정보 등 작업 정보를 쿼리할 수 있습니다.

3. 녹음 메커니즘

녹음 작업 실행 메커니즘

[StartRecord\(\)](#) API를 호출하면 지정된 방 녹음 작업이 시작됩니다.

녹음 작업 종료 메커니즘

[StopRecord\(\)](#) API를 호출하면 지정된 방 녹음 작업이 종료됩니다.

녹음 파일 생성 타이밍

방 혼합 스트림 녹음 파일: 방 녹음 작업이 실행된 후 이미 방에 사용자가 있는 경우 혼합 스트림 오디오 파일이 즉시 생성되기 시작합니다. 방에 사용자가 없는 경우 첫 번째 사용자가 방에 들어온 직후 혼합 스트림 오디오 파일이 생성됩니다.

사용자 단일 스트림 녹음 파일: 방 녹음 작업이 시작된 후 녹음 범위 내의 사용자가 방에 들어오면 사용자의 단일 스트림 오디오 파일이 즉시 생성되기 시작합니다.

녹음 작업 분할 메커니즘

단일 오디오 파일 길이가 2시간이 되면 자동으로 오디오 파일을 분할합니다

사용자가 마이크를 끄면 사용자의 단일 스트림 녹음이 자동으로 분할되고 사용자가 마이크를 켜면 새로운 세그먼트가 시작됩니다

녹음 작업이 비정상적으로 중단된 경우 작업이 자동으로 재연결된 후 새로운 파일이 시작됩니다

녹음 작업 이벤트 알림 메커니즘

녹음 작업 이벤트는 콜백 메커니즘을 통해 설정한 콜백 주소로 알림됩니다. **녹음 시작**, **녹음 중지**, **녹음 파일 업로드 완료** 이벤트가 발생하면 콜백 알림을 받게 됩니다.

콜백 정보에 대한 자세한 내용은 [녹음 콜백](#)을 참고하십시오

4. 저장 위치

GME 서버 녹음 파일은 귀하의 계정에 있는 **COS** 서비스의 지정된 버킷에 저장됩니다. 버킷의 리전은 귀하가 지정합니다. 선택 가능한 리전 목록은 [리전 및 액세스 도메인](#)을 참고하십시오.

5. 녹음 파일 형식

.mp3

6. 녹음 파일 이름 생성 규칙

사용자 단일 스트림 녹음 파일: bizid_roomid_userid/{작업 시작 시간}_{id}_audio.mp3

방 혼합 스트림 녹음 파일: bizid_roomid/{taskid}/{작업 시작 시간}_{id}_audio.mp3

bizid: GME 애플리케이션 ID입니다. [GME 콘솔](#)에서 가져올 수 있습니다.

roomid: 음성 방 ID입니다. 실시간 음성 채팅 서비스 이용 시 정의되어 GME SDK로 전달됩니다.

userid: 플레이어 ID입니다. 실시간 음성 채팅 서비스 이용 시 정의되어 GME SDK로 전달됩니다.

taskid: GME 녹음 서비스에서 생성한 녹음 작업 ID입니다. 각 녹음 작업에는 고유한 작업 ID가 있습니다.

id: 녹음 작업을 위한 세그먼트의 일련 번호입니다. 초기 일련 번호는 0입니다.

통합 단계

1. 콘솔에서 녹음 서비스 구성 완료비즈니스
2. 서버 API를 호출하여 녹음 범위 정의비즈니스
3. 녹음 작업 콜백 수신(선택 사항)비즈니스
4. 녹음 파일 조회/관리비즈니스

1단계: 콘솔에서 녹음 서비스 구성 완료

[GME 콘솔](#)에 로그인하여 [서비스 관리] 메뉴로 들어가 녹음 서비스를 활성화할 애플리케이션의 [설정]을 클릭하여 애플리케이션 상세 페이지로 이동합니다.

녹음 서비스 활성화/비활성화

Voice Recording ServiceRecording enable/disable Enable DisableStore recording to [Bind](#)Callback URL [Modify](#)Recording mode Custom recording Full recording I have read and agree to [Billing Description for Voice Recording Service](#).[Save](#)[Cancel](#)

페이지의 [음성 녹음 서비스]에서 **수정**을 클릭하고 녹음 스위치를 **ON**으로 설정합니다.

녹음 서비스를 처음 시작할 때 GME는 귀하의 **COS 서비스**에 액세스하기 위해 귀하에게 서비스 승인을 신청합니다.

팝업 창 페이지에서 권한 부여에 동의해주셔야 정상적으로 서버 녹음 서비스를 이용하실 수 있습니다.

Service Authorization

Other cloud service features will be used when performing operations related to this service.

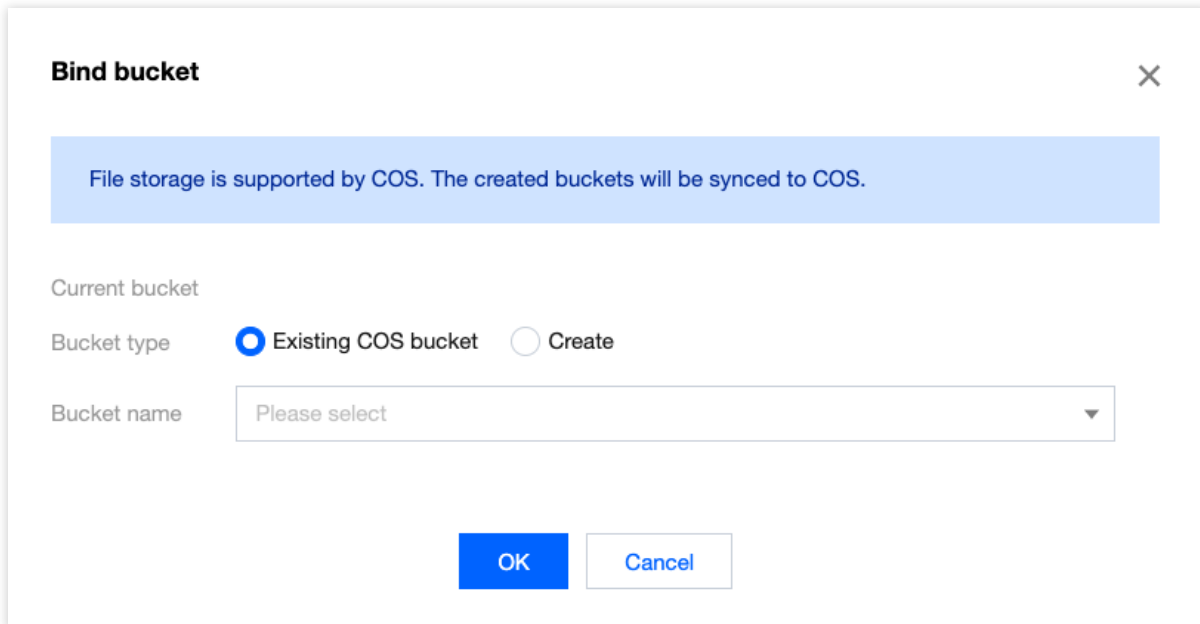
Please create Service-Linked roles for **GME** and authorize the it to use other cloud services. The relevant information is as

Role Name	GME_QCSLinkedRoleInGameMedia (Service-Linked roles)
Role Description	The current role is the GME service linked role, which will access your other service resources with scope of the permissions of the associated policy.
(Preset) Policy	QcloudAccessForGMELinkedRoleInGameMedia ⓘ

[Authorize](#)[Cancel](#)**녹음 파일 저장 구성**

녹음 파일 버킷에서 바인딩을 클릭합니다.

스토리지 버킷 바인딩 팝업 창에서 기존 COS 스토리지 버킷(COS 콘솔에서 사전에 생성한 버킷)을 바인딩하거나 새로운 스토리지 버킷을 생성할 수 있습니다.



Bind bucket ✕

File storage is supported by COS. The created buckets will be synced to COS.

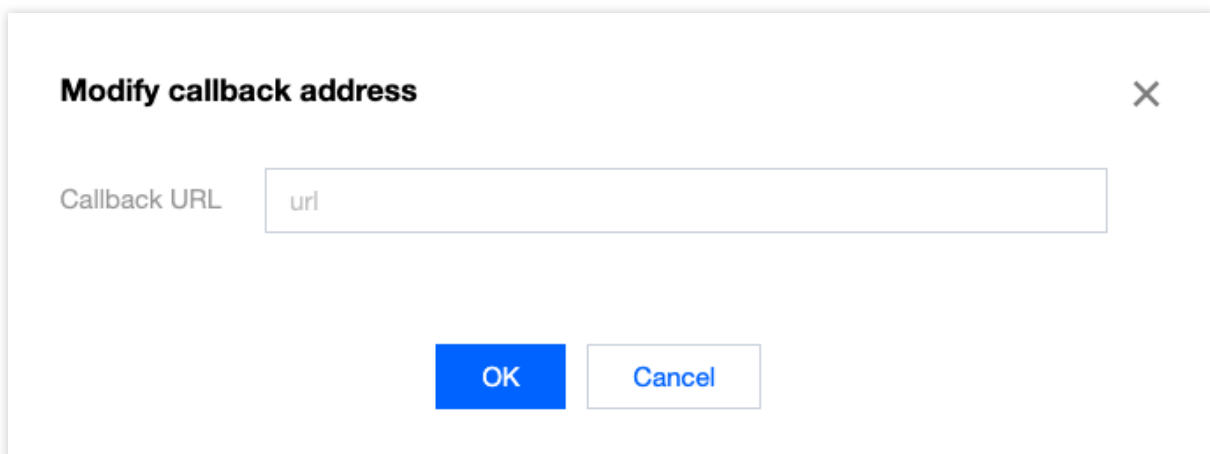
Current bucket

Bucket type Existing COS bucket Create

Bucket name

녹음 이벤트 콜백 설정(선택 사항)

녹음 서비스의 이벤트 콜백을 받으려면 콜백 주소를 설정합니다. 작업 경로: 애플리케이션 세부 정보 페이지 이동 후 해당 페이지의 [음성 녹음 서비스] 수정을 클릭하고, 콜백 주소에서 수정 클릭한 후 팝업 창에 콜백을 받을 url 주소를 입력합니다. 현재 녹음 작업 완료 상태 이벤트 콜백 메시지만 푸시됩니다.



Modify callback address ✕

Callback URL

녹음 범위 구성

녹음 범위는 사용자 지정 녹음 또는 전체 녹음 중에서 선택할 수 있습니다. 사용자 지정 녹음을 선택해야 합니다. 그렇지 않으면 서버에서 관련 API를 호출할 수 없습니다.

상기 구성을 완료한 후 **저장**을 클릭하면 녹음 서비스가 시작됩니다. 더 이상 녹음 서비스를 사용할 필요가 없으면 예치 않은 요금이 발생되지 않도록 콘솔에서 녹음 서비스 스위치를 **OFF**로 설정하십시오.

2단계: 녹음 작업 콜백 수신(선택 사항)

1단계에서 콜백 주소를 설정한 경우 녹음 작업 이벤트 콜백을 받을 수 있습니다.

3단계: 서버 API 호출 및 녹음 범위 정의

비즈니스 시나리오의 실제 요구 사항에 따라 관련 API를 호출합니다. 호출 순서 및 프로세스 설계 시 다음 사항에 주의하십시오.

(1) 이미 존재하는 RoomId에만 녹화 시작 요청을 전송할 수 있습니다. RoomId가 없으면 녹음 작업이 생성되지 않습니다.

(2) 직접 녹음 중지를 호출하지 않는 한 방에 사용자가 있으면 녹음 프로세스가 지속됩니다. 방에 있는 모든 사용자가 퇴장한 경우 기존 녹음 작업 프로세스는 12시간 동안 지속됩니다. 이 시간 동안 사용자가 방에 다시 입장하면 녹음이 자동으로 시작됩니다. 녹음 프로세스의 지속 시간이 끝난 후에 사용자가 방에 입장하면 녹음이 자동으로 시작되지 않습니다.

4단계: 녹음 파일 조회/관리

녹음 작업이 끝난 후 몇 분 안에 오디오 파일이 생성됩니다. [COS 콘솔](#)에 로그인하여 녹음된 파일을 조회 및 관리할 수 있습니다.

녹음 콜백 설명

최종 업데이트 날짜: : 2024-01-18 16:00:04

서버 녹음 콜백 설명

설명

네트워크의 영향을 받아 서버가 알림을 받는 순서와 이벤트가 발생하는 순서가 정확히 일치하지 않을 수 있습니다.서비스에는 재시도 메커니즘이 있지만 모든 메시지가 도달한다는 보장이 없습니다.상기 두 가지 사항을 고려할 때 비즈니스의 핵심 비즈니스 로직이 메시지 알림 서비스에 의존하는 것은 권장되지 않습니다.

네트워크 프로토콜

콘솔에서 콜백 주소, 즉 HTTP(S) 프로토콜 API의 URL을 설정한 경우 POST 요청을 지원해야 하며 전송 데이터 인코딩은 UTF-8을 채택합니다.

HTTP 헤더 매개변수

이름	유형	필수 여부	설명
Signature	string	Yes	서명, 자세한 내용은 아래의 서명 생성 을 참고하십시오

서명 생성

Signature = HMAC-SH1 (strContent, SecretKey)

strContent: body의 전체 JSON 콘텐츠(길이는 Content-Length 기준)인 서명 문자열입니다.

body: 비즈니스로 콜백되는 JSON 콘텐츠로, 아래 [콜백 예시](#)의 전체 콘텐츠가 body입니다.

SecretKey: 키. 애플리케이션의 권한 키로, [콘솔 > 애플리케이션 상세 정보](#)를 통해 확인할 수 있습니다.

HMAC-SH1: 서명 알고리즘입니다.

콜백 매개변수

이름	유형	설명
BizID	Integer	애플리케이션의 AppID로, 콘솔 > 애플리케이션 세부 정보 를 통해 확인할 수 있습니다.
RoomID	String	방 ID
UserID	String	사용자 ID
RecordMode	Integer	녹음 모드 0: 단일 스트림

		1: 혼합 스트림
Timestamp	Integer	콜백 전송 시의 타임스탬프(s)
TaskID	Integer	클라우드 녹음 서비스에서 할당한 작업 ID입니다. 작업 ID는 녹음의 라이프사이클 프로세스에 대한 고유 식별자이며 녹음이 종료되면 의미를 잃습니다. 사용자 지정 녹음 모드를 사용하면, 작업 ID는 녹음이 시작될 때 응답 매개변수를 통해 얻을 수 있으며, 비즈니스에서 저장해 두어야 합니다. 이는 이번 녹음 작업에 대한 다음 번 작업의 요청 매개변수입니다.
EventType	Integer	이벤트 유형
Detail	EventDetail	이벤트 세부 정보, 형식은 EventType에 의해 결정됩니다

EventDetail 이벤트 세부 정보 설명

EventType	설명	Detail
1	오디오 파일 녹음 시작	SeqNo: Number, 세그먼트 시리얼 넘버 FileName: String, 파일 이름
2	오디오 파일 녹음 완료	SeqNo: Number, 세그먼트 시리얼 넘버 FileName: String, 파일 이름
3	오디오 파일 업로드 완료	SeqNo: Number, 세그먼트 시리얼 넘버 FileName: String, 파일 이름

콜백 예시



```
{
  "BizID":1400000000,
  "RoomID":"100",
  "UserID":"999",
  "TaskID":446946705284000000,
  "RecordMode":1,
  "Timestamp":1675930605,
  "EventType":1,
  "Detail":{
    "SeqNo":0,
    "FileName":"1400000000_100_999/2023-02-09-16-16-45_446946705284000000_audio
```

```
}  
}
```

클라이언트 녹음

실시간 음성 채팅 녹음

최종 업데이트 날짜: : 2023-03-13 17:12:23

본문은 개발자가 쉽게 통합하고 디버깅할 수 있도록 실시간 음성 채팅 클라이언트 녹음용 Game Multimedia Engine(GME) API에 대해 설명합니다.

전제 조건

- 음성 채팅 서비스 활성화 완료: [서비스 활성화](#)를 참고하십시오.
- GME SDK 통합 완료: 핵심 API 및 음성 채팅 API 연동을 포함합니다. 자세한 내용은 [Native SDK 빠른 통합](#), [Unity SDK 빠른 통합](#), [Unreal SDK 빠른 통합](#)을 참고하십시오.
- 실시간 음성 채팅 서비스를 정상적으로 사용할 수 있는 상태여야 합니다.

순서도

녹화 API

녹화 시작

이 API를 통해 방 내 사운드 녹음을 시작하려면 입장에 성공한 후에만 API를 호출할 수 있습니다. 다음 세 가지 유형의 사운드를 개별 녹음하거나 동시에 녹음할 수 있습니다:

- 마이크 사운드: 마이크에서 수집된 사운드를 녹음합니다. 먼저 `EnableMic`로 마이크를 켜야 합니다.
- 다른 사람의 음성: GME를 통해 수신된 음성 채팅 방 내 다른 사람들이 말하는 혼합된 음성을 녹음합니다. 먼저 `EnableSpeaker`로 스피커를 켜야 합니다.
- 반주 사운드: GME를 통해 재생되는 로컬 반주 사운드를 녹음합니다. 먼저 `StartAccompany`로 반주를 재생해야 합니다.

녹음 파일 형식은 MP3 오디오 파일입니다.

API 프로토타입

【Unity】

```
public abstract int StartRecord(string filePath, int sampleRate, int channels, bool recordLocalMic, bool recordRemote, bool recordAccompany);
```

매개변수	유형	설명
filePath	String	녹음 파일 저장 경로입니다.
sampleRate	int	원본 오디오 데이터의 오디오 샘플링 레이트, 48000 입력을 권장합니다.
channels	int	원본 오디오 데이터의 오디오 채널 수, 2 입력을 권장합니다.
recordLocalMic	bool	로컬 마이크에서 수집된 사운드의 녹음 여부입니다.
recordRemote	bool	수신된 방에 있는 다른 사람들의 음성의 녹음 여부입니다.
recordAccompany	bool	GME를 통해 로컬로 재생되는 반주 사운드 녹음 여부입니다.

예시 코드

【Unity】

```
string recordPath = Application.persistentDataPath + string.Format("/{0}.mp3", filename);
ITMGContext.GetInstance().GetAudioCtrl().StartRecord(recordPath, 48000, 2, true, false, false);
```

반환된 값

반환 값 0은 성공을 의미하고 0이 아닌 값은 실패를 의미합니다.

녹음 중지

녹음을 중지하려면 이 API를 호출합니다.

API 프로토타입

【Unity】

```
public abstract int StopRecord();
```

예시 코드

【Unity】

```
ITMGContext.GetInstance().GetAudioCtrl().StopRecord();
```

녹음 일시 정지

녹음을 일시 정지하려면 이 API를 호출합니다.

API 프로토타입

【Unity】

```
public abstract int PauseRecord();
```

예시 코드

【Unity】

```
ITMGContext.GetInstance().GetAudioCtrl().PauseRecord();
```

녹음 재개

녹음을 재개하려면 이 API를 호출합니다.

API 프로토타입

【Unity】

```
public abstract int ResumeRecord();
```

예시 코드

【Unity】

```
ITMGContext.GetInstance().GetAudioCtrl().ResumeRecord();
```

마이크 사운드 녹음 스위치

녹음이 시작된 후, 이 API를 사용하여 마이크가 수집한 사운드에 대한 녹음을 활성화/비활성화 합니다.

API 프로토타입

【Unity】

```
public abstract int EnableRecordLocalMic (bool recordLocalMic);
```

매개변수	유형	설명
EnableRecordLocalMic	bool	true는 녹음 활성화를 의미하고 false는 녹음 비활성화를 의미합니다.

다른 사람의 사운드 녹음 스위치

녹음이 시작된 후, 이 API를 통해 다른 사람의 사운드에 대한 녹음을 활성화/비활성화 합니다.

API 프로토타입

【Unity】

```
public abstract int EnableRecordRemote (bool recordRemote);
```

매개변수	유형	설명
EnableRecordRemote	bool	true는 녹음 활성화를 의미하고 false는 녹음 비활성화를 의미합니다.

반주 사운드 녹음 스위치

녹음이 시작된 후, 이 API를 통해 반주 사운드에 대한 녹음을 활성화/비활성화 합니다.

API 프로토타입

【Unity】

```
EnableRecordAccompany (bool recordAccompany);
```

매개변수	유형	설명
recordAccompany	bool	true는 녹음 활성화를 의미하고 false는 녹음 비활성화를 의미합니다.

레인지 보이스

최종 업데이트 날짜: : 2024-01-18 16:00:04

기능 소개

음성 채팅방에서는 사용자가 일정 거리 내에서 다른 사용자와 실시간 음성 통화를 할 수 있습니다. 이 기능은 비즈니스 레이아웃을 통해 GME 클라이언트 API를 호출하여 음원 방향을 업데이트하고 로컬의 위치를 서버에 알립니다. 서버는 로컬 세계 좌표+로컬이 수신한 오디오 범위와 기타 엔드의 세계 좌표 + 기타 엔드가 수신한 오디오 범위를 판단한 후 서버는 플레이어 범위 내의 오디오 스트림을 전달합니다. 기본적으로 플레이어에 가장 가까운 20개의 오디오 스트림을 전달하며 최대 만 명 이상의 사람들이 방에서 동시에 마이크를 켤 수 있습니다.

사용 사례

배틀 로얄 게임	<p>배틀 로얄 게임 및 서바이벌 슈팅 모바일 게임에서 '팀원만' 또는 '모든 사람' 음성 모드를 제공합니다. 게임 설정에서 플레이어 선택에 따라:</p> <ol style="list-style-type: none"> '팀원만' 모드를 사용하면 같은 팀의 팀원이 말하는 음성만 들립니다. '모든 사람' 모드를 사용하면 팀원의 목소리와 일정 범위 내의 다른 플레이어의 음성을 들을 수 있습니다.
몰입형 버추얼 시나리오	<p>예를 들어 게임 콘서트와 같은 가상의 시나리오에서 가수는 가상의 방에서 레인지 보이스를 통해 노래를 부를 수 있고, 객석에 있는 모든 시청자들은 노래를 듣고 일정 범위 내의 다른 관객과 소통할 수 있습니다. 같은 방에 있는 수만 명의 사람들이 동시에 마이크를 활성화할 수 있습니다.</p>

데모 효과

시나리오 기반 Demo를 다운로드하여 3D 음향 효과와 다양한 음성을 체험할 수 있습니다.

기본 개념

레인지 보이스 기능에는 음성 모드, 음성 수신 범위 및 TeamID의 세 가지 개념이 포함됩니다.

음성 모드

사용자가 레인지 음성 채팅방에 방에 들어갈 때 사용자는 두 가지 음성 모드 중에서 선택할 수 있습니다.

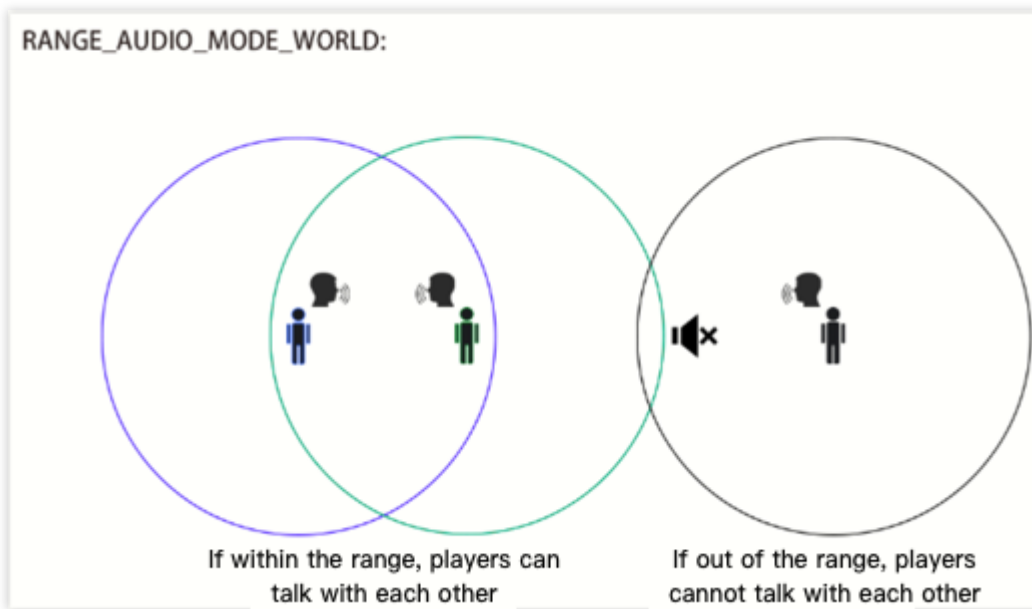
음성 모드	매개변수	기능

모든 사람	RANGE_AUDIO_MODE_WORLD	이 모드에서 로컬 플레이어는 특정 범위 내에 있는 다른 플레이어의 소리를 들을 수 있고, 그들도 이 모드에 있는 경우 서로 대화할 수 있습니다. 팀원들은 서로의 말을 들을 수 있습니다
팀원만	RANGE_AUDIO_MODE_TEAM	팀원만 서로의 말을 들을 수 있습니다

주의사항 :

팀원은 거리와 음성 모드에 관계없이 서로 대화할 수 있습니다.

음성 수신 범위



음성 모드가 **모든사람(RANGE_AUDIO_MODE_WORLD)**으로 설정된 경우 수신 범위는 UpdateAudioRecvRange API에 따라 달라집니다.

서로 다른 팀의 플레이어 A와 B가 음성 모드를**모든 사람(RANGE_AUDIO_MODE_WORLD)**으로 설정했다고 가정합니다.

플레이어 좌표	음성 수신 범위	다른 플레이어의 음성 도달 가능성	팀원의 음성 도달 가능성
A(0,0,0)	10미터	플레이어 B는 플레이어로부터 10미터 이내에 있으므로 플레이어 B의 소리를 들을 수 있습니다	팀원끼리 대화가 가능합니다
B(0,8,0)	5미터	플레이어 A가 5미터 이상 떨어져 있기 때문에 플레이어 B는 플레이어 A의 소리를 들을 수 없습니다	팀원끼리 대화가 가능합니다

설명 :

구체적인 플레이어의 음성 도달 가능성에 대한 자세한 내용은 [레인지 보이스](#)를 참고하십시오.

TeamID

레인지 보이스 기능을 사용하려면 SetRangeAudioTeamID API를 호출하여 팀 ID(TeamID)를 설정한 후 EnterRoom API를 호출하여 방에 입장해야 합니다.

음성 채팅방 입장 시 지정한 TeamID != 0인 경우 레인지 음성 채팅방 모드로 진입하게 됩니다. 사용자가 음성 채팅방 입장 시 TeamID를 1로 설정하고 음성 모드를 RANGE_AUDIO_MODE_TEAM으로 설정하면 TeamID = 1인 구성원만 사용자의 음성을 들을 수 있습니다. 사용자가 음성 모드를 RANGE_AUDIO_MODE_WORLD로 설정하면 TeamID = 1인 구성원뿐만 아니라 일정 범위 내의 플레이어도 사용자의 음성을 들을 수 있습니다.

TeamID	음성 모드	범위	음성 도달 가능성
TeamID != 0, TeamID = 1이라고 가정합니다	RANGE_AUDIO_MODE_TEAM	10미터	사용자는 TeamID = 1인 구성원과만 대화할 수 있습니다
	RANGE_AUDIO_MODE_WORLD	10미터	사용자는 TeamID = 1인 구성원 및 RANGE_AUDIO_MODE_WORLD 음성 모드의 구성원과 10m 이내의 같은 방에서 대화할 수 있습니다

사용자가 TeamID = 0으로 설정된 음성 채팅방에 입장하면 방에 있는 모든 구성원(음성 모드에 관계없이)이 사용자의 음성을 들을 수 있는 레인지 보이스 호스트 모드로 들어갑니다.

TeamID 상황	TeamID 수정 시간	범위	음성 도달 가능성
TeamID = 0	방 입장 전에는 TeamID != 0이며, 입장 후에는 TeamID = 0으로 변경됩니다	10미터	방에 있는 모든 구성원(음성 모드에 관계없이)이 음성을 들을 수 있습니다 사용자는 TeamID = 0인 구성원과 대화할 수 있습니다 사용자는 같은 방에서 10미터 이내의 RANGE_AUDIO_MODE_WORLD 음성 모드로 구성원의 목소리를 들을 수 있습니다
	방 입장 전 TeamID = 0, 방 입장 중 TeamID = 0	10미터	방에 있는 모든 구성원(음성 모드에 관계없이)이 음성을 들을 수 있습니다 사용자는 TeamID = 0인 구성원과 대화할 수 있습니다 방에 있는 다른 구성원의 소리를 들을 수 없습니다

예시 시나리오

배틀 로얄 게임: 배틀 로얄 게임에서는 4명의 플레이어가 한 팀으로 그룹화되며 이들을 위한 TeamID가 설정되어야 합니다. 배틀 룸은 100명의 플레이어, 즉 25개 팀을 포함할 수 있으므로 모든 팀이 동일한 음성 채팅방에 입장합니다. 전투 중 플레이어가 10m 이내의 낯선 사람과 대화하고 싶다면 플레이어는 음성을 들을 수 있는 범위를 10m로, 음성 모드를 RANGE_AUDIO_MODE_WORLD로 설정하고 마이크와 스피커를 모두 활성화할 수 있습니다. 플레이어가 팀원과만 소통하고 싶다면 음성 모드를 RANGE_AUDIO_MODE_TEAM으로 설정할 수 있습니다.

게임 콘서트: 가수가 게임에서 콘서트를 하고 싶지만 플레이어와 교류할 필요가 없는 경우, 플레이어는 자신의 TeamID = OpenID로 설정하고, 음성 모드를 RANGE_AUDIO_MODE_WORLD로 설정하고, 음성을 들을 수 있는 범위를 적절한 값으로 설정하여 레인지 음성 채팅방에 입장하여 근처에 있는 플레이어와 대화할 수 있습니다. 가수가 방에 들어가기 전에 TeamID를 0으로 설정하면 방에 있는 모든 구성원이 가수의 소리를 들을 수 있지만 가수는 다른 구성원의 소리를 듣지 않습니다.

호스트 모드: 가상 보드 게임과 같은 게임에서 호스트는 방에 있는 모든 구성원이 들을 수 있어야 하고 특정 범위 내의 플레이어의 소리를 들을 수 있어야 합니다. 이 경우 호스트는 TeamID != 0 값으로 설정한 상태로 방에 입장할 수 있으며, 방 입장 후 TeamID를 0으로 변경하여 방에 있는 모든 구성원이 들을 수 있고 일정 범위 내의 플레이어는 들을 수 있습니다.

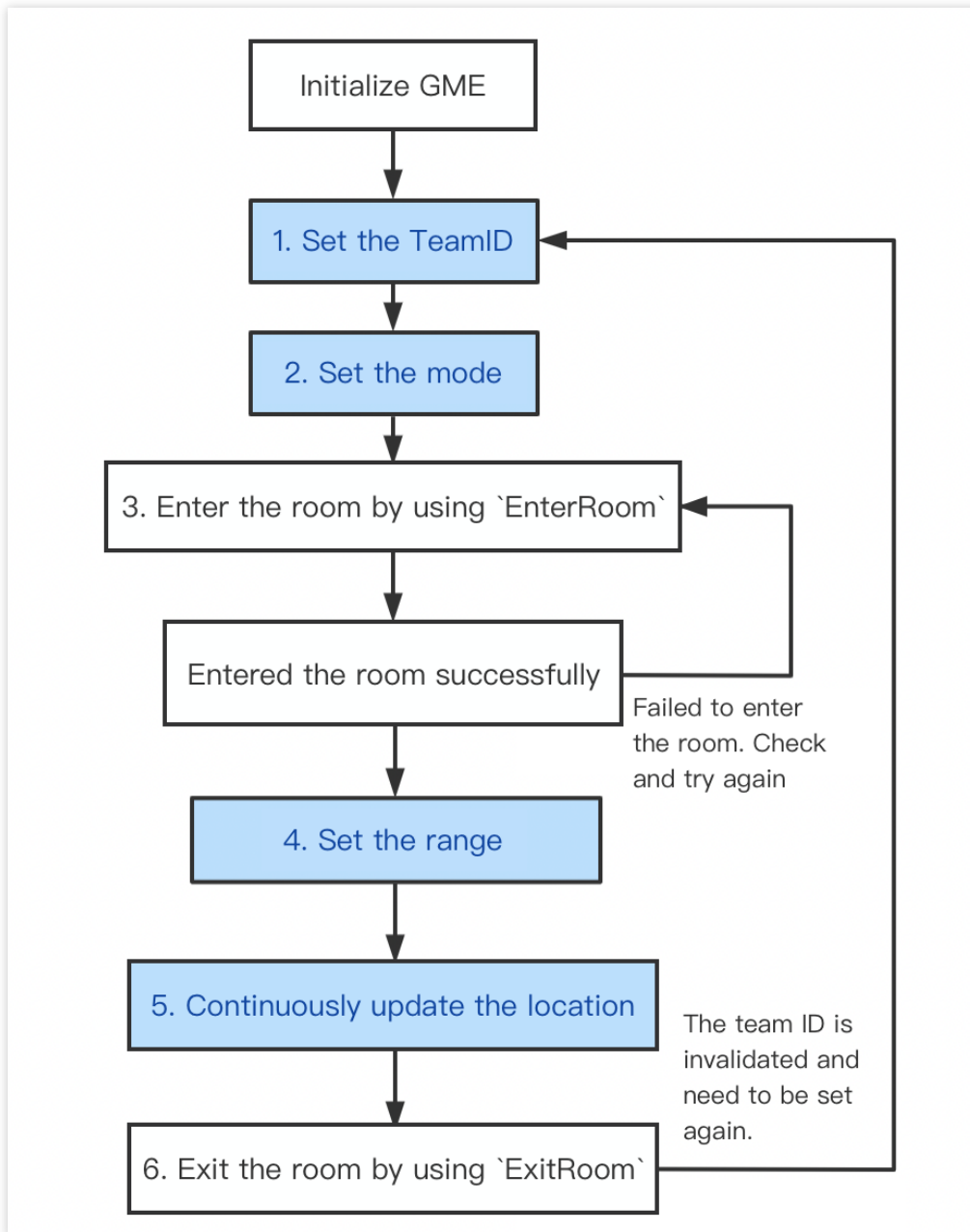
전제 조건

실시간 음성 채팅 서비스 활성화 완료: [서비스 활성화](#)를 참고하십시오.

GME SDK 통합 완료: 핵심 API 및 실시간 음성 채팅 API 통합을 포함합니다. 자세한 내용은 [Native SDK 빠른 통합](#), [Unity SDK 빠른 통합](#), [Unreal SDK 빠른 통합](#)을 참고하십시오.

만명 레인지 보이스: 같은 방에 있는 레인지 보이스의 수가 1000개를 초과하는 경우 [Submit Ticket](#)을 통해 GME 개발자에게 문의하십시오.

사용 프로세스



주의사항 :

순서도에 따라 API를 호출해야 합니다.

순서도의 파란색 부분은 레인지 보이스의 프로세스를 보여줍니다.

레인지 음성 채팅방은 팀 음성 채팅방과 달리 **입장 시 원할 음질을 선택해야 합니다.**

방에 들어가면 UpdateAudioRecvRange(적어도 한 번)를 호출하고 프레임당 한 번 UpdateSelfPosition을 호출합니다.

1. TeamID 설정

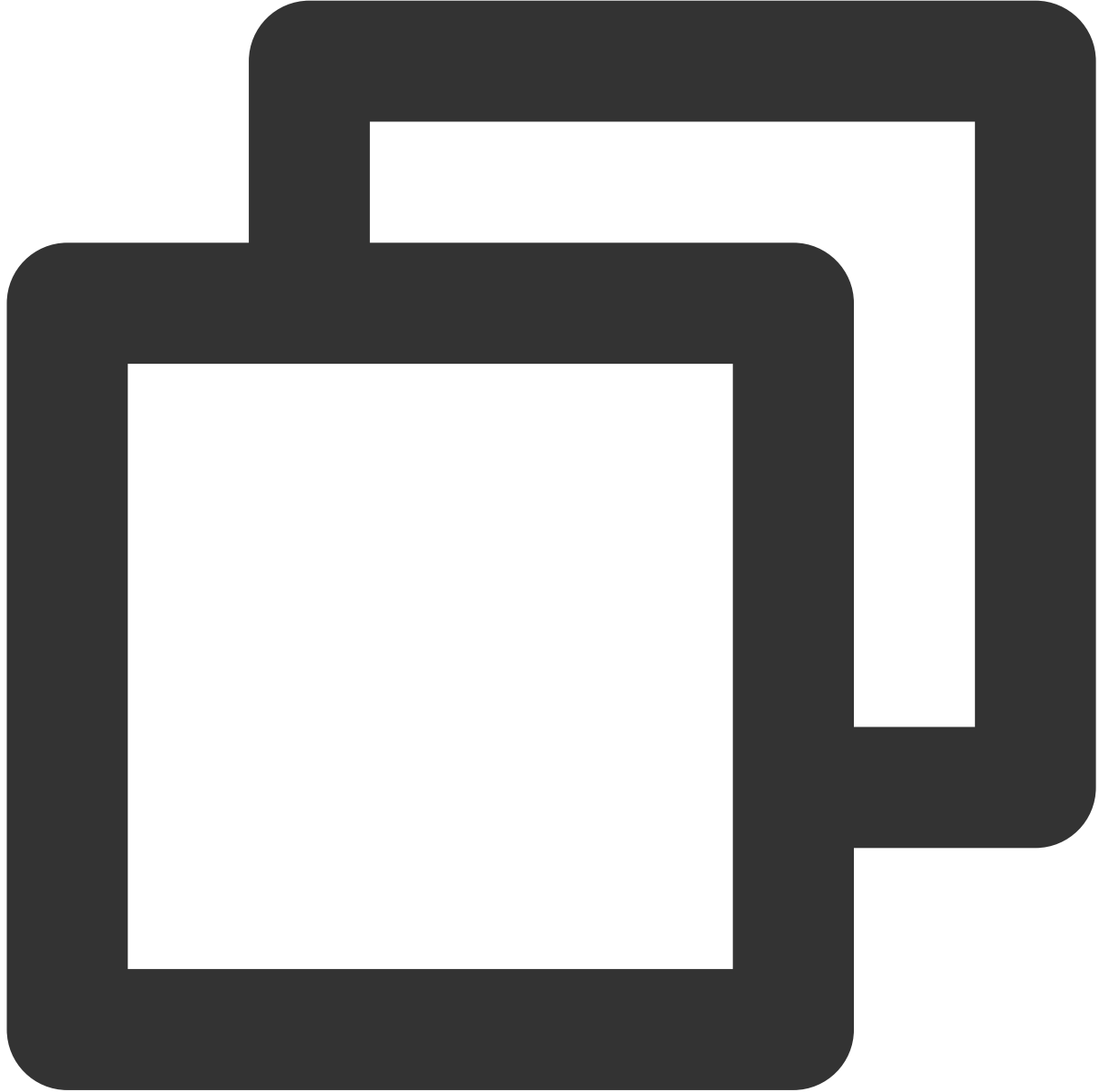
방 입장 전에 팀 ID를 설정하기 위해 이 API를 호출하면 다음 방 입장에 영향을 미칩니다.

이 API는 방 입장 후 팀 ID를 수정하기 위해 호출할 수 있습니다. 수정 사항은 즉시 적용됩니다.

방 퇴장 후에 TeamID가 자동으로 0으로 재설정되지 않습니다. 따라서 이 음성 모드를 호출하기로 결정한 후에는 EnterRoom을 호출하기 전에 TeamID를 설정하십시오.

방 퇴장 후 재입장 시, 방 퇴장 성공 콜백이 반환된 후 다시 팀 ID 설정 API를 호출합니다.

함수 프로토타입



```
ITMGContext SetRangeAudioTeamID(int teamID)
```

매개변수	유형	설명
teamID	int	레인지 보이스 모드에서만 사용되는 TeamID. 0(기본값)으로 설정하면 팀 음성 모드가 사용됩니다.

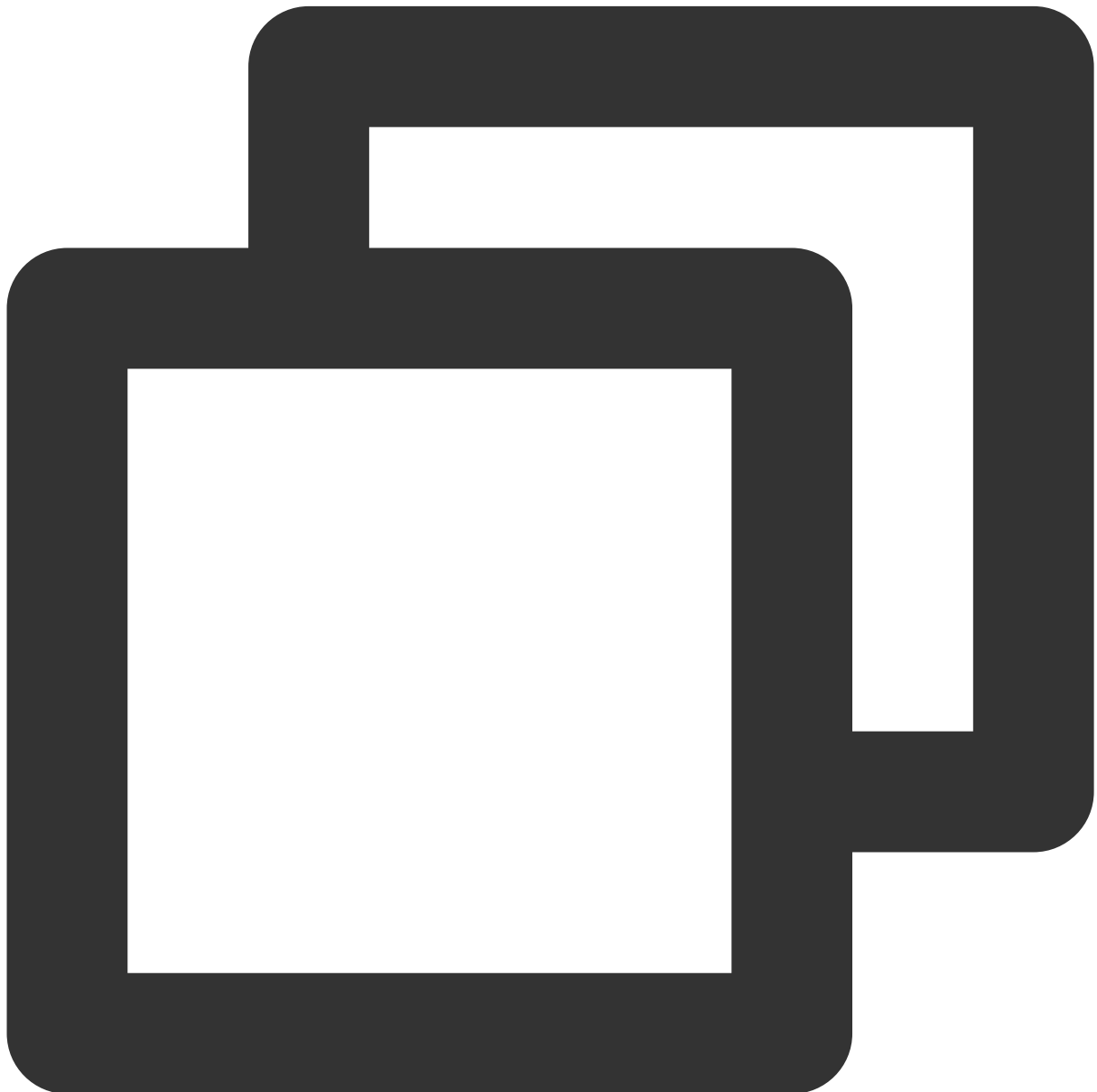
2. 음성 모드 설정

방 입장 전에 음성 모드를 변경하기 위해 이 API를 호출하면 다음 방 입장에 영향을 미칩니다.

방 입장 후 음성 모드를 변경하기 위해 이 API를 호출하면 현재 음성 모드가 바로 변경됩니다.

이 매개변수는 방 퇴장 후에 자동으로 MODE_WORLD로 재설정되지 않습니다. 따라서 이 메소드를 호출하기로 결정하였다면 EnterRoom을 호출하기 전에 수행하십시오.

함수 프로토타입



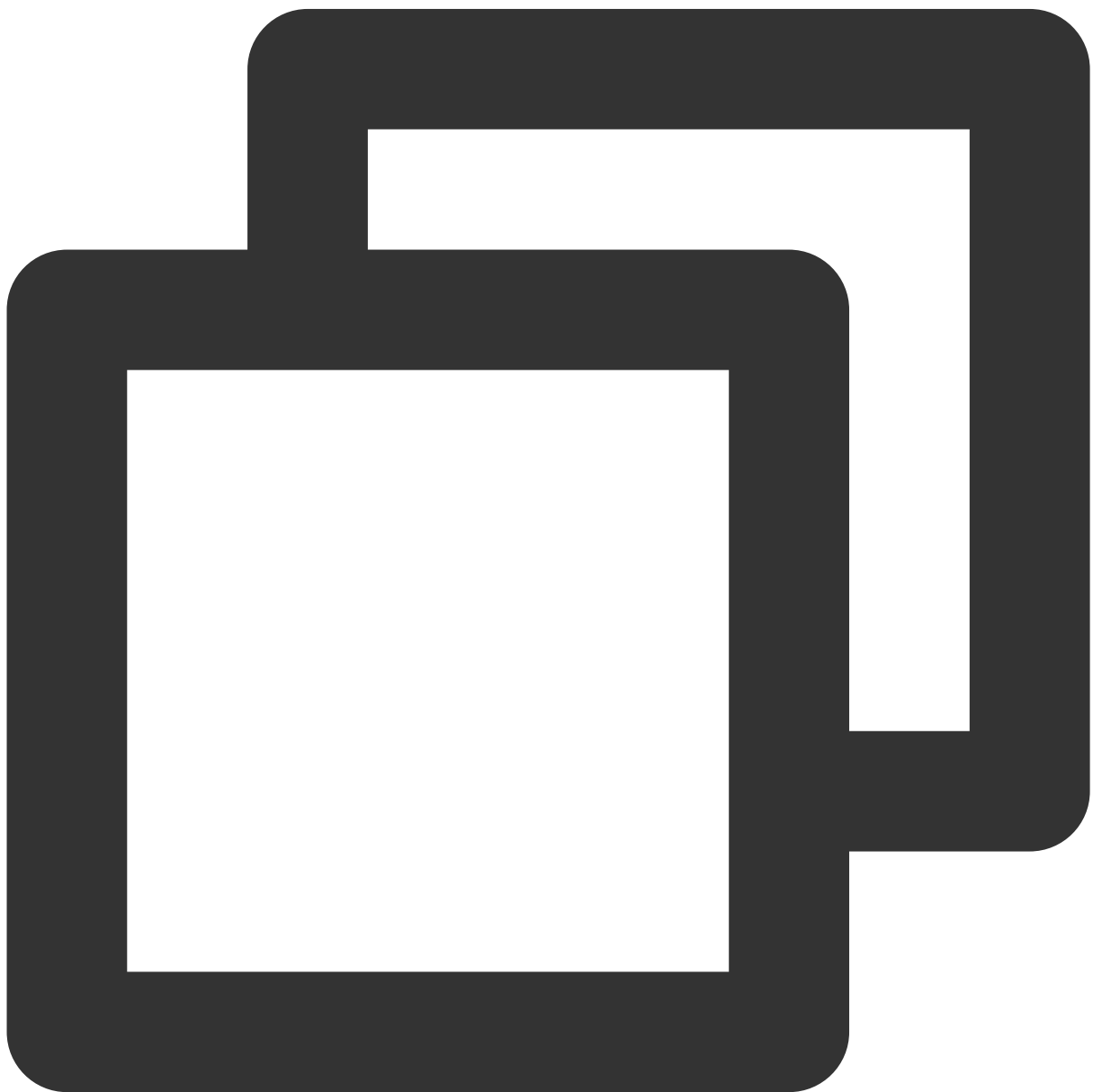
```
ITMGRoom int SetRangeAudioMode (RANGE_AUDIO_MODE rangeAudioMode)
```

매개변수	유형	설명
rangeAudioMode	int	0(MODE_WORLD)은 '모든 사람', 1(MODE_TEAM)은 '팀원만'을 의미합니다

3. 음성 채팅방 입장

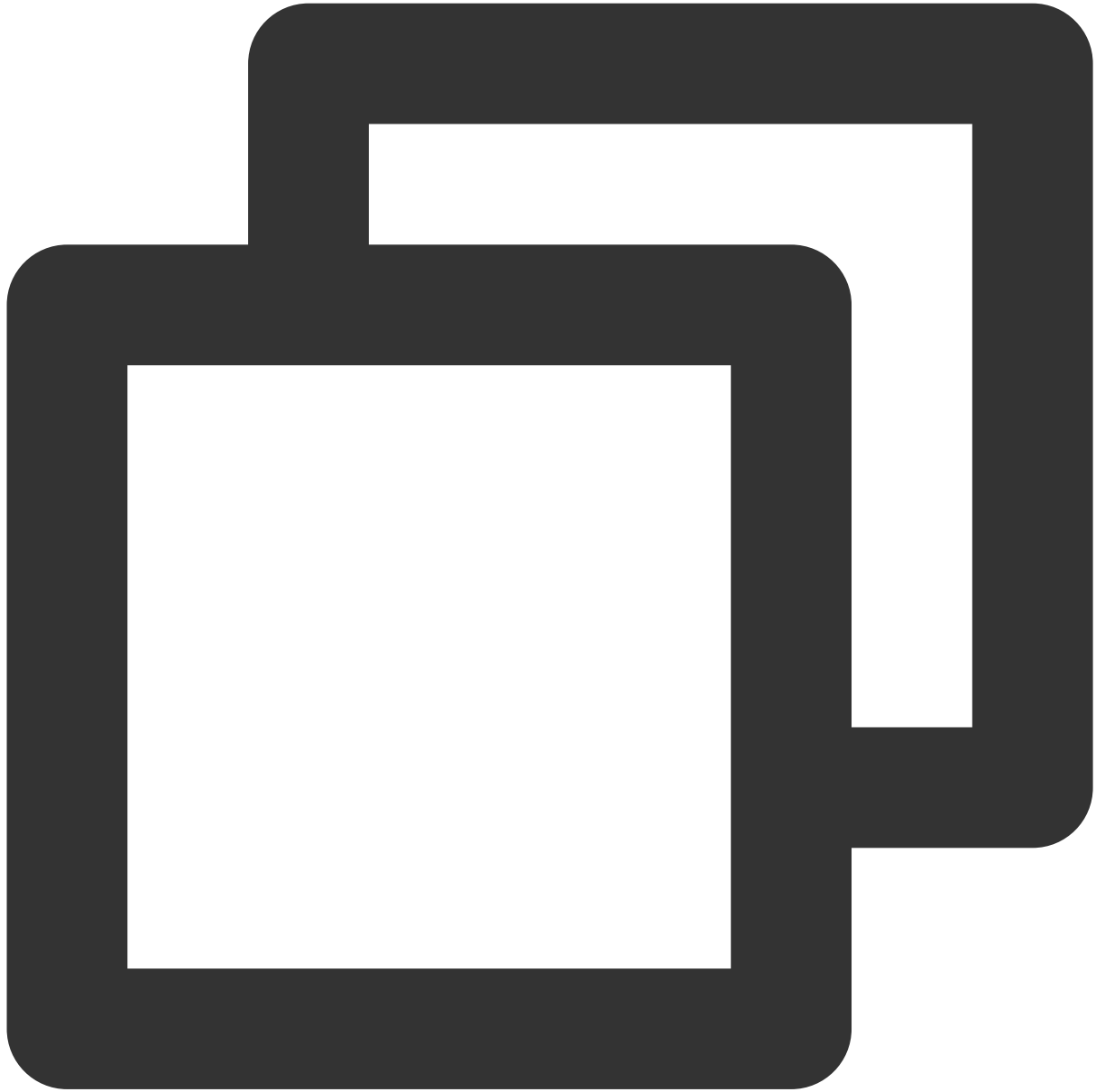
EnterRoom을 호출하여 음성 채팅방에 들어가기 전에 SetRangeAudioTeamID 및 SetRangeAudioMode라는 두 가지 API를 호출해야 합니다.

함수 프로토타입




```
ITMGContext.GetInstance(this).EnterRoom(roomId, ITMG_ROOM_TYPE_FLUENCY, authBuffer);
```

음성 채팅방 입장 시 원활 음질을 선택하고, 방 입장 콜백을 수신 및 처리합니다.



```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {  
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ENTER_ROOM == type)  
    {  
        //반환된 Data 분석  
        int nErrCode = data.getIntExtra("result" , -1);  
        String strErrMsg = data.getStringExtra("error_info");  
    }  
}
```

```
if (nErrCode == AVError.AV_OK)
{
    //방 입장에 대한 성공 응답을 받으면 작업을 진행할 수 있습니다
    ScrollView_ShowLog("EnterRoom success");
    Log.i(TAG, "EnterRoom success!");
}
else
{
    //방 입장에 실패한 경우 반환되는 에러 메시지를 분석해야 합니다
    ScrollView_ShowLog("EnterRoom fail :" + strErrMsg);
    Log.i(TAG, "EnterRoom fail!");
}
}
```

방에 들어가면 UpdateAudioRecvRange(적어도 한 번)를 호출하고 UpdateSelfPosition을 프레임당 한 번 호출합니다.

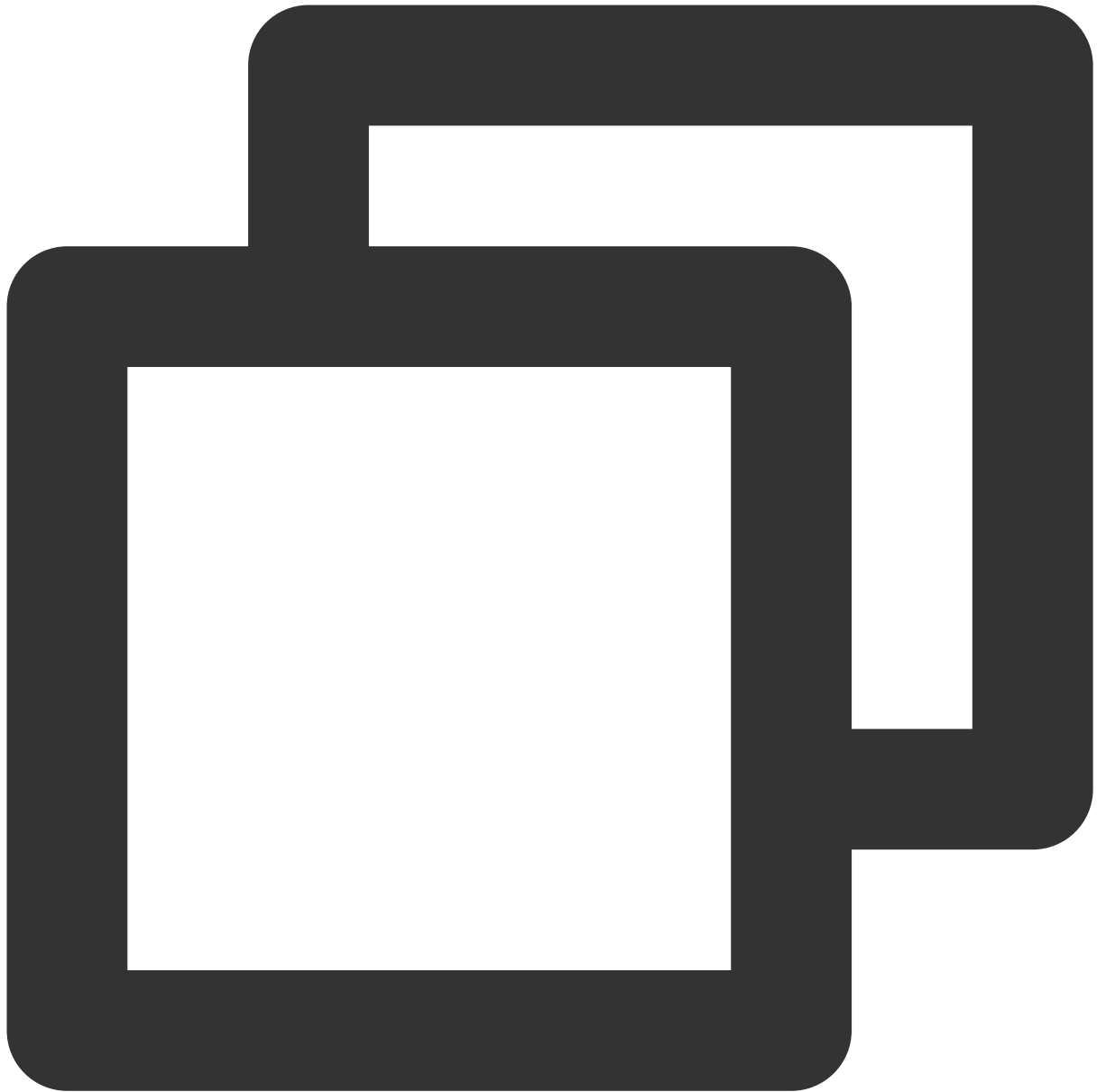
4. 음성 수신 범위 설정

이 메소드는 음성 수신 범위(게임 엔진에 따름) 설정에 사용되며 방 입장 성공 후에만 호출할 수 있습니다.

이 메소드는 UpdateSelfPosition과 함께 사용하여 음원 위치를 업데이트해야 합니다.

이 메소드는 한 번만 호출하면 적용되며 매개변수 값을 수정할 수 있습니다.

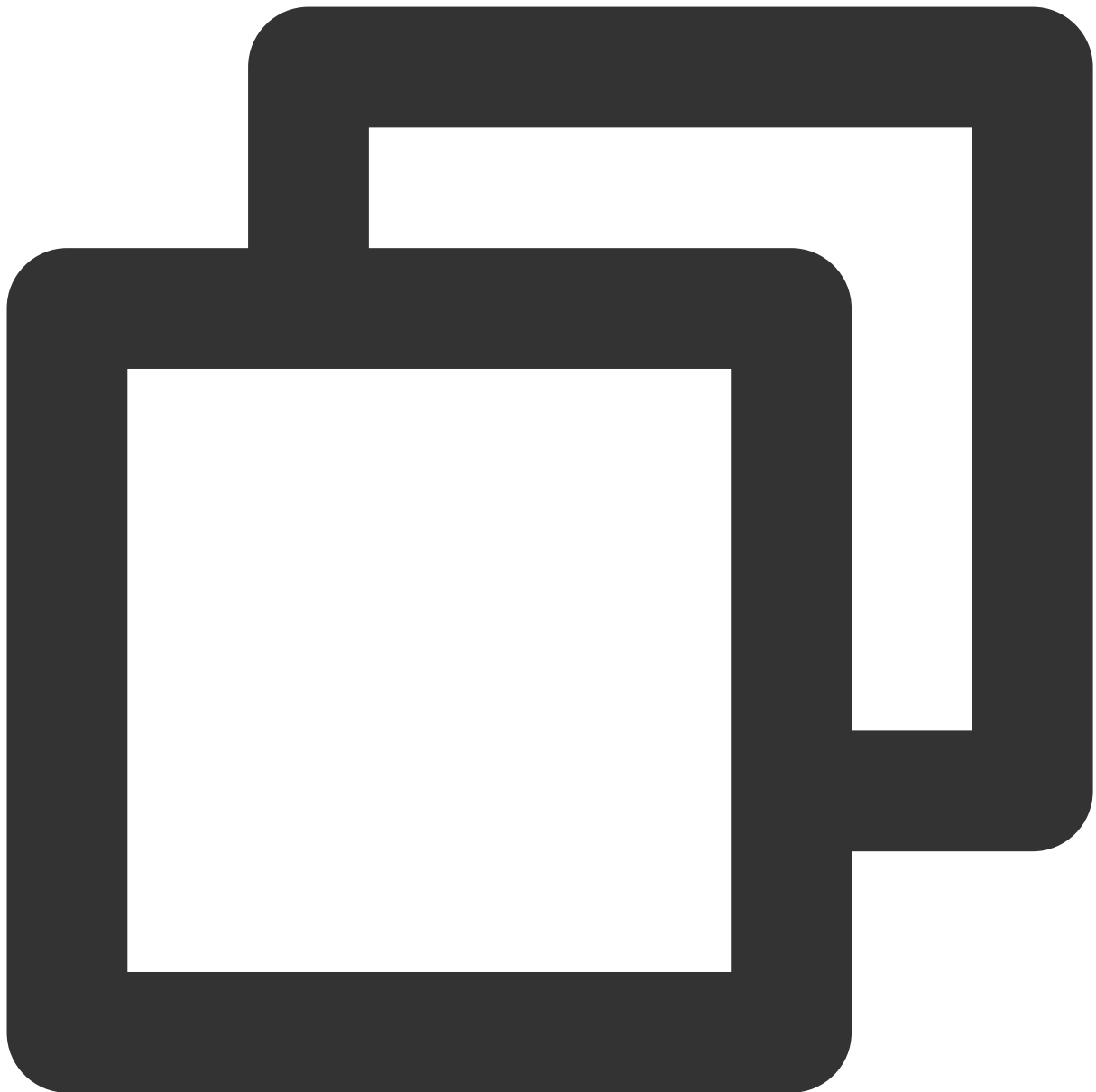
함수 프로토타입



```
ITMGRoom int UpdateAudioRecvRange(int range)
```

매개변수	유형	설명
range	int	게임 엔진에서 사용하는 거리 단위의 최대 음성 수신 범위

예시 코드



```
ITMGContext.GetInstance().GetRoom().UpdateAudioRecvRange(300);
```

5. 음원 위치 업데이트

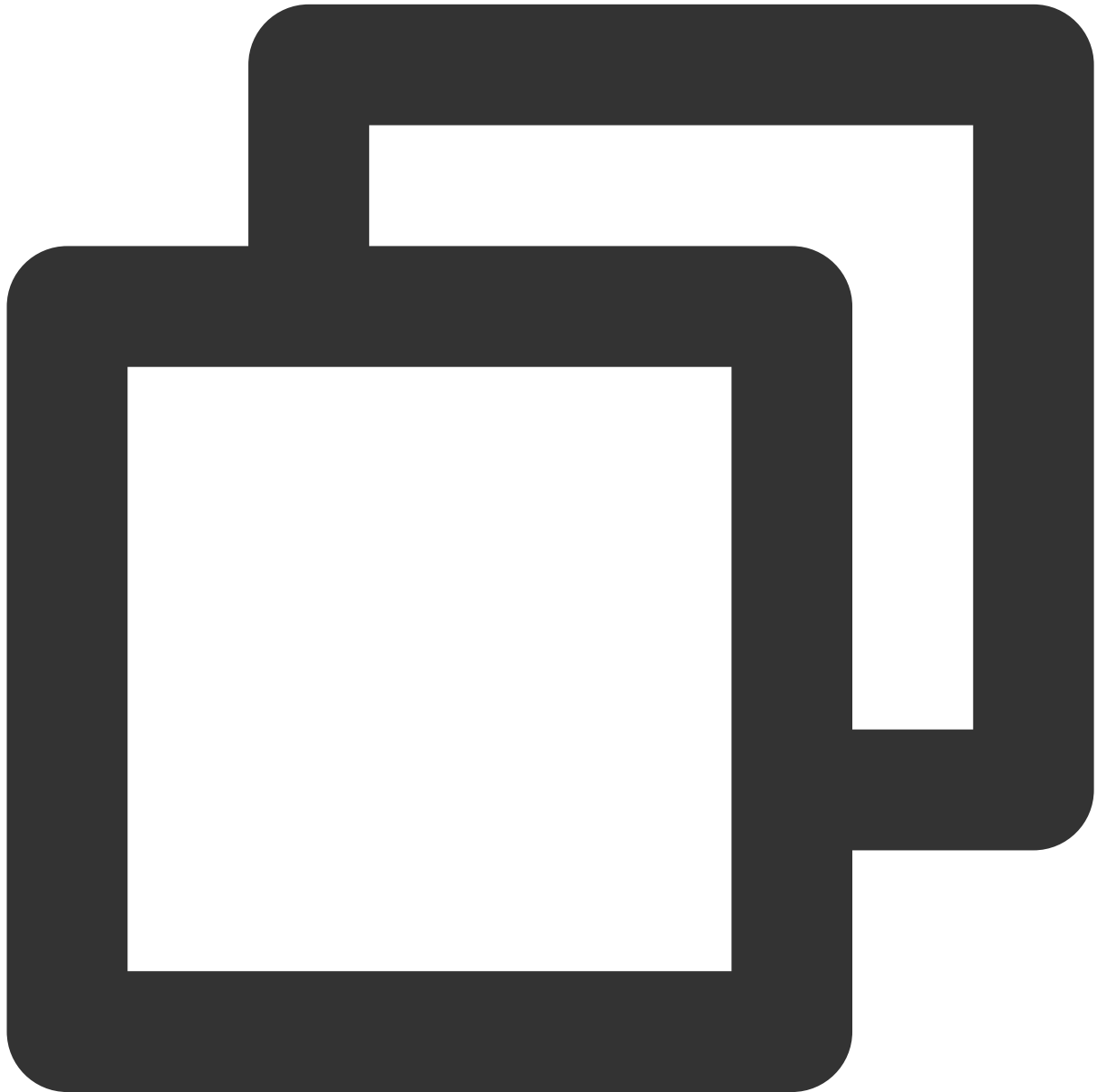
음원 위치 업데이트의 목적은 로컬 플레이어의 위치를 서버에 알리기 위함입니다. 시스템은 **원격 좌표+음성 수신 범위**와 **로컬 좌표+음성 수신 범위**를 확인하여 레인지 보이스를 구현합니다.

이 기능은 음원 위치 정보 업데이트에 사용됩니다. **방 입장이 성공한 후에만 호출**할 수 있으며 **프레임당 한 번만 호출**해야 합니다. Unity 엔진을 예로 들면 Update에서 이 API를 호출해야 합니다.

레인지 보이스가 활성화된 경우 음원 위치를 업데이트해야 합니다. 범위 결정이 필요하지 않더라도 이 **API**는 여전히 **방 입장 후 한 번 호출해야 합니다.**

동시에 3D 음향 효과를 활성화하려면 아래의 **3D 음향 효과** 섹션에 지정된 대로 axisForward, axisRight 및 axisUp 매개변수를 설정합니다.

함수 프로토타입



```
public abstract int UpdateSelfPosition(int position[3], float axisForward[3], float
```

매개변수	유형	설명
------	----	----

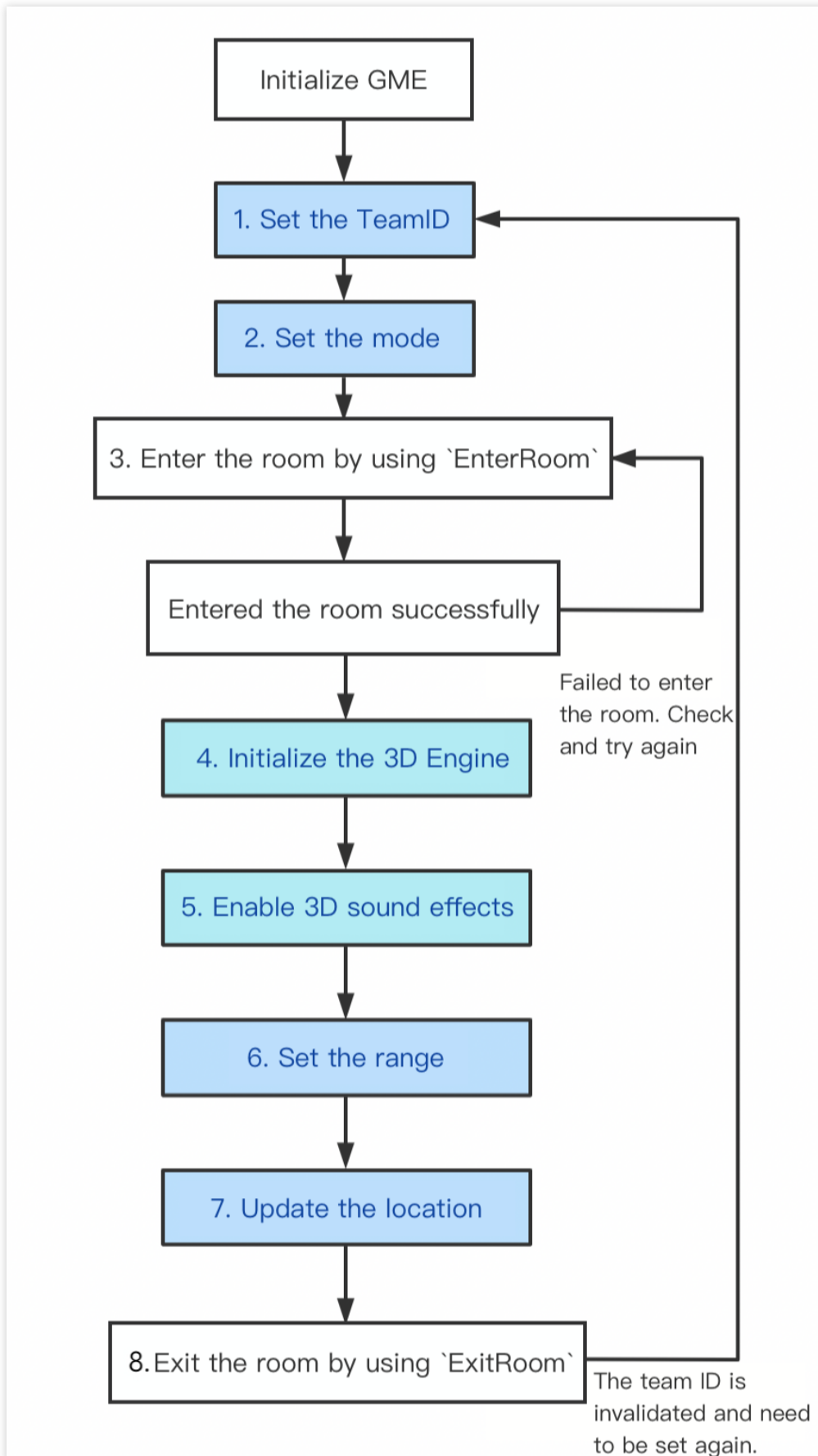
position	int[]	세계 좌표계에서 로컬 위치(전방, 오른쪽, 위)
axisForward	float[]	이 매개변수는 이 제품에서 무시해도 됩니다
axisRight	float[]	이 매개변수는 이 제품에서 무시해도 됩니다
axisUp	float[]	이 매개변수는 이 제품에서 무시해도 됩니다

레인지 보이스와 3D 음향 효과 결합

레인지 보이스 기능은 거리를 통해 소리의 도달 가능성을 제어합니다. 보다 몰입감 있는 경험을 원하신다면 3D 음향 효과와 함께 사용하는 것을 권장합니다.

사용 프로세스

레인지 보이스를 사용하면서 3D 음향 효과를 사용하려면 1, 2, 3단계를 완료한 후 3D 엔진을 초기화하고 3D 음향 효과를 엽니다.



설명 :

순서도의 파란색 부분은 3D 음향 효과의 프로세스를 보여줍니다.

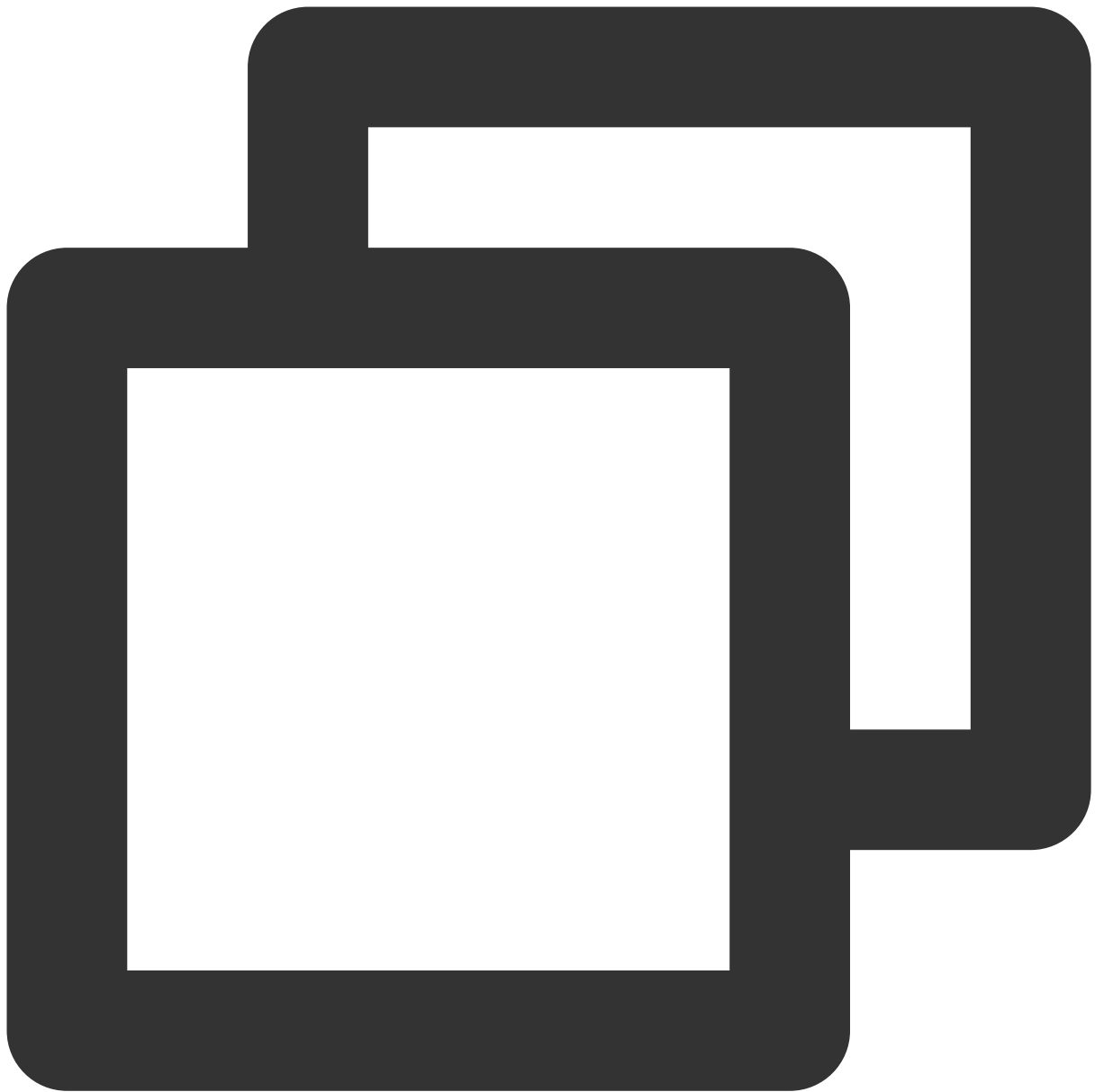
전제 조건

[레인지 보이스 사용](#)을 참고하여 1,2,3 단계를 완료하십시오.

4. 3D 음향 효과 엔진 초기화

이 함수는 3D 음향 효과 엔진을 초기화하는 데 사용되며 방 입장 후 호출해야 합니다. 3D 음향 효과 재생이 아닌 3D 음향 효과 수신만 가능하게 하고 싶은 경우에도 3D 음향 효과 사용 전에 이 API를 호출해야 합니다.

함수 프로토타입



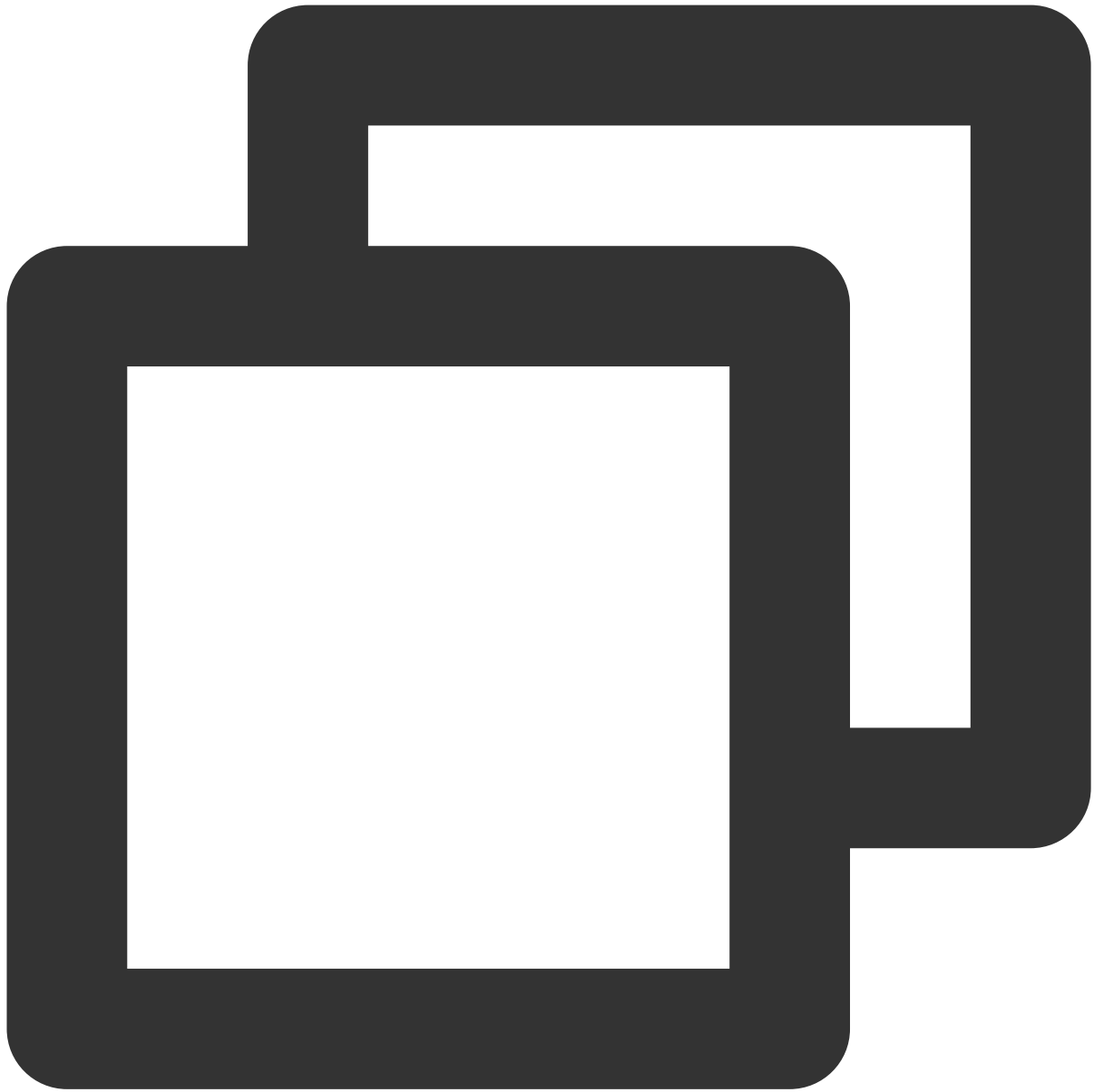
```
public abstract int InitSpatializer(string modelPath)
```

매개변수	유형	설명
modelPath	string	공란만 입력

5. 3D 음향 효과 활성화/비활성화

이 함수는 3D 음향 효과를 활성화하거나 비활성화하는 데 사용됩니다. 활성화 후 3D 음향 효과를 들을 수 있습니다.

함수 프로토타입



```
public abstract int EnableSpatializer(bool enable, bool applyToTeam)
```

매개변수	유형	설명
enable	bool	활성화하면 3D 음향 효과를 들을 수 있습니다
applyToTeam	bool	팀 내 3D 음향 효과 사용 여부를 지정합니다. enable이 true인 경우에만 적용됩니다.

IsEnableSpatializer API를 사용하여 3D 음향 효과 상태를 가져올 수 있습니다.

6. 음성이 들리는 범위 설정 (3D 음향 효과용)

이 메소드는 음성 수신 범위(게임 엔진에 따름) 설정에 사용되며 **방 입장 성공 후에만 호출할 수 있습니다.**

이 메소드는 UpdateSelfPosition과 함께 사용하여 음원 위치를 업데이트해야 합니다.

이 메소드는 한 번만 호출하면 됩니다.

3D 음향 효과에서 사운드 소스까지의 거리가 지정된 임계값(range/10)을 초과하면 사운드가 거의 0으로 감쇠되기 시작합니다.

거리와 소음 감쇠의 관계에 대한 자세한 내용은 부록을 참고하십시오.

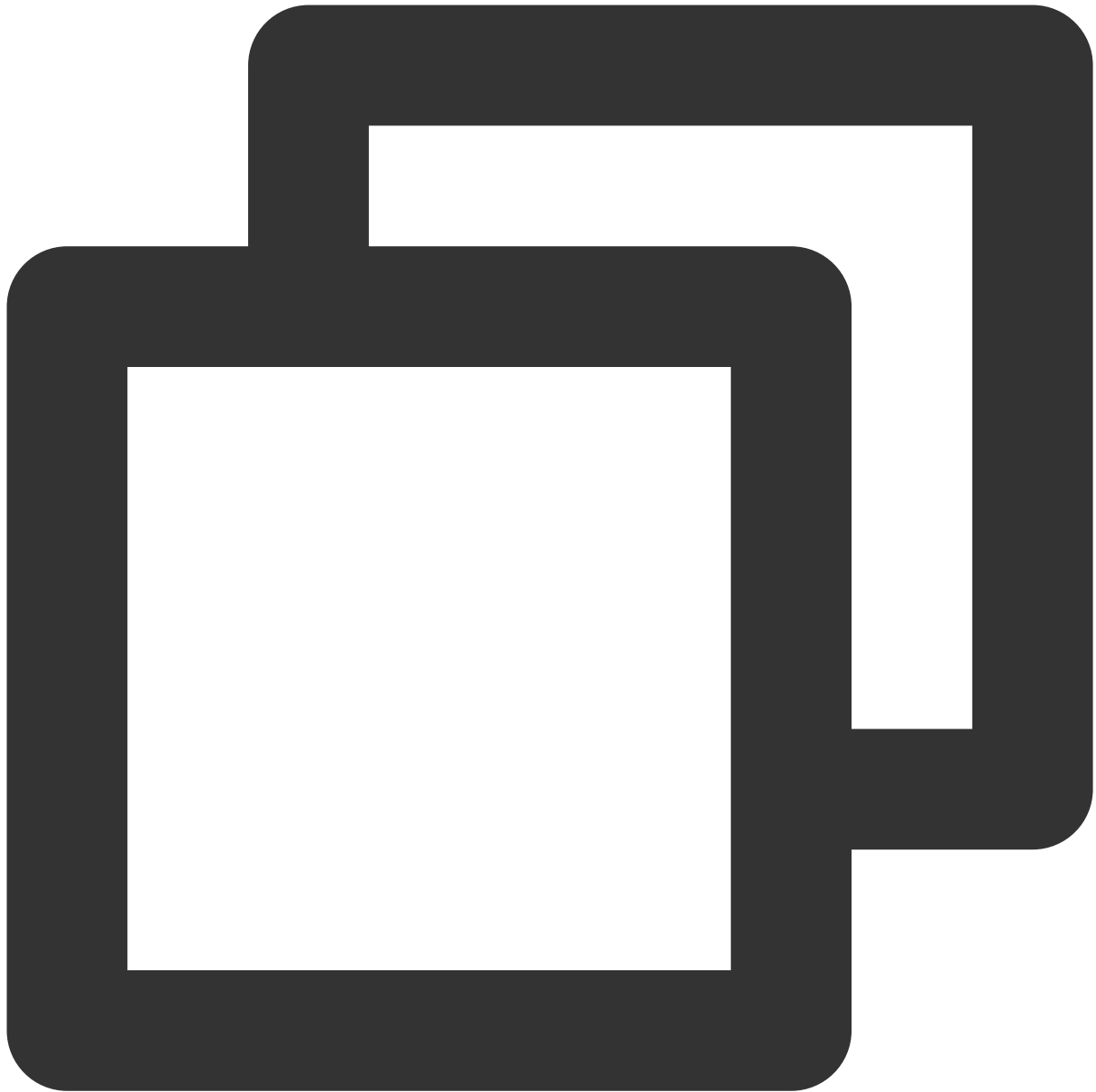
함수 프로토타입



```
ITMGRoom int UpdateAudioRecvRange(int range)
```

매개변수	유형	설명
range	int	게임 엔진에서 사용하는 거리 단위의 최대 음성 수신 범위

예시 코드



```
ITMGContext.GetInstance().GetRoom().UpdateAudioRecvRange(300);
```

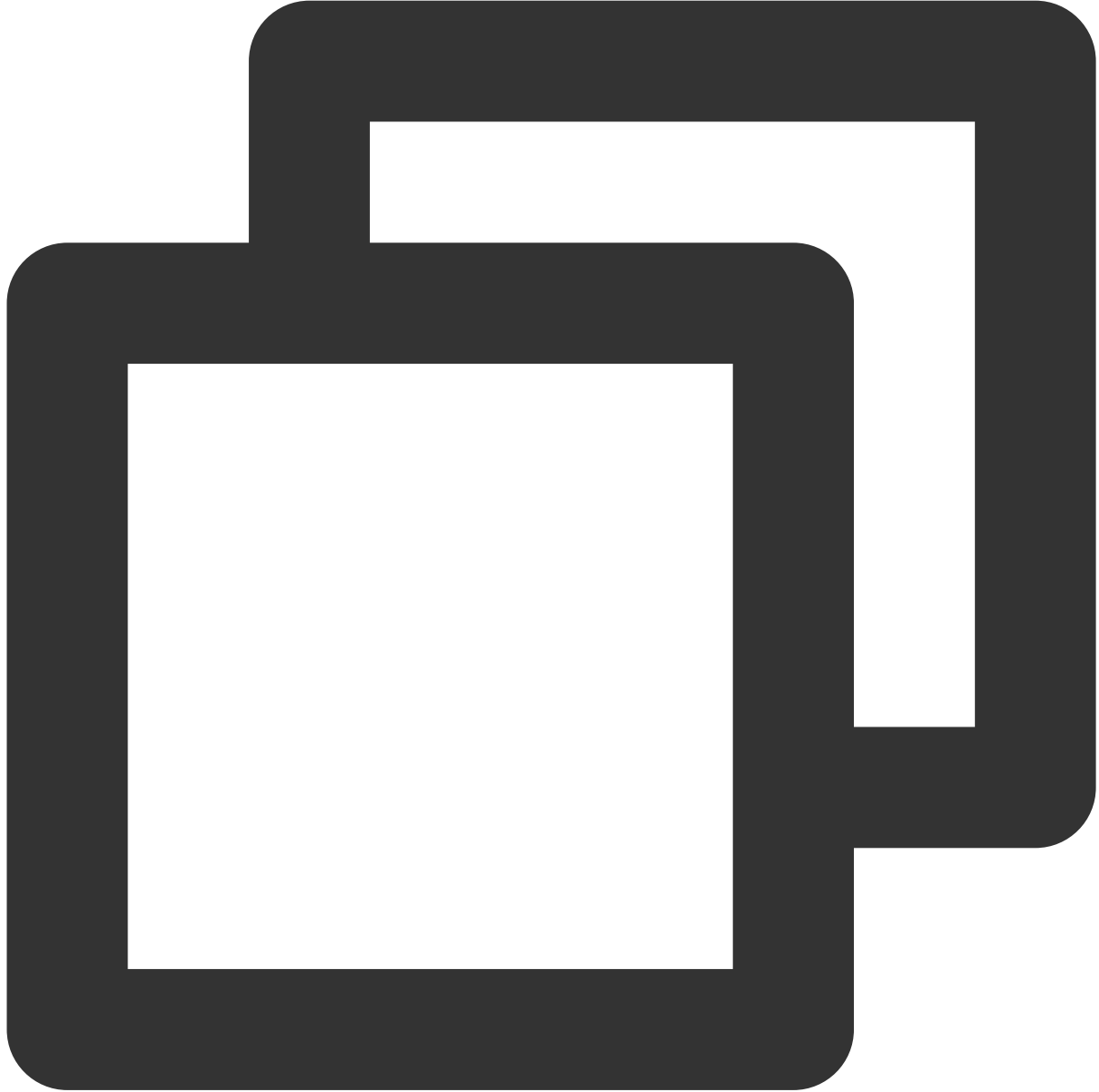
7. 음원 위치 업데이트(3D 음향 효과용)

음원 위치 업데이트의 목적은 로컬 플레이어의 위치를 서버에 알리기 위함입니다. 시스템은 **원격 좌표+음성 수신 범위**와 **로컬 좌표+음성 수신 범위**를 확인하여 레인지 보이스를 구현합니다.

이 기능은 음원 위치 정보를 업데이트하는 데 사용됩니다. **방 입장이 성공한 후에만 호출할 수 있으며 프레임당 한 번만 호출**해야 합니다. Unity 엔진을 예로 들면 Update에서 이 API를 호출해야 합니다.

예시 코드를 복사하고 호출하여 이 기능을 사용할 수 있습니다.

함수 프로토타입



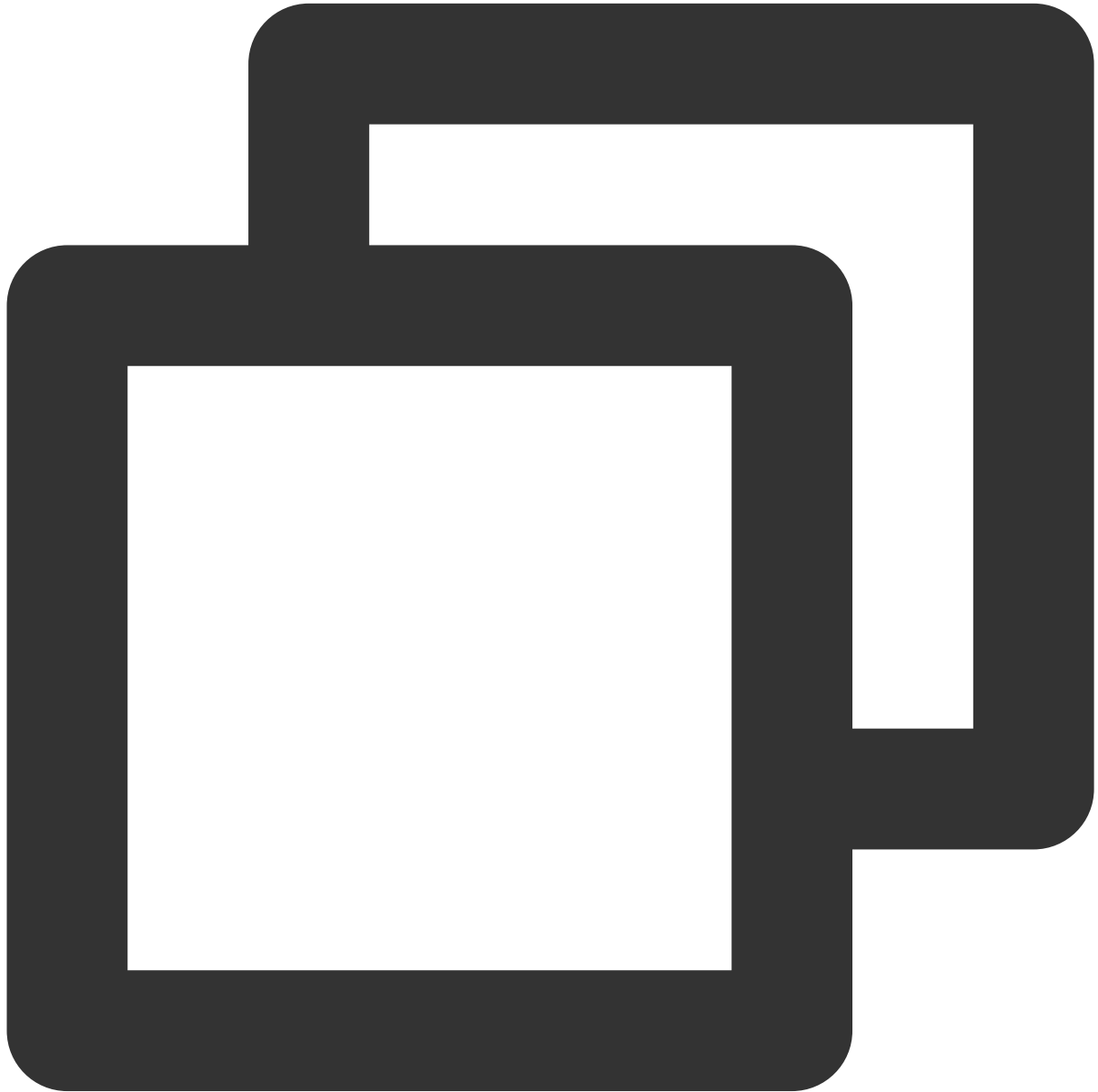
```
public abstract int UpdateSelfPosition(int position[3], float axisForward[3], float
```

매개변수	유형	설명
position	int[]	세계 좌표계에서 로컬 위치(전방, 오른쪽, 위)
axisForward	float[]	로컬 좌표계의 전방 벡터

axisRight	float[]	로컬 좌표계의 오른쪽 벡터
axisUp	float[]	로컬 좌표계의 위 벡터

예시 코드

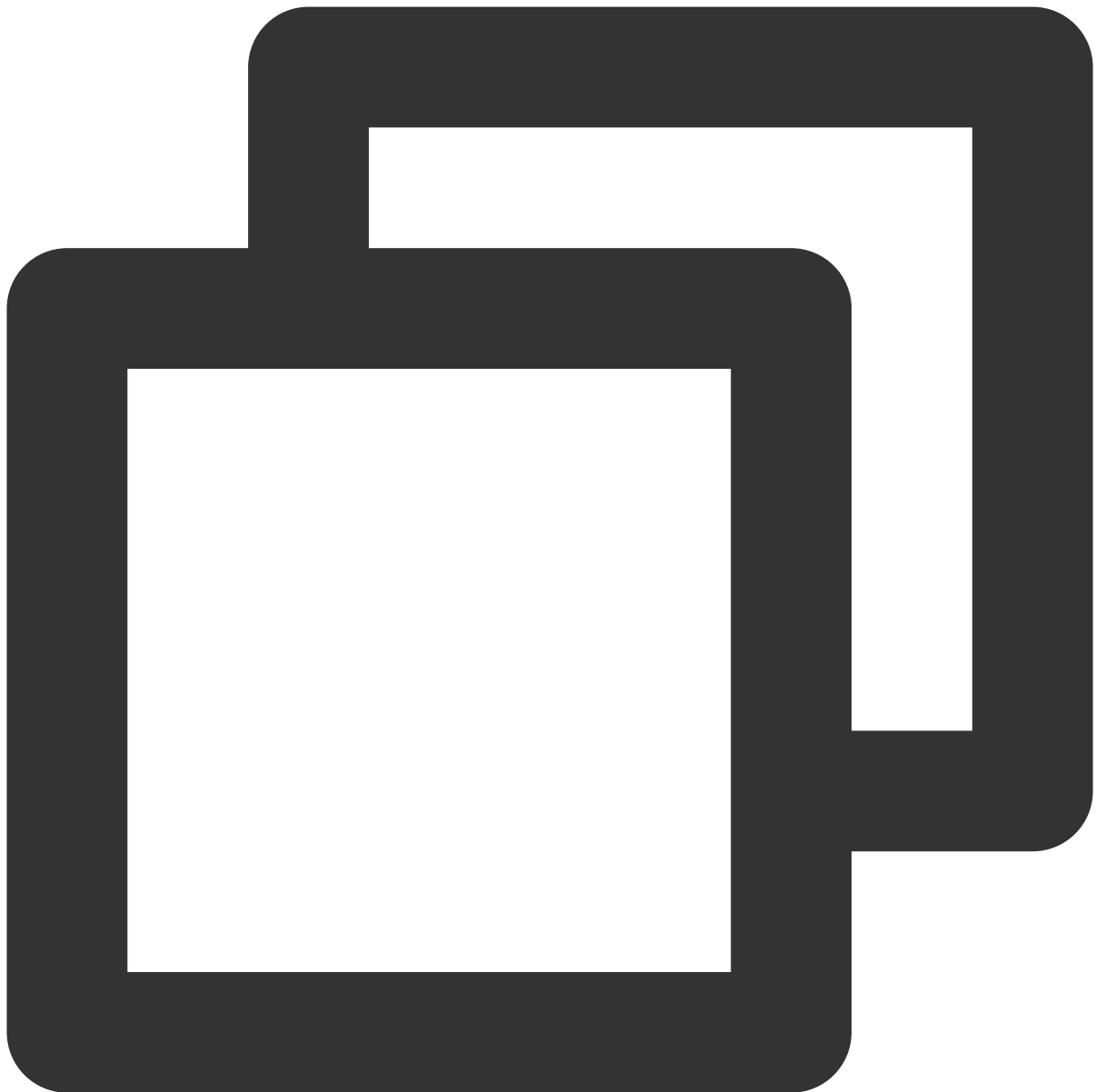
Unreal



```
FVector cameraLocation = UGameplayStatics::GetPlayerCameraManager(GetWorld(), 0)->G
FRotator cameraRotation = UGameplayStatics::GetPlayerCameraManager(GetWorld(), 0)->
int position[] = {
```

```
(int)cameraLocation.X,  
(int)cameraLocation.Y,  
(int)cameraLocation.Z };  
FMatrix matrix = ((FRotationMatrix)cameraRotation);  
float forward[] = {  
    matrix.GetColumn(0).X,  
    matrix.GetColumn(1).X,  
    matrix.GetColumn(2).X };  
float right[] = {  
    matrix.GetColumn(0).Y,  
    matrix.GetColumn(1).Y,  
    matrix.GetColumn(2).Y };  
float up[] = {  
    matrix.GetColumn(0).Z,  
    matrix.GetColumn(1).Z,  
    matrix.GetColumn(2).Z};  
ITMGContextGetInstance()->GetRoom()->UpdateSelfPosition(position, forward, right, u
```

Unity



```
Transform selftrans = currentPlayer.gameObject.transform;
Matrix4x4 matrix = Matrix4x4.TRS(Vector3.zero, selftrans.rotation, Vector3.one);
int[] position = new int[3] {
    selftrans.position.z,
    selftrans.position.x,
    selftrans.position.y};
float[] axisForward = new float[3] {
    matrix.m22,
    matrix.m02,
    matrix.m12 };
float[] axisRight = new float[3] {
```

```

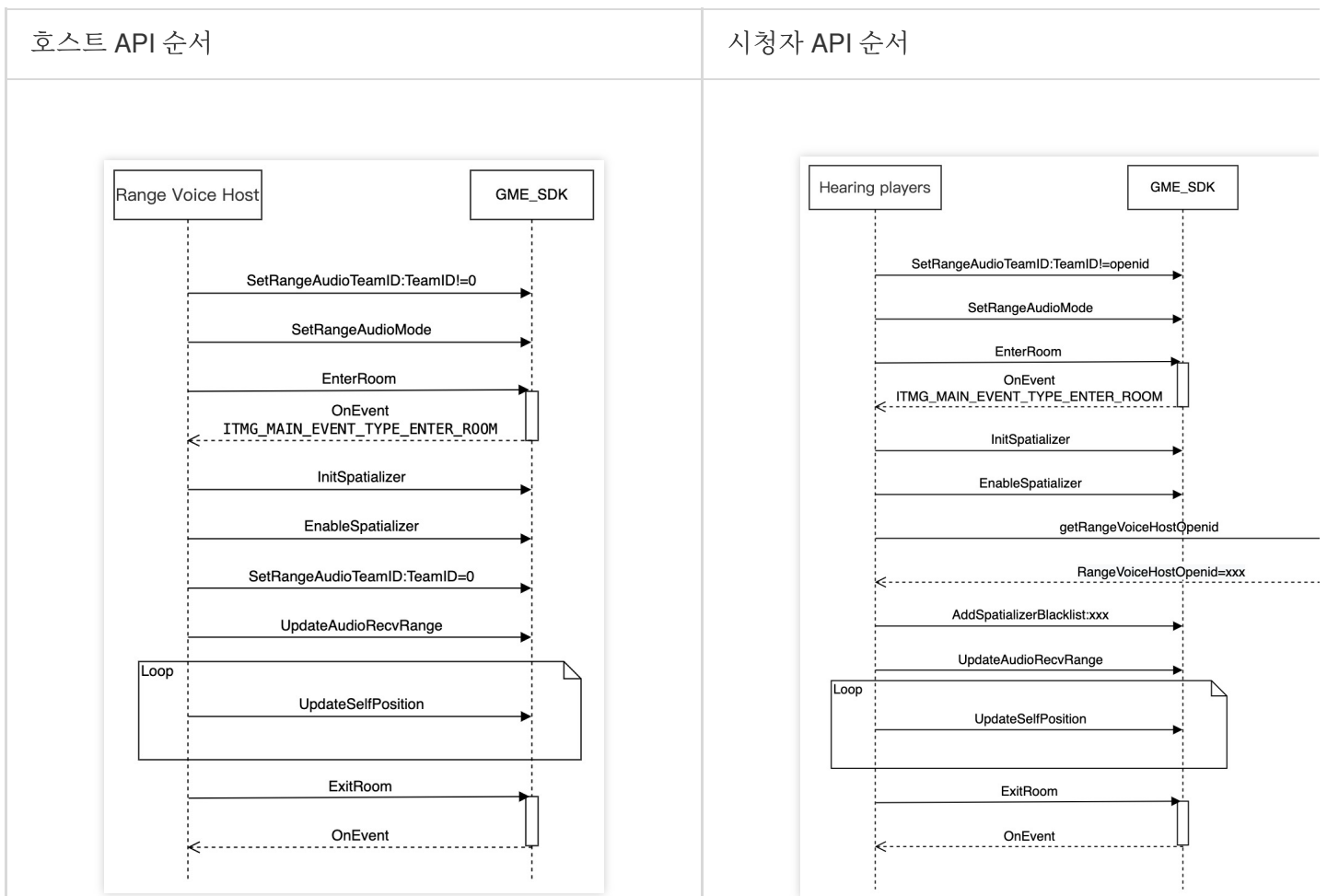
matrix.m20,
matrix.m00,
matrix.m10 };
float[] axisUp = new float[3] {
matrix.m21,
matrix.m01,
matrix.m11 };
ITMGContext.GetInstance().GetRoom().UpdateSelfPosition(position, axisForward, axisR

```

호스트 3D 음향 효과 사용 중지

시나리오에 레인지 보이스 호스트 모드를 사용하는 플레이어가 있는 경우, 즉 방에 있는 모든 플레이어가 그의 음성을 들어야 하는 경우 **3D 음성 블록리스트 API**를 참고하여 모든 수신측에서 3D 음성 블록리스트에 호스트를 추가합니다. 이를 통해 3D 음성 기능의 감쇠 효과가 호스트의 사운드 도달 가능성에 미치는 영향을 방지합니다.

다양한 역할의 API 호출 순서는 다음과 같습니다.



부록

음성 모드

각 음성 모드의 플레이어 음성 도달 가능성은 다음과 같습니다.

플레이어 A가 '모든 사람' 모드를 선택한 경우, 각 음성 모드에서 플레이어 B의 음성 도달 가능성은 다음과 같습니다:

같은 팀 여부	범위 내 여부	음성 모드	A와 B가 서로 들을 수 있는지 여부
Yes	Yes	MODE_WORLD	Yes
		MODE_TEAM	Yes
	No	MODE_WORLD	Yes
		MODE_TEAM	Yes
Yes	Yes	MODE_WORLD	Yes
		MODE_TEAM	No
	No	MODE_WORLD	No
		MODE_TEAM	No

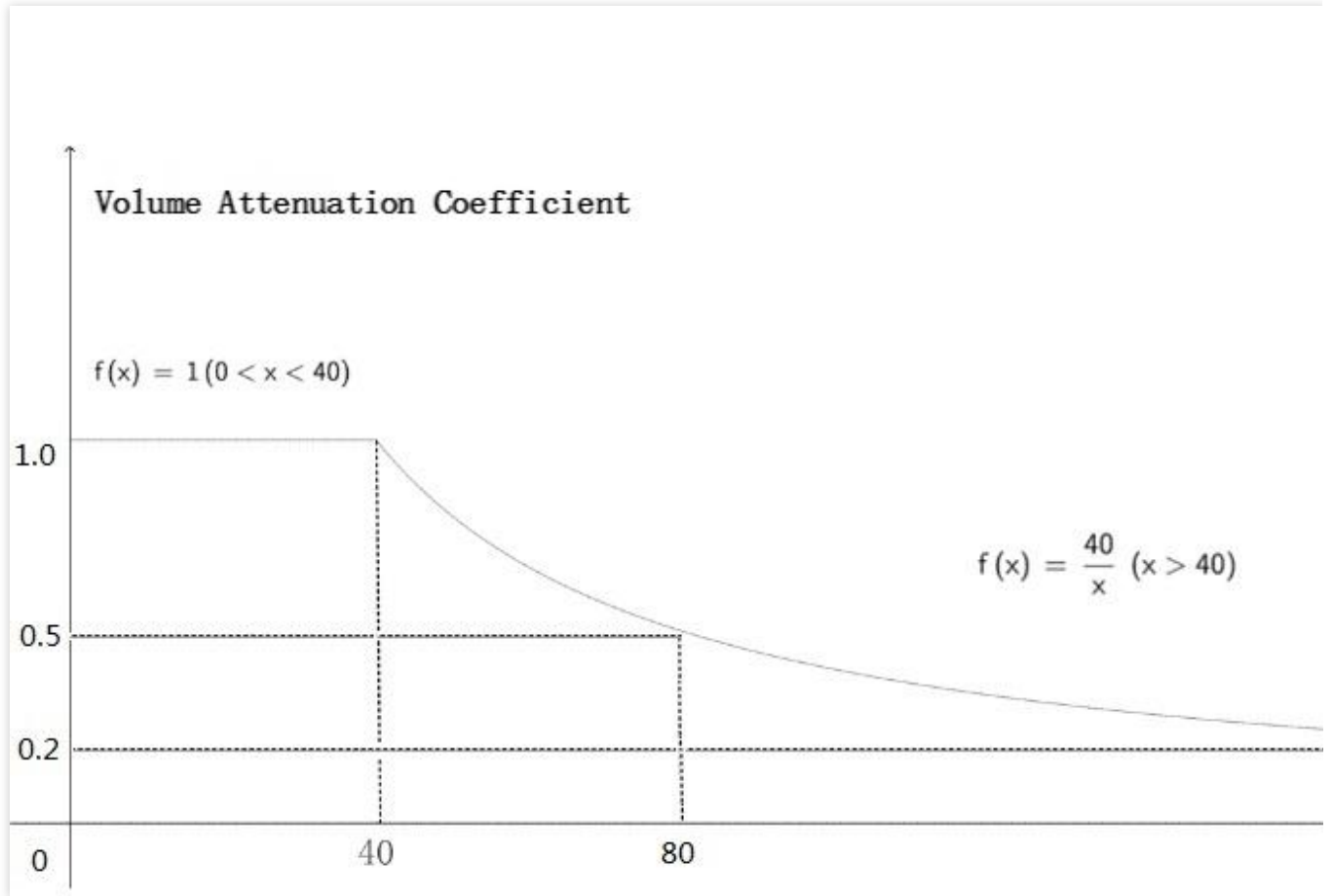
플레이어 A가 '팀원만' 모드를 선택한 경우, 각 음성 모드에서 플레이어 B의 음성 도달 가능성은 다음과 같습니다:

같은 팀 여부	범위 내 여부	음성 모드	A와 B가 서로 들을 수 있는지 여부
Yes	Yes	MODE_WORLD	Yes
		MODE_TEAM	Yes
	No	MODE_WORLD	Yes
		MODE_TEAM	Yes
Yes	Yes	MODE_WORLD	No
		MODE_TEAM	No
	No	MODE_WORLD	No
		MODE_TEAM	No

거리와 사운드 감쇠 관계

3D 음향 효과에서 사운드 소스까지의 거리가 지정된 임계값(range/10)을 초과하면 사운드가 거의 0으로 감쇠되기 시작합니다.

거리(엔진 단위)	감쇠 공식
$0 < N < range/10$	감쇠 계수: 1.0(감쇠 없음)

$N \geq \text{range}/10$ 감쇠 계수: $\text{range}/10/N$ 

3D 음향 효과

최종 업데이트 날짜: : 2024-01-18 16:00:04

본문은 3D 음향 효과를 위해 GME(Game Multimedia Engine) API를 통합하고 디버깅하는 방법을 설명합니다.

사용 사례

방 입장을 위한 일반적인 음성 채팅에서 플레이어 음성에는 3D 음향 효과가 없으며 플레이어 간에 간단한 인터랙션만 가능합니다. 3D 음향 효과로 플레이어는 말을 하는 동안 자신의 방향과 위치 정보를 노출할 수 있으며 거리에 따라 목소리가 실시간으로 변할 수 있습니다. 3D 음향 효과 기능은 <배틀 로얄 게임>에서 플레이어에게 보다 사실적이고 몰입감 있는 커뮤니케이션 및 전투 경험을 제공합니다.

[demo 다운로드](#)를 클릭하여 3D 음향 효과 기능을 사용해 볼 수 있습니다.

전제 조건

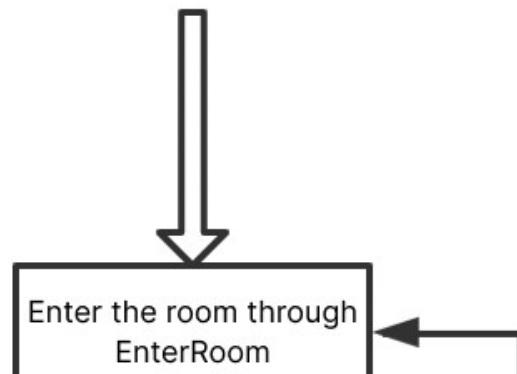
음성 채팅 서비스 활성화: [음성 서비스 활성화 가이드](#)를 참고하십시오.

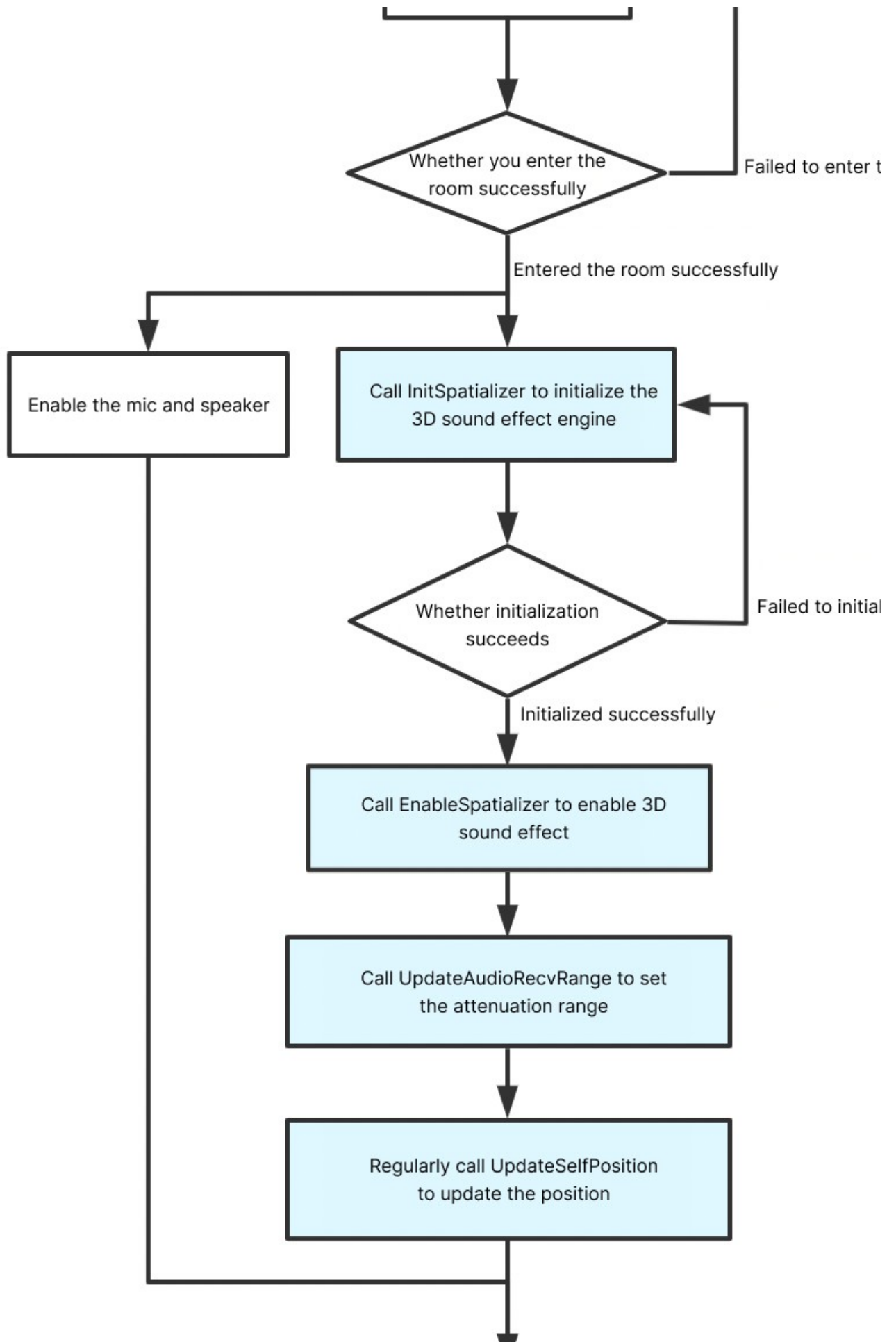
GME SDK 통합: 핵심 API 및 음성 채팅 API를 포함한 GME SDK를 통합 완료해야 합니다. 자세한 내용은 [Native SDK 빠른 통합](#), [Unity SDK 빠른 통합](#) 및 [Unreal SDK 빠른 통합](#)을 참고하십시오.

구현 프로세스

구현 순서도

3D 음향 효과 구현 순서도는 다음과 같습니다. 파란색 부분은 방 입장을 위한 일반 음성 채팅과 비교하여 새로운 통합 단계입니다.



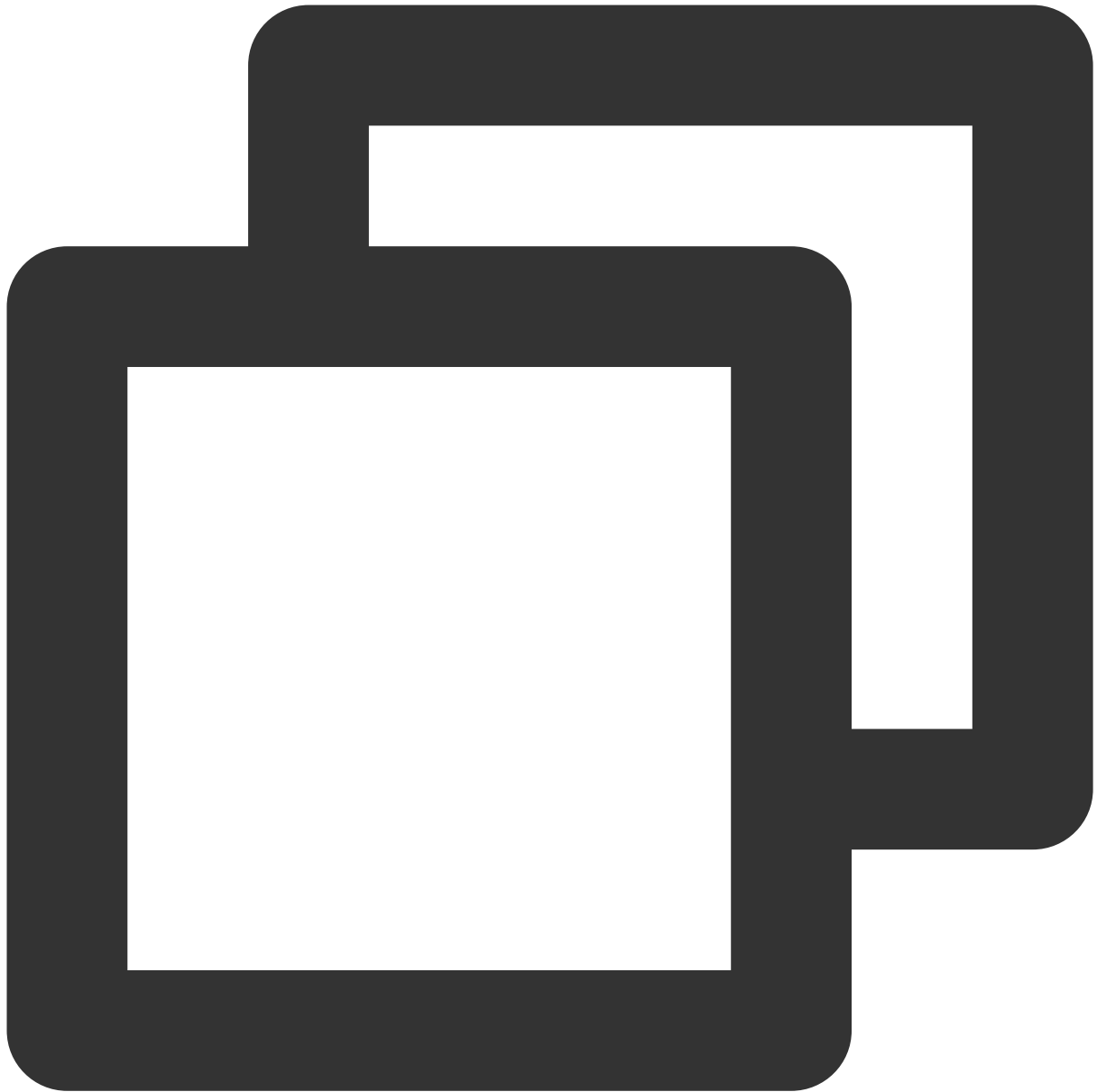




3D 음향 효과 엔진 초기화

이 함수는 3D 음향 효과 엔진을 초기화하는 데 사용되며 방 입장 후 호출해야 합니다. 3D 음향 효과 재생이 아닌 3D 음향 효과 수신만 가능하게 하고 싶은 경우에도 3D 음향 효과 사용 전에 이 API를 호출해야 합니다.

함수 프로토타입



```
public abstract int InitSpatializer(string modelPath)
```

매개변수	유형	설명
modelPath	string	3D 음향 효과 리소스 파일의 절대 경로

3D 음향 효과 리소스 파일을 로컬 디스크에 다운로드해야 합니다. SDK 버전에 따라 하나를 선택하십시오.

v2.8 이전 SDK의 경우 [여기](#)를 클릭하여 다운로드하십시오(md5 값: d0b76aa64c46598788c2f35f5a8a8694).

v2.8-v2.9.5 SDK의 경우 [여기](#)를 클릭하여 다운로드하십시오(md5 값: 3d4d04b3949e267e34ca809e8a0b9243).

v2.9.6 이상 버전인 경우 3D 음향 효과 리소스 파일이 이미 내장되어 있으므로 여기에 `modelPath`를 입력할 수 있습니다.

GME SDK 릴리스 업데이트는 [제품 업데이트](#)를 참고하십시오.

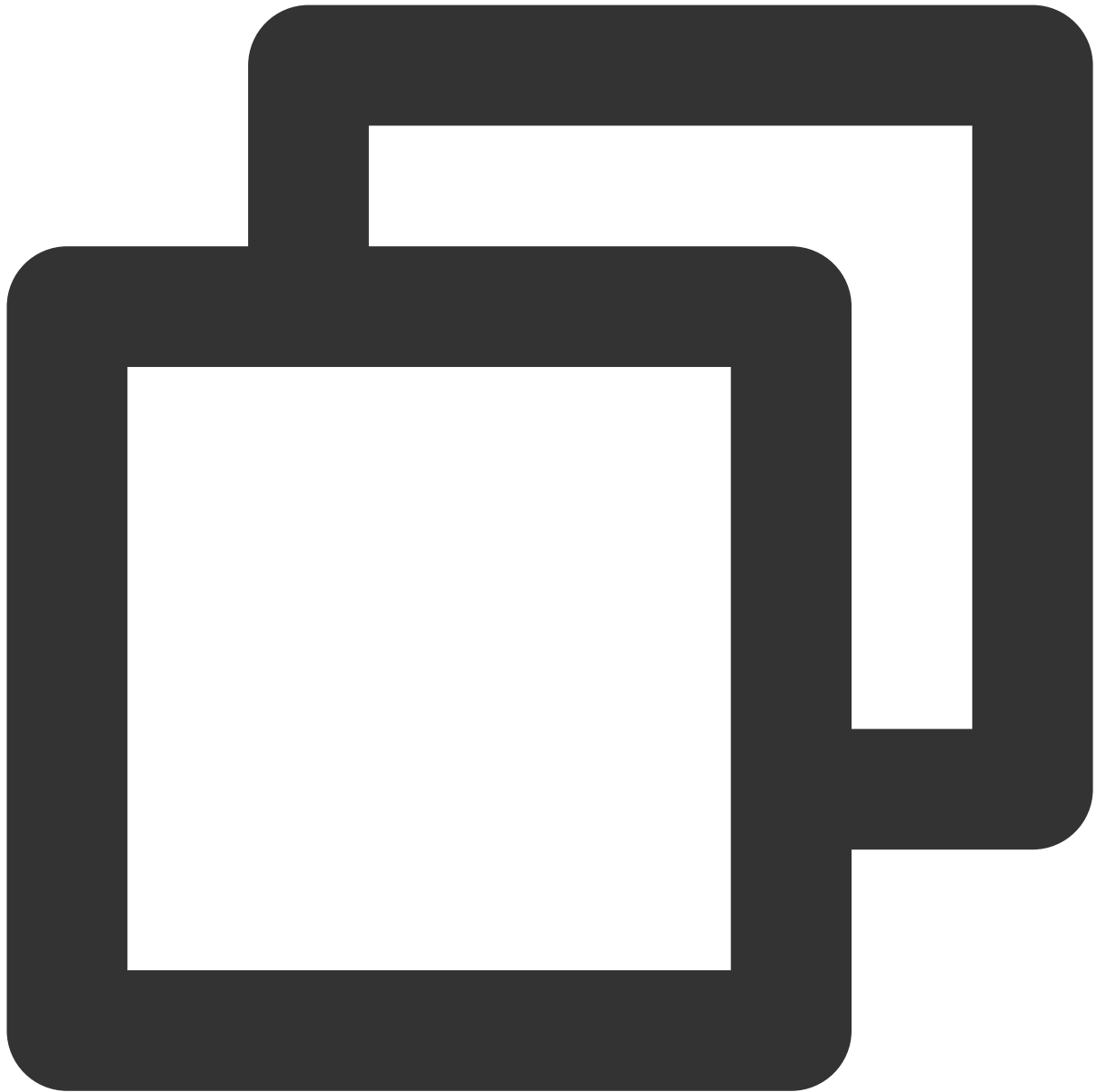
리소스 경로 정보

Unity를 예로 들면 3D 음향 효과 리소스 파일을 프로젝트의 `StreamingAssets` 디렉터리에 넣고 `SampleCode`의 `copyFileFromAssetsToPersistent` 함수 내용에 따라 다른 플랫폼의 해당 디렉터리에 복사하는 것이 좋습니다. Unreal Engine을 예로 들어 3D 모델 파일을 복사하고 `SampleCode`의 `CopyAllAssetsToExternal` 함수 내용을 기반으로 경로를 읽습니다.

3D 음향 효과 활성화/비활성화

이 함수는 3D 음향 효과를 활성화하거나 비활성화하는 데 사용됩니다. 활성화 후 3D 사운드를 들을 수 있습니다.

함수 프로토타입



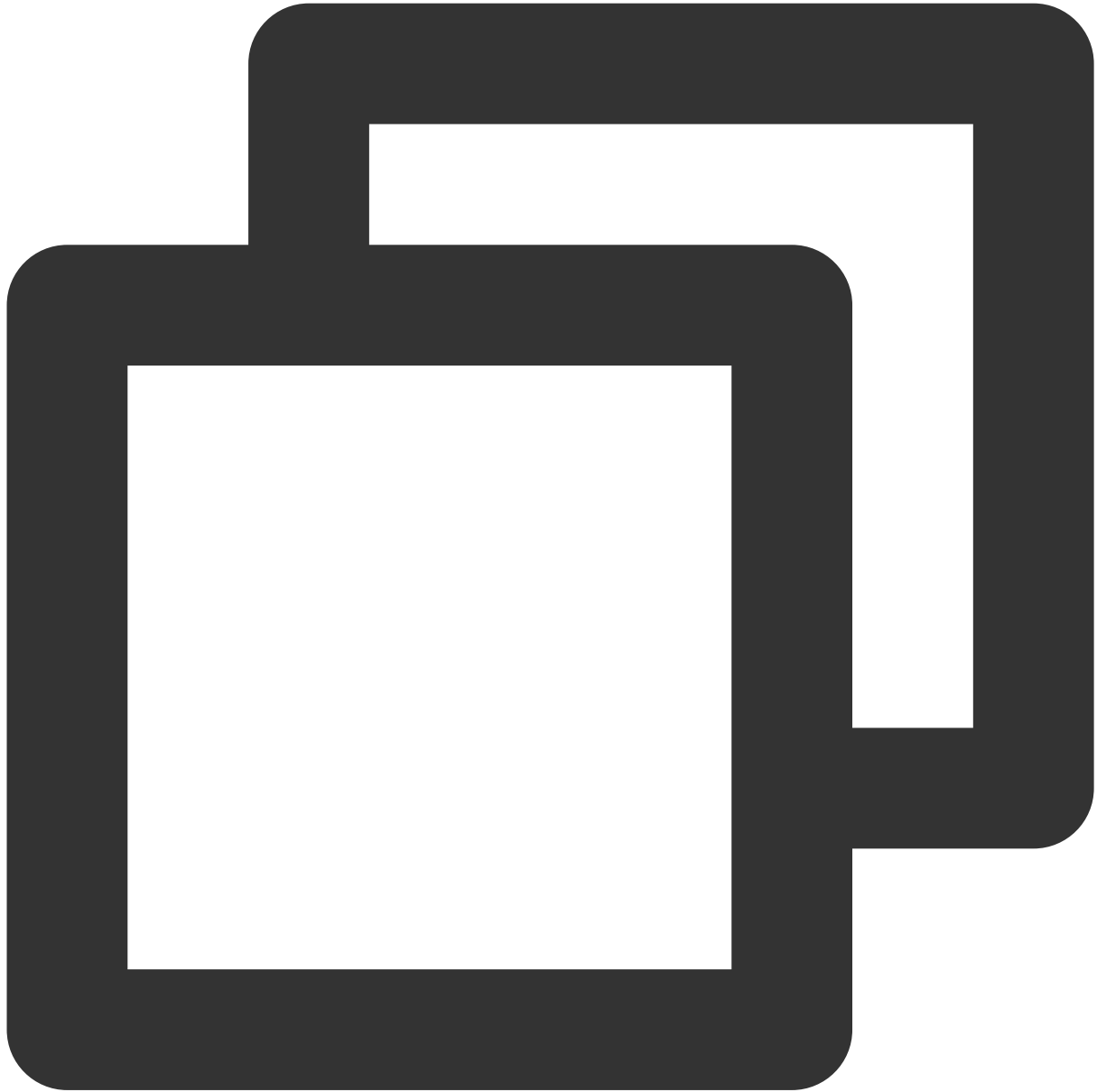
```
public abstract int EnableSpatializer(bool enable, bool applyToTeam)
```

매개변수	유형	설명
enable	bool	활성화하면 3D 음향 효과를 들을 수 있습니다
applyToTeam	bool	팀 내 3D 음향 효과 활성화 여부를 지정합니다. enable이 true인 경우에만 적용됩니다.

3D 음향 효과 상태 가져오기

이 함수는 3D 음향 효과의 상태를 얻기 위해 사용됩니다.

함수 프로토타입



```
public abstract bool IsEnableSpatializer()
```

반환 값	설명
true	활성화 상태
false	비활성화 상태

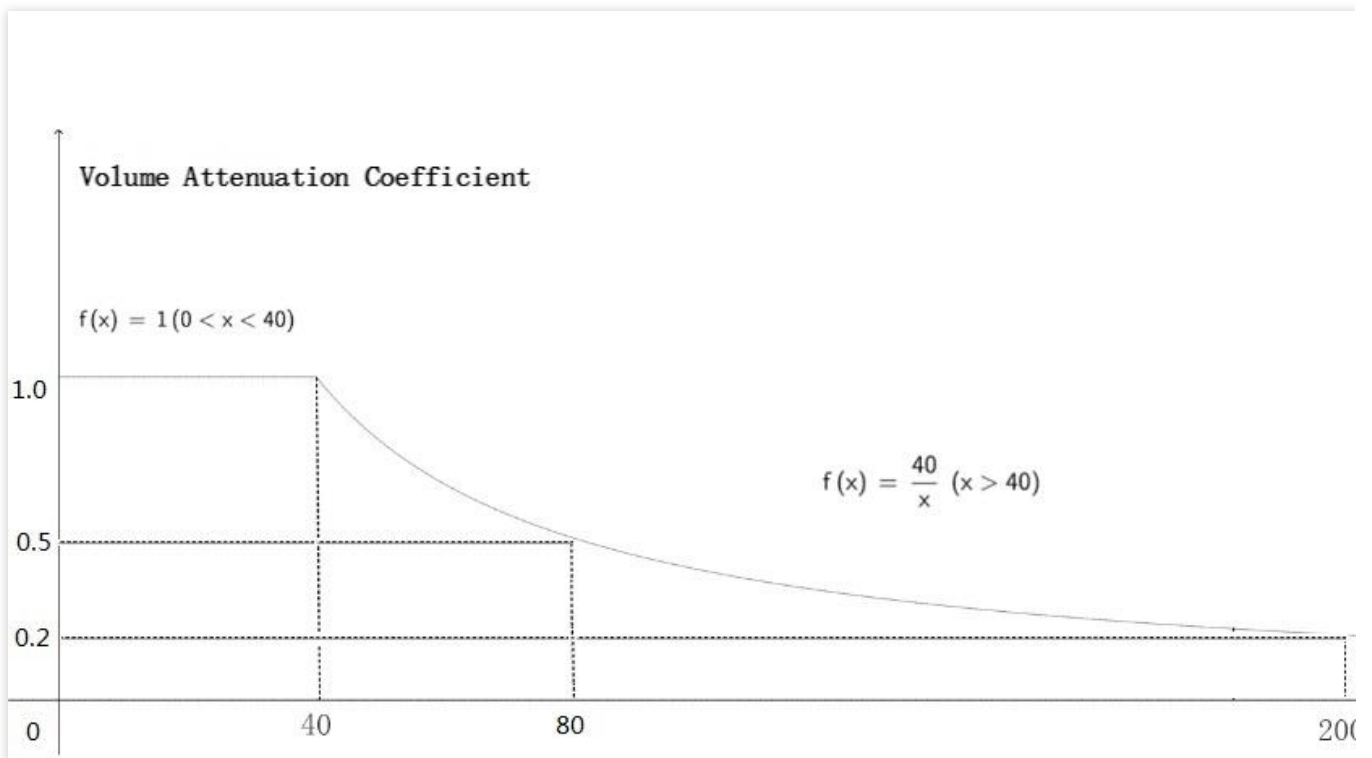
3D 음향 효과 감쇠 범위 설정

3D 음향 효과 감쇠 범위를 설정해야 합니다. 100으로 설정하는 것이 좋습니다.

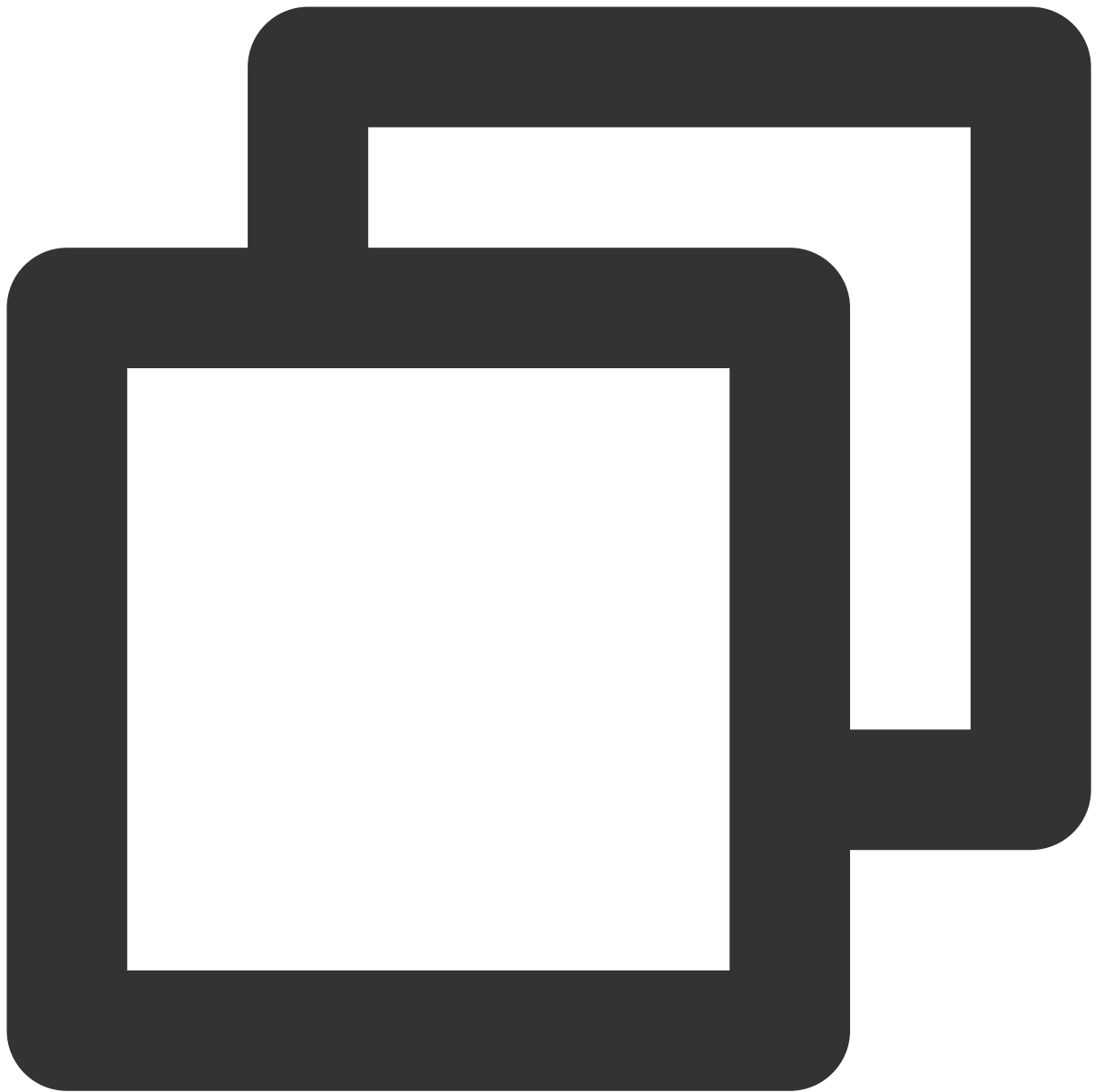
거리와 사운드 감쇠의 관계

3D 음향 효과에서 음원까지의 거리가 지정된 임계값(range/10)을 초과하면 볼륨이 거의 0으로 감쇠되기 시작합니다.

거리 범위(엔진 단위)	감쇠 공식
$0 < N < \text{range}/10$	감쇠 계수: 1.0(볼륨 감쇠 없음)
$N \geq \text{range}/10$	감쇠 계수: $\text{range}/10/N$



함수 프로토타입



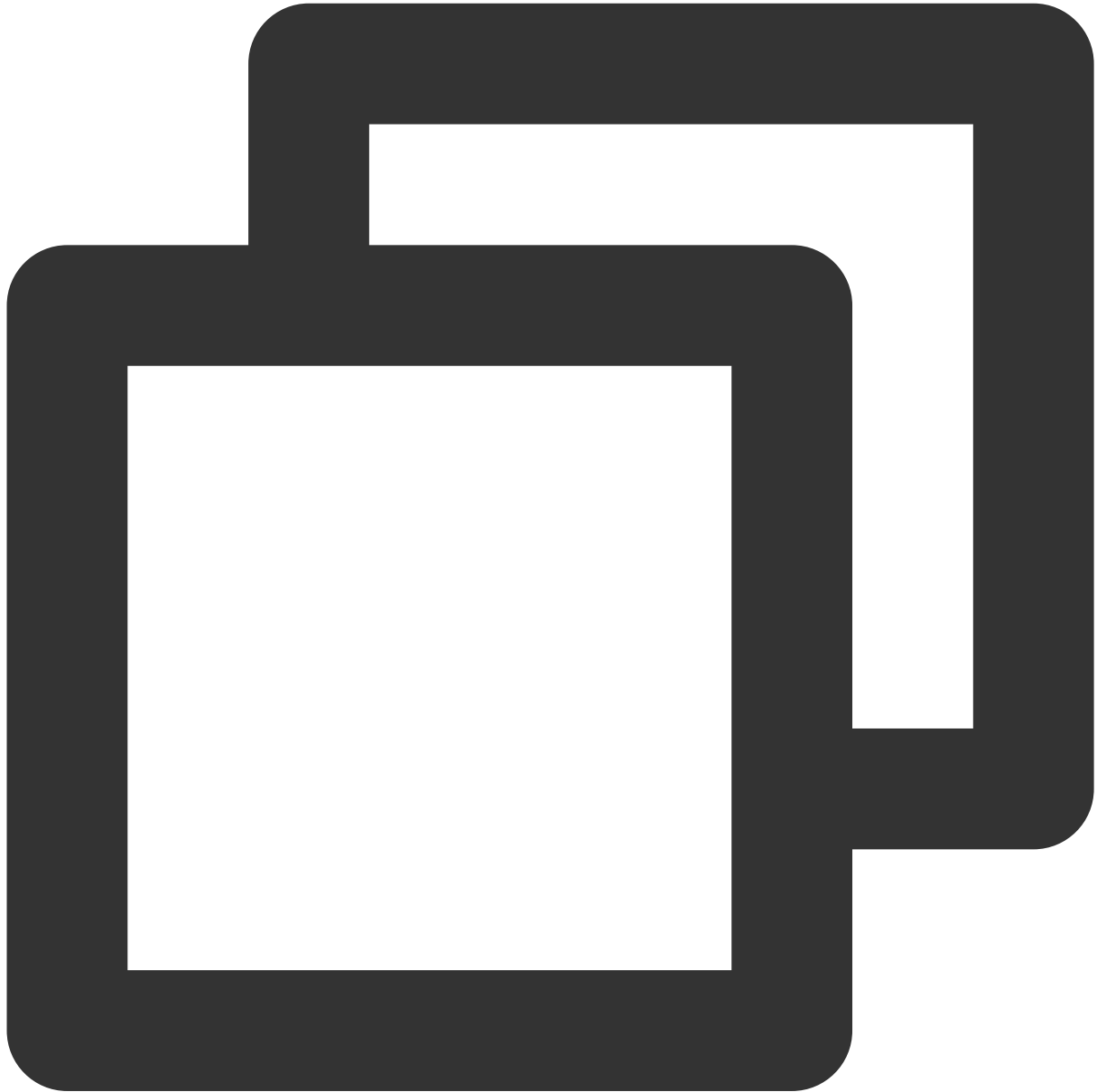
```
public abstract void UpdateAudioRecvRange(int range)
```

매개변수	유형	설명
range	int	게임 엔진에서 사용하는 거리 단위로 음향 효과를 들을 수 있는 범위입니다. 100으로 설정하는 것이 좋습니다.

음원 위치 업데이트(방향 포함)

이 함수는 음원의 위치 정보를 업데이트하는 데 사용됩니다. 3D 음향 효과를 구현하기 위해 모든 프레임을 호출할 수 있습니다.

함수 프로토타입



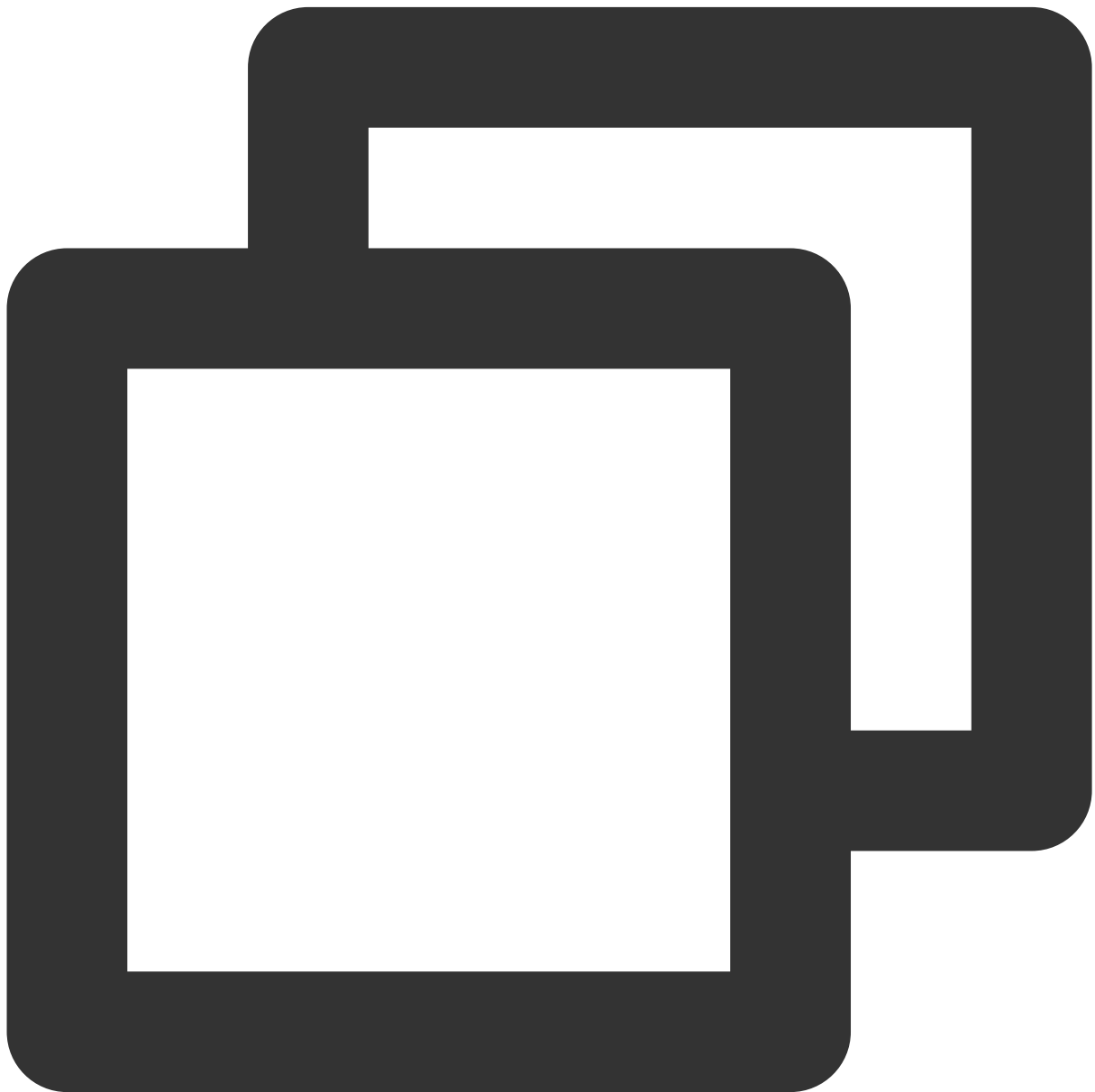
```
public abstract int UpdateSelfPosition(int position[3], float axisForward[3], float
```

매개변수	유형	설명
position	int[]	세계 좌표계에서 로컬 위치(전방, 오른쪽, 위)

axisForward	float[]	로컬 좌표계의 전방 벡터
axisRight	float[]	로컬 좌표계의 오른쪽 벡터
axisUp	float[]	로컬 좌표계의 위쪽 벡터

예시 코드

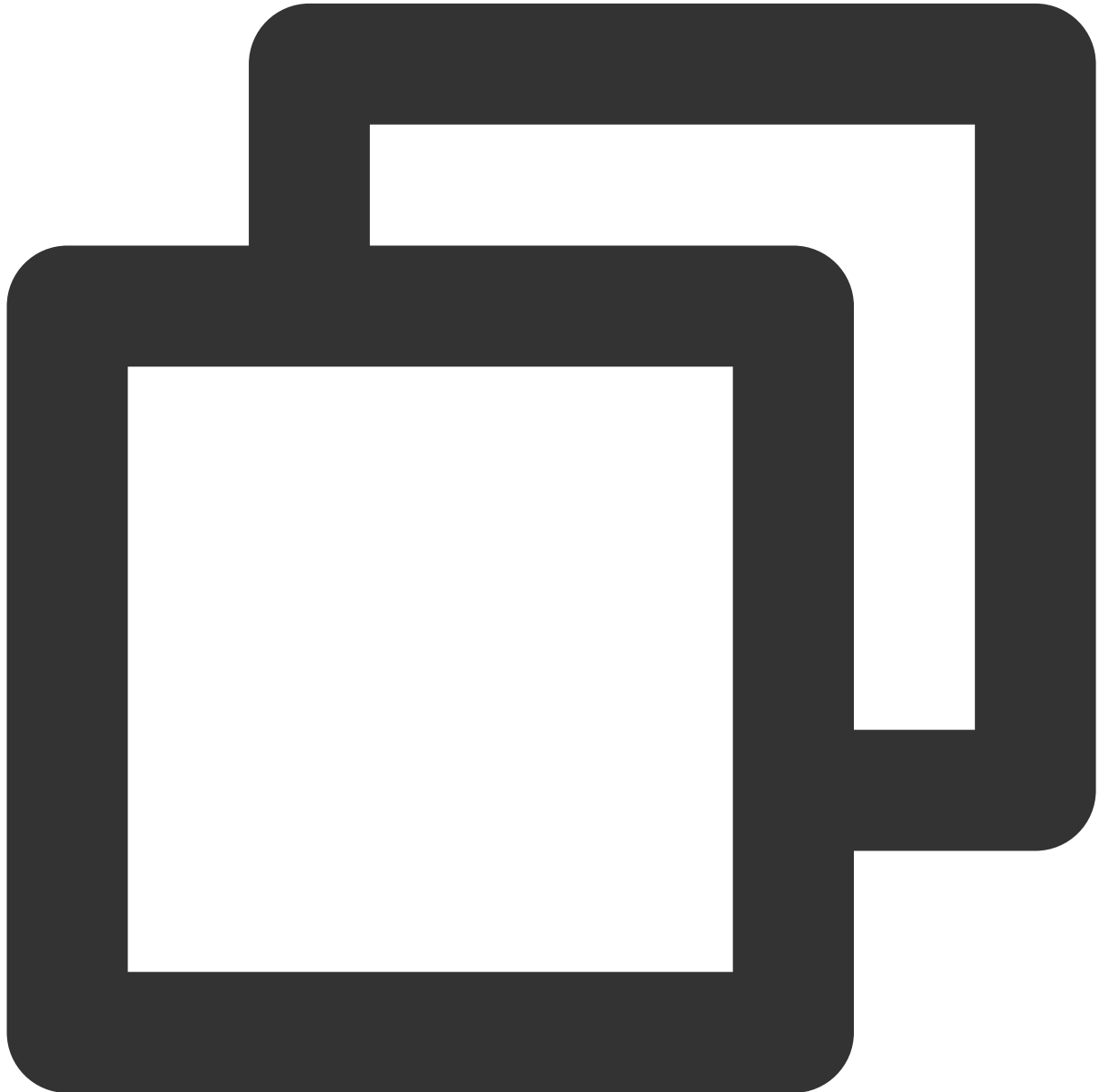
Unreal



```
FVector cameraLocation = UGameplayStatics::GetPlayerCameraManager(GetWorld(), 0)->G  
FRotator cameraRotation = UGameplayStatics::GetPlayerCameraManager(GetWorld(), 0)->
```

```
int position[] = { (int)cameraLocation.X, (int)cameraLocation.Y, (int)cameraLocation.Z};
FMatrix matrix = ((FRotationMatrix)cameraRotation);
float forward[] = { matrix.GetColumn(0).X, matrix.GetColumn(1).X, matrix.GetColumn(2).X};
float right[] = { matrix.GetColumn(0).Y, matrix.GetColumn(1).Y, matrix.GetColumn(2).Y};
float up[] = { matrix.GetColumn(0).Z, matrix.GetColumn(1).Z, matrix.GetColumn(2).Z};
ITMGContextGetInstance()->GetRoom()->UpdateSelfPosition(position, forward, right, up);
```

Unity



```
Transform selftrans = currentPlayer.gameObject.transform;
Matrix4x4 matrix = Matrix4x4.TRS(Vector3.zero, selftrans.rotation, Vector3.one);
int[] position = new int[3] { selftrans.position.z, selftrans.position.x, selftrans.position.y};
```



```
float[] axisForward = new float[3] { matrix.m22, matrix.m02, matrix.m12 };
float[] axisRight = new float[3] { matrix.m20, matrix.m00, matrix.m10 };
float[] axisUp = new float[3] { matrix.m21, matrix.m01, matrix.m11 };
ITMGContext.GetInstance().GetRoom().UpdateSelfPosition(position, axisForward, axisR
```

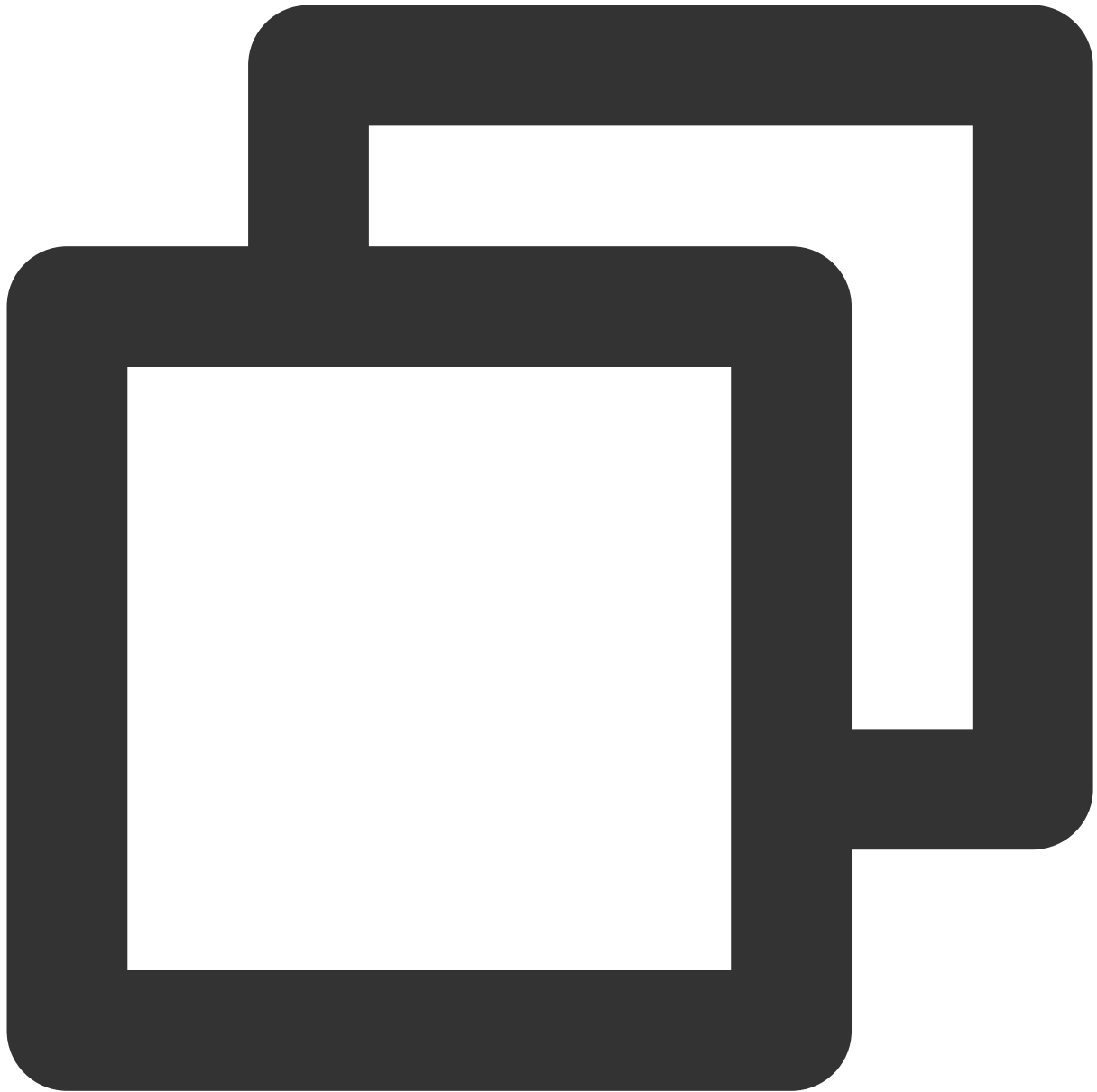
로컬 위치 API(VR 시나리오용)

이 API는 VR 장치에서 3D 위치 변경에 대한 요구 사항이 높은 시나리오에 적합합니다. 이 기능은 고급 API이며 GME 2.9.2 이상에서만 사용할 수 있습니다.

일반적인 3D 시나리오에서 사용자는 `UpdateSelfPosition` 함수를 사용하여 위치 정보를 업데이트하고 네트워크를 통해 다른 사용자에게 전송하기만 하면 됩니다. `UpdateOtherPosition`은 네트워크를 사용하지 않고 로컬에서 다른 플레이어의 위치 정보를 전달할 수도 있으므로 VR 게임 시나리오에 적합합니다.

원격으로 업데이트된 좌표와의 충돌을 피하기 위해 `UpdateOtherPosition`을 호출하면 원격 좌표가 삭제되어 다음 방 입장에 영향을 미칩니다. 따라서 로컬에서 플레이어 좌표를 업데이트하려면 모든 플레이어의 좌표를 업데이트해야 합니다.

함수 프로토타입



```
public abstract int UpdateOtherPosition(int position[3])
```

매개변수	유형	설명
position	int[]	원격 위치. 좌표 순서는 전방, 오른쪽, 위

주의사항 :

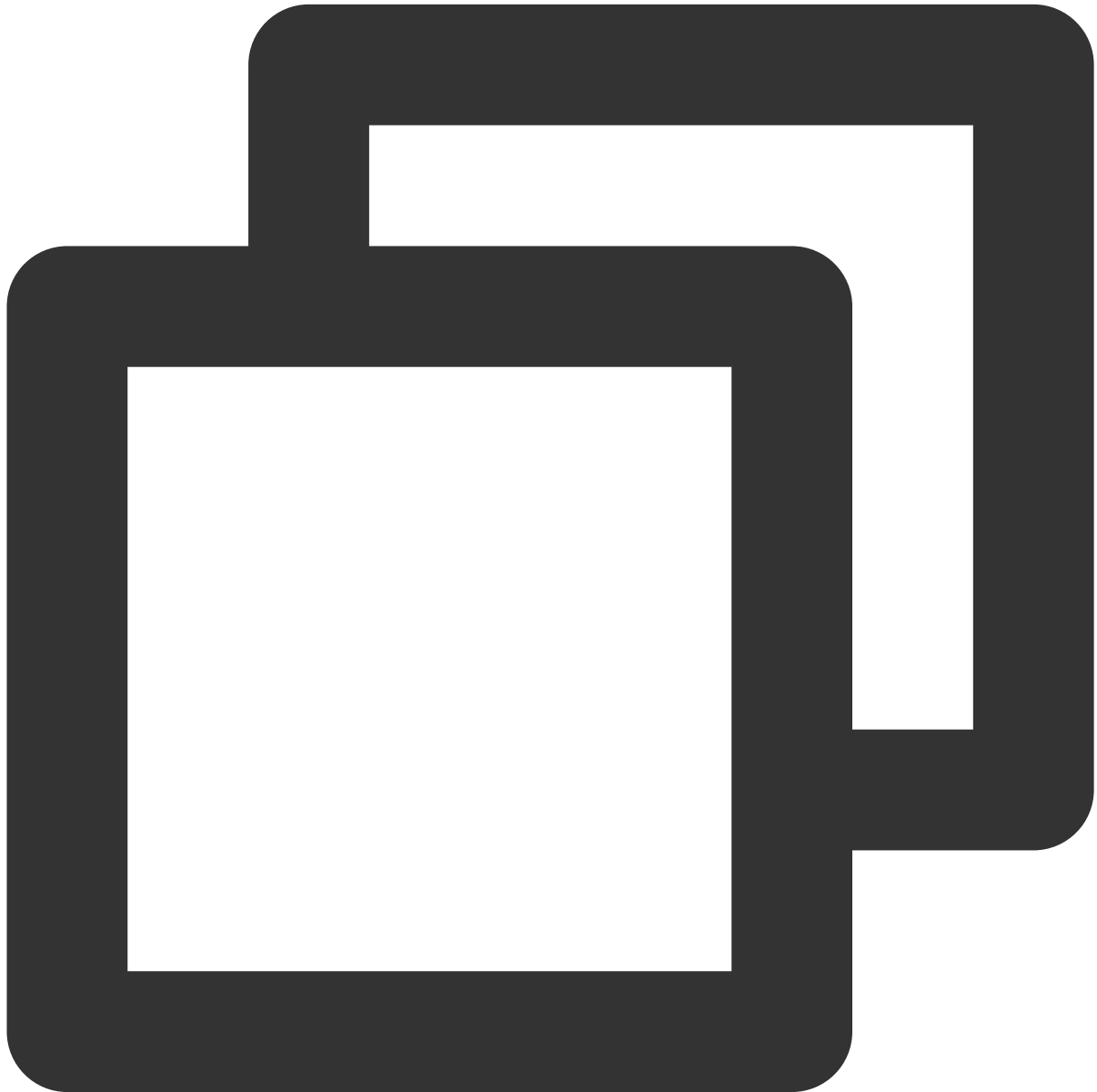
플레이어 좌표를 로컬에서 업데이트하려면 **모든 플레이어 좌표**를 트래버스하고 이 API를 통해 전달합니다.

3D 음향 효과 블랙리스트 API

주의사항 :

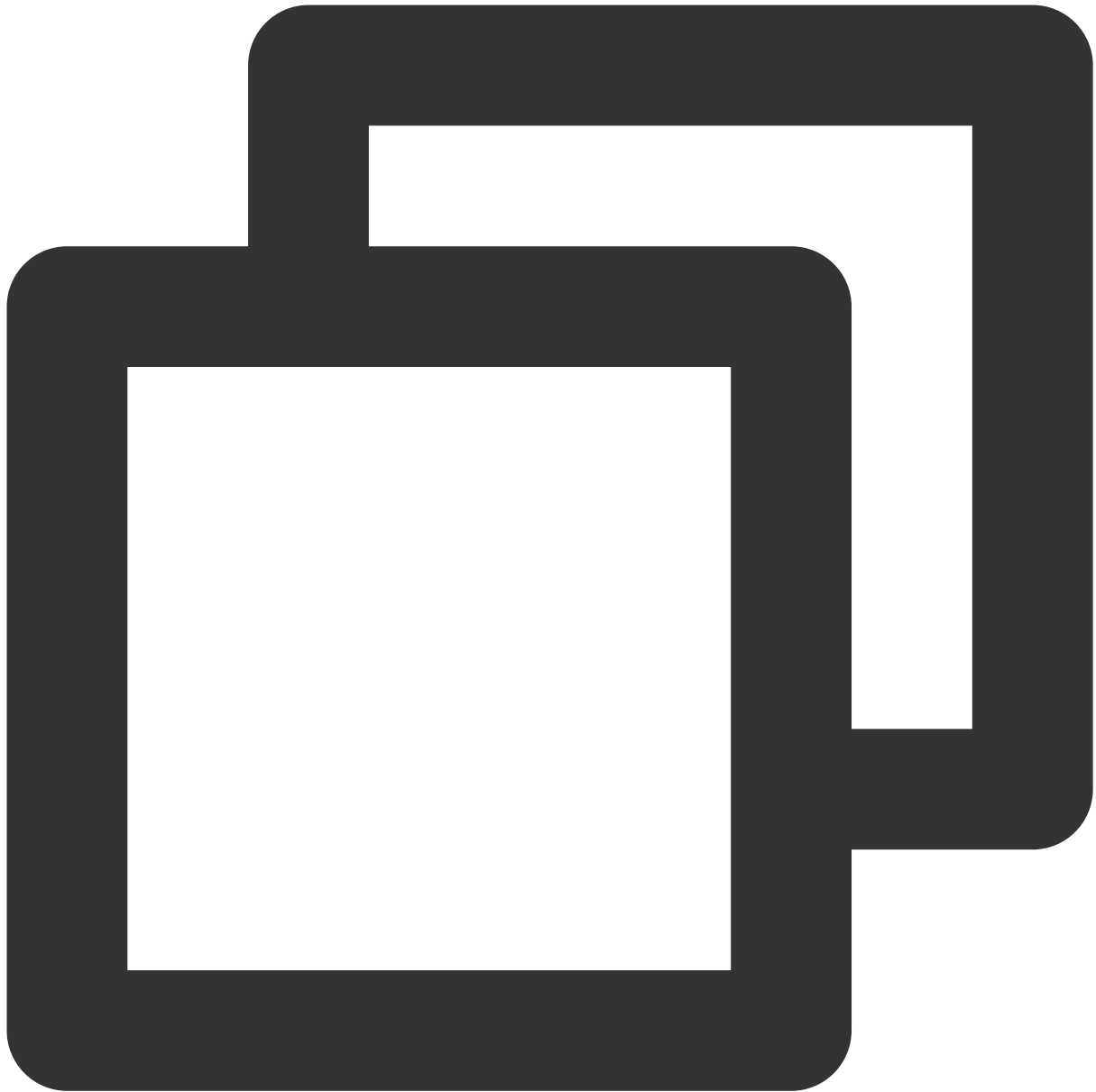
이 API는 GME SDK 2.9.3 이상에서 적용됩니다.

현재 3D 음향 효과 기능은 호출 후 방에 있는 모든 사용자에게 적용됩니다. 그러나 일부 특수 시나리오에서는 3D 음향 효과로 인해 누군가의 목소리가 약해지는 것을 원치 않을 수 있습니다. 이 경우 이 API를 호출하여 3D 음향 효과 블랙리스트에 사용자를 추가할 수 있으며 지정된 `openid`의 음성에는 더 이상 3D 음향 효과가 적용되지 않습니다.



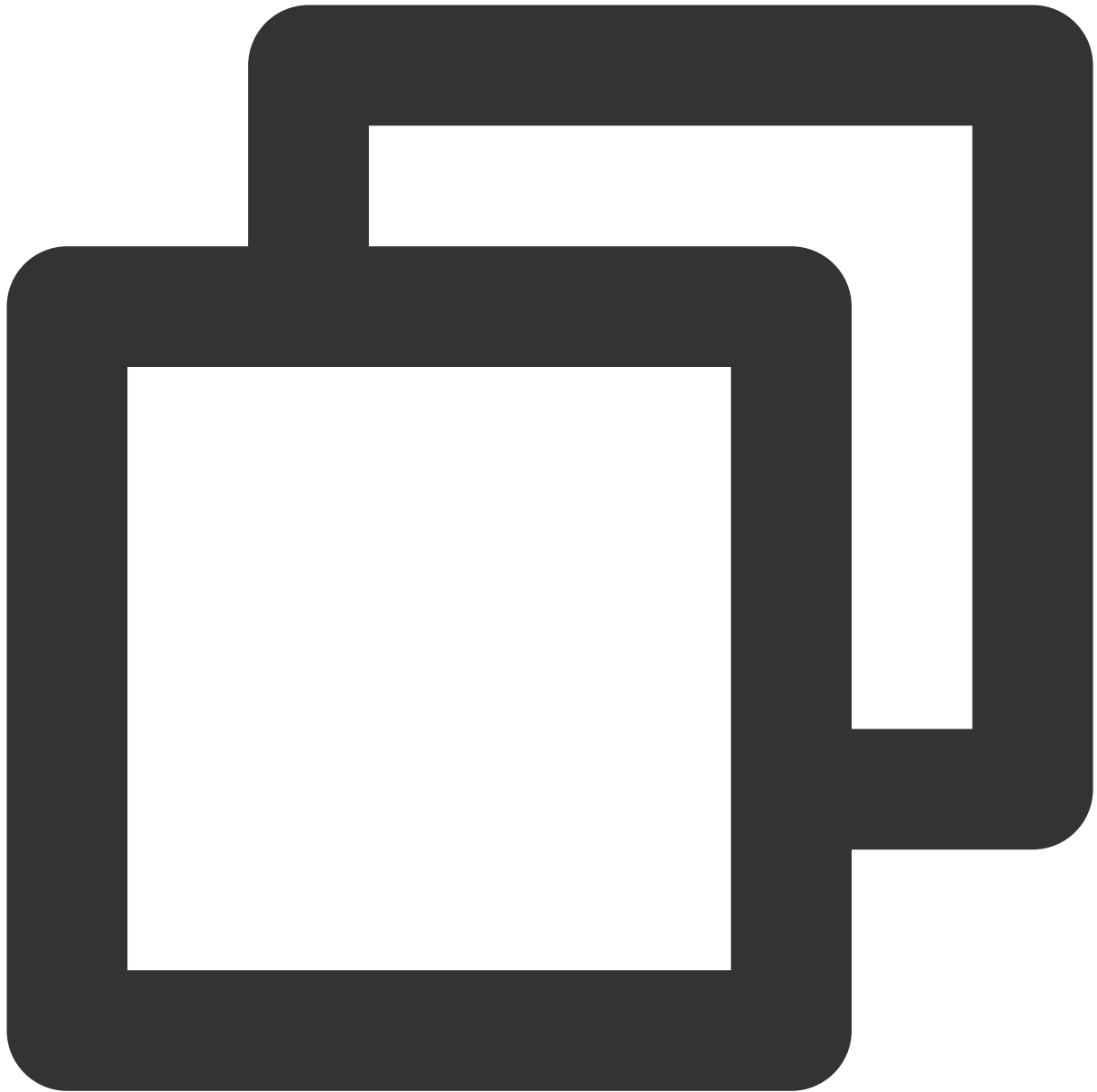
```
virtual int AddSpatializerBlacklist(const char* openId);
```

블랙리스트에서 `openid`를 제거하려면 다음 API를 호출해야 합니다.



```
virtual int RemoveSpatializerBlacklist(const char* openId);
```

블록리스트를 지우려면 다음 API를 호출해야 합니다.



```
virtual int ClearSpatializerBlacklist();
```

문제 해결

이 기능을 연결한 후 음성에 3D 음향 효과가 없으면 다음과 같이 문제를 해결할 수 있습니다.

1. 사용자가 방에 성공적으로 입장했는지, 마이크를 켜는지, 서로의 말을 들을 수 있는지 확인합니다.
2. 스테레오 헤드셋이 사용되는지 확인합니다.

3. InitSpatializer의 반환 값이 0인지 확인합니다.
4. UpdateAudioRecvRange 값이 너무 작은지 확인합니다.
5. UpdateSelfPosition API가 주기적으로 호출되는지 확인합니다.
6. [예러 코드](#)를 참고하여 문제를 해결하십시오.

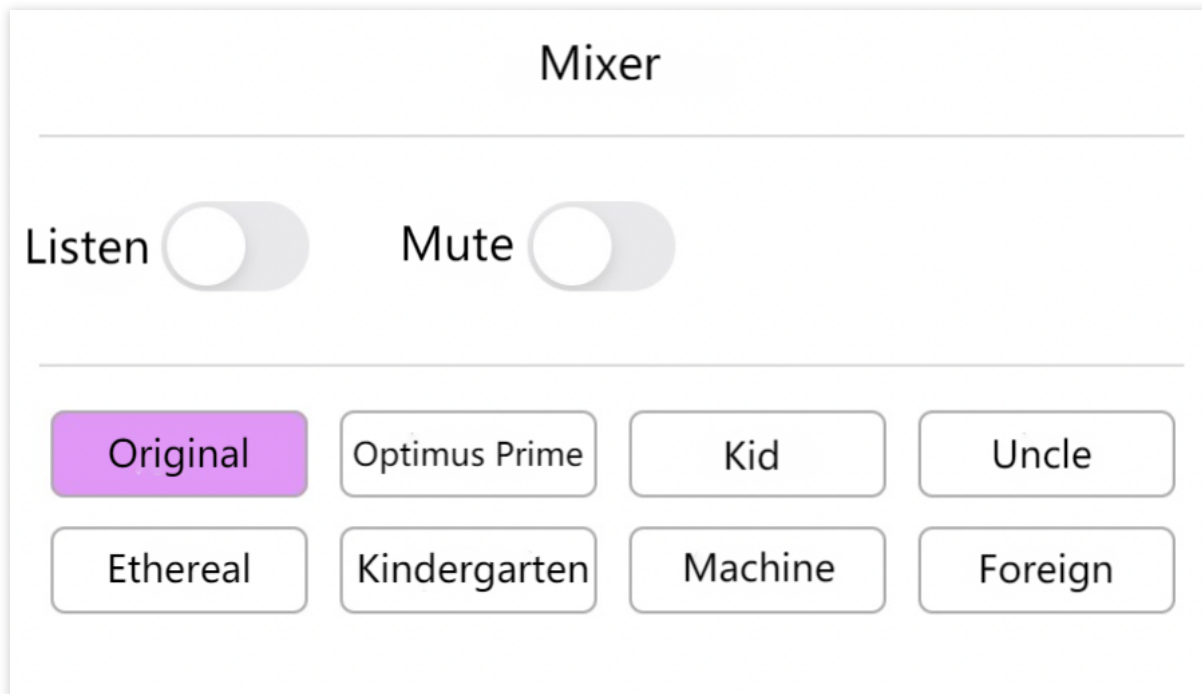
음향 효과 및 반주

음성 변조

최종 업데이트 날짜: : 2024-01-18 16:00:04

이 문서는 음성 변조 효과를 위해 GME(Game Multimedia Engine) API와 통합하고 디버깅하는 방법을 설명합니다.

시나리오



전제 조건

음성 채팅 서비스 활성화: [서비스 활성화](#)를 참고하십시오.

음성-텍스트 변환 서비스 활성화: [서비스 활성화](#)를 참고하십시오.

GME SDK 통합: 핵심 API 및 음성 채팅 API를 포함합니다. 자세한 내용은 [Quick Integration of Native SDK](#), [Quick Integration of SDK for Unity](#), [Quick Integration of SDK for Unreal Engine](#)을 참고하십시오.

GME SDK의 libgmesoundtouch 라이브러리 파일 통합: 프로젝트의 라이브러리 파일에 libgmesoundtouch가 포함되어 있는지 확인하십시오. 자세한 내용은 [Library files' corresponding features](#)를 참고하십시오.

실시간 음성 변조 통합

음성 변조 API

성공적인 방 입장 및 마이크 활성화 후, SetVoiceType API를 호출하여 음성 변조 효과를 설정합니다. API가 0을 반환하면 호출이 성공한 것이고 방에 있는 사용자에게 들리는 로컬 사운드가 음성 변조 효과를 갖습니다. 음성 변조 효과를 테스트하려면 인이어 모니터링 기능(API: EnableLoopBack)을 사용하십시오.

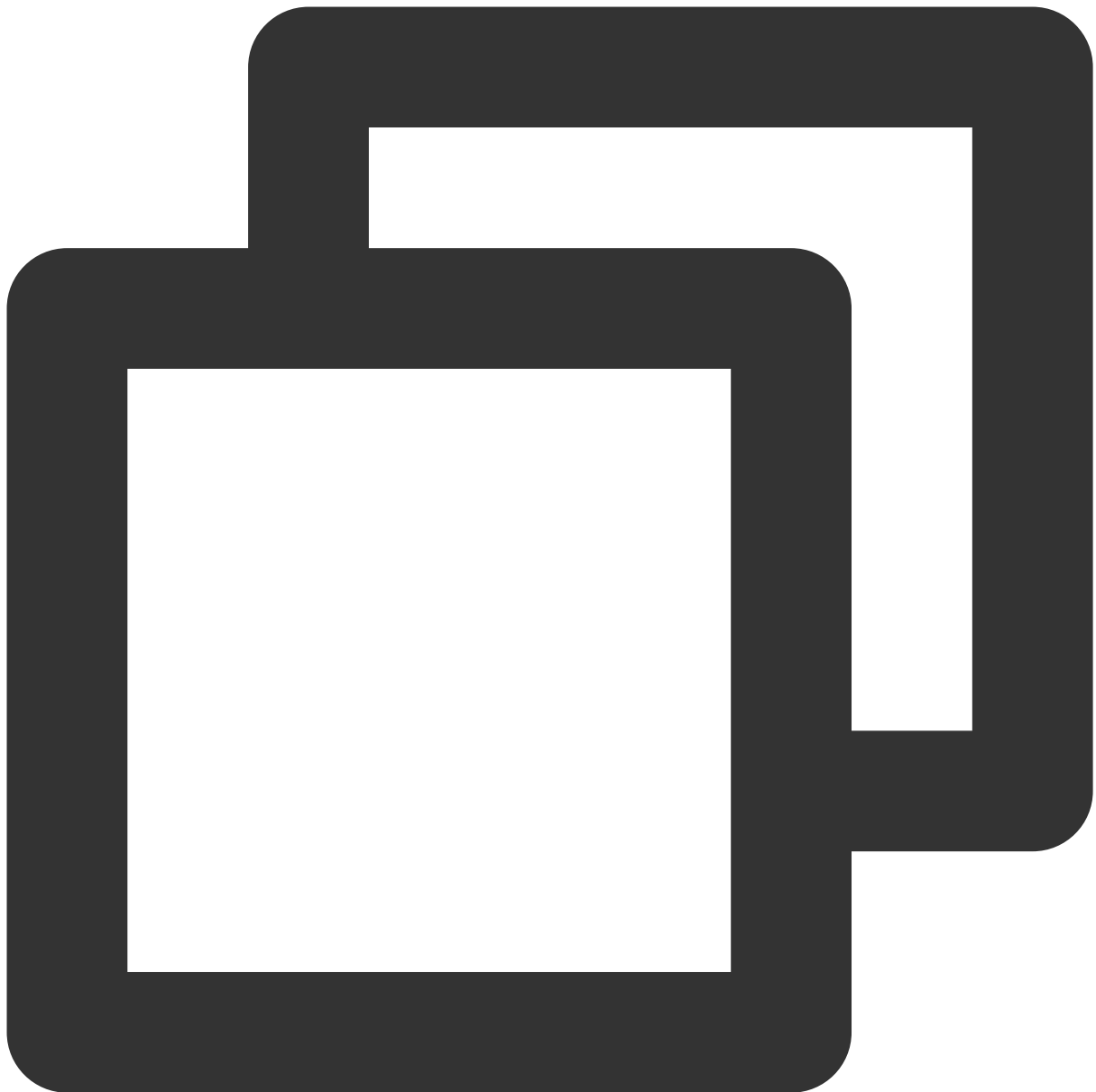
함수 프로토타입

Android

iOS

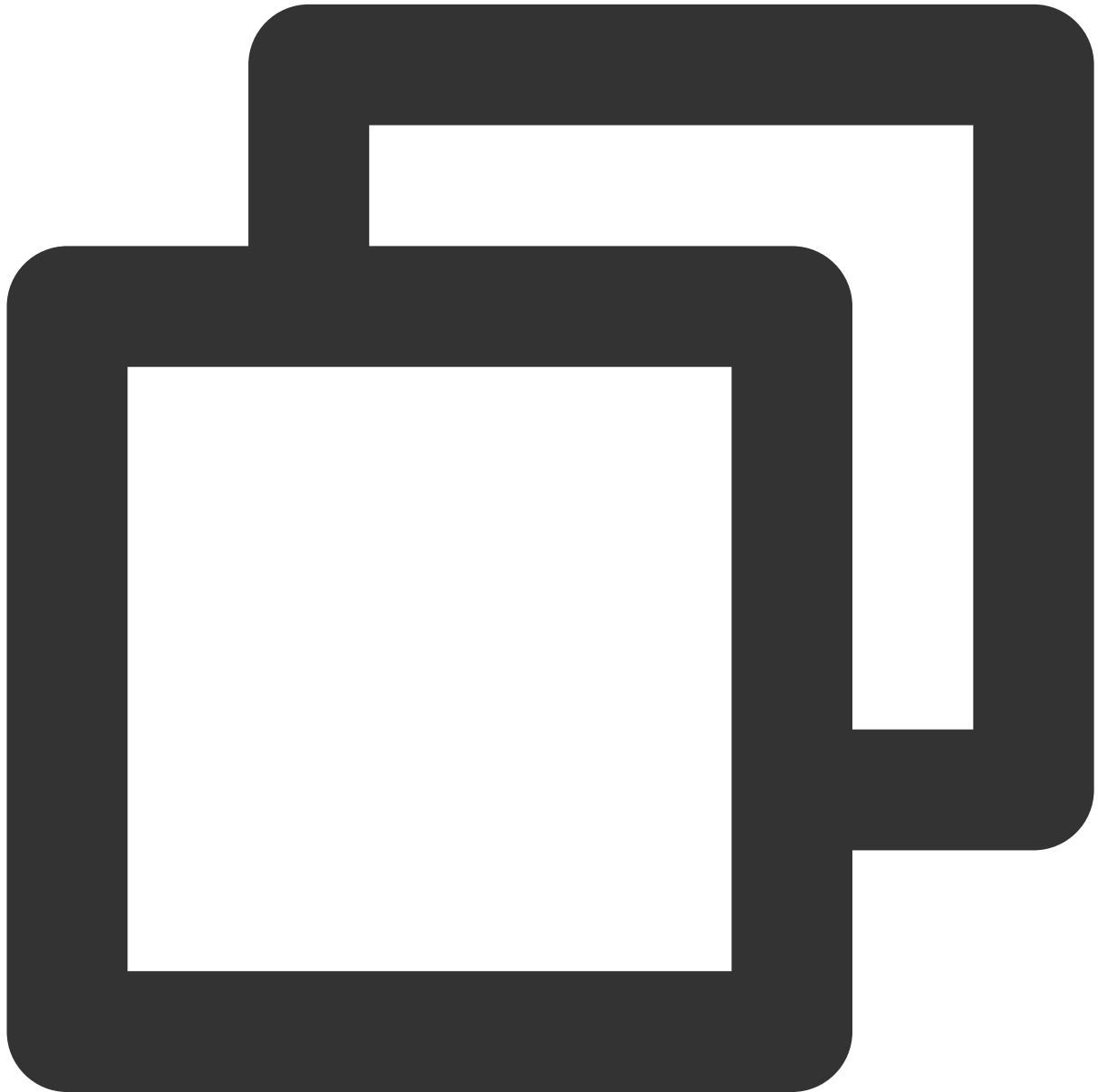
Unity

C++

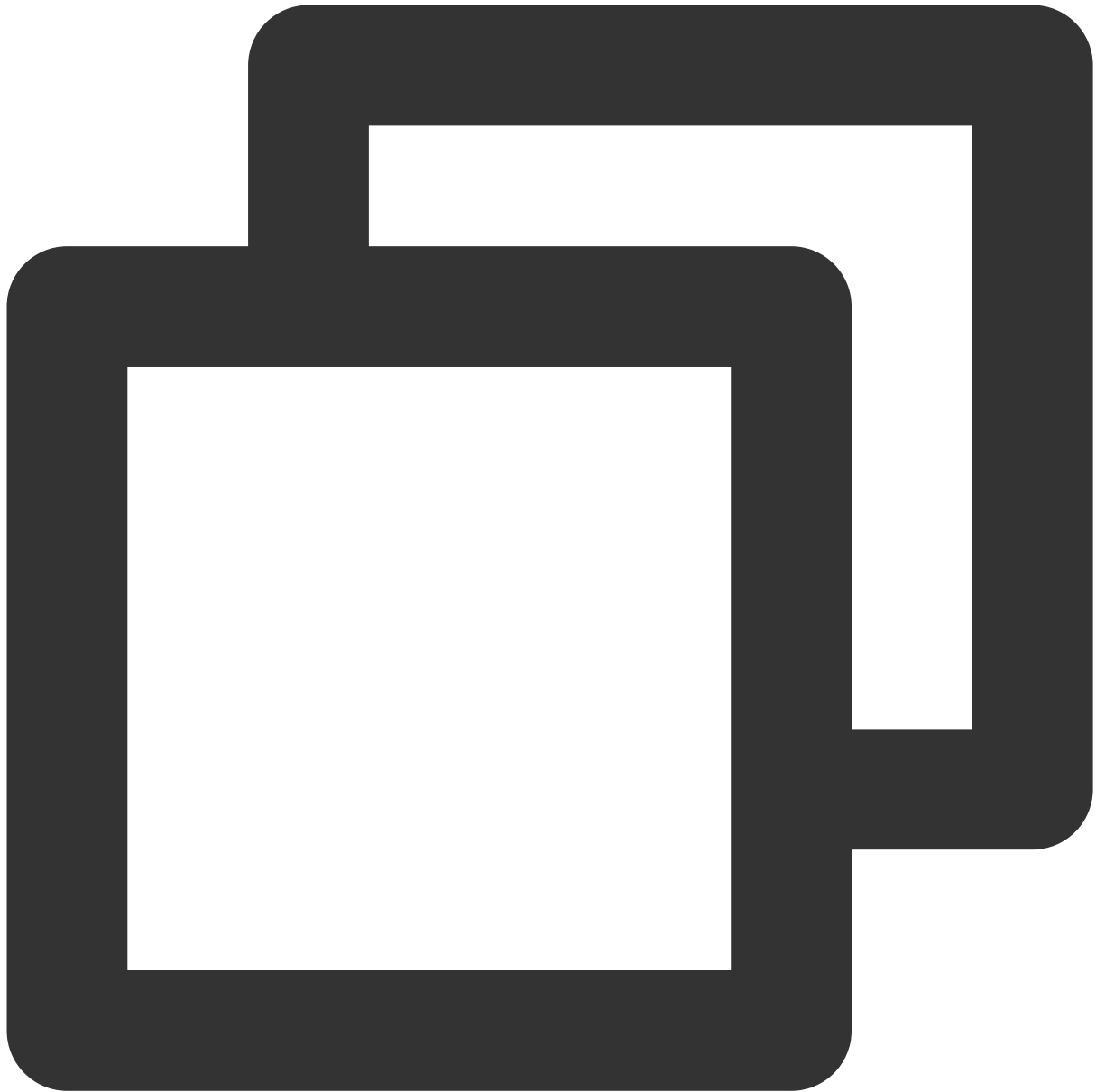


```
public static class ITMG_VoiceType {
    public static final int ITMG_VOICE_TYPE_ORIGINAL_SOUND = 0;
    public static final int ITMG_VOICE_TYPE_LOLITA = 1;
    public static final int ITMG_VOICE_TYPE_UNCLE = 2;
    public static final int ITMG_VOICE_TYPE_INTANGIBLE = 3;
    public static final int ITMG_VOICE_TYPE_DEAD_FATBOY = 4;
    public static final int ITMG_VOICE_TYPE_HEAVY_MENTAL = 5;
    public static final int ITMG_VOICE_TYPE_DIALECT = 6;
    public static final int ITMG_VOICE_TYPE_INFLUENZA = 7;
    public static final int ITMG_VOICE_TYPE_CAGED_ANIMAL = 8;
    public static final int ITMG_VOICE_TYPE_HEAVY_MACHINE = 9;
}
```

```
public static final int ITMG_VOICE_TYPE_STRONG_CURRENT = 10;
public static final int ITMG_VOICE_TYPE_KINDER_GARTEN = 11;
public static final int ITMG_VOICE_TYPE_HUANG = 12;
};
public abstract int SetVoiceType(int type);
```

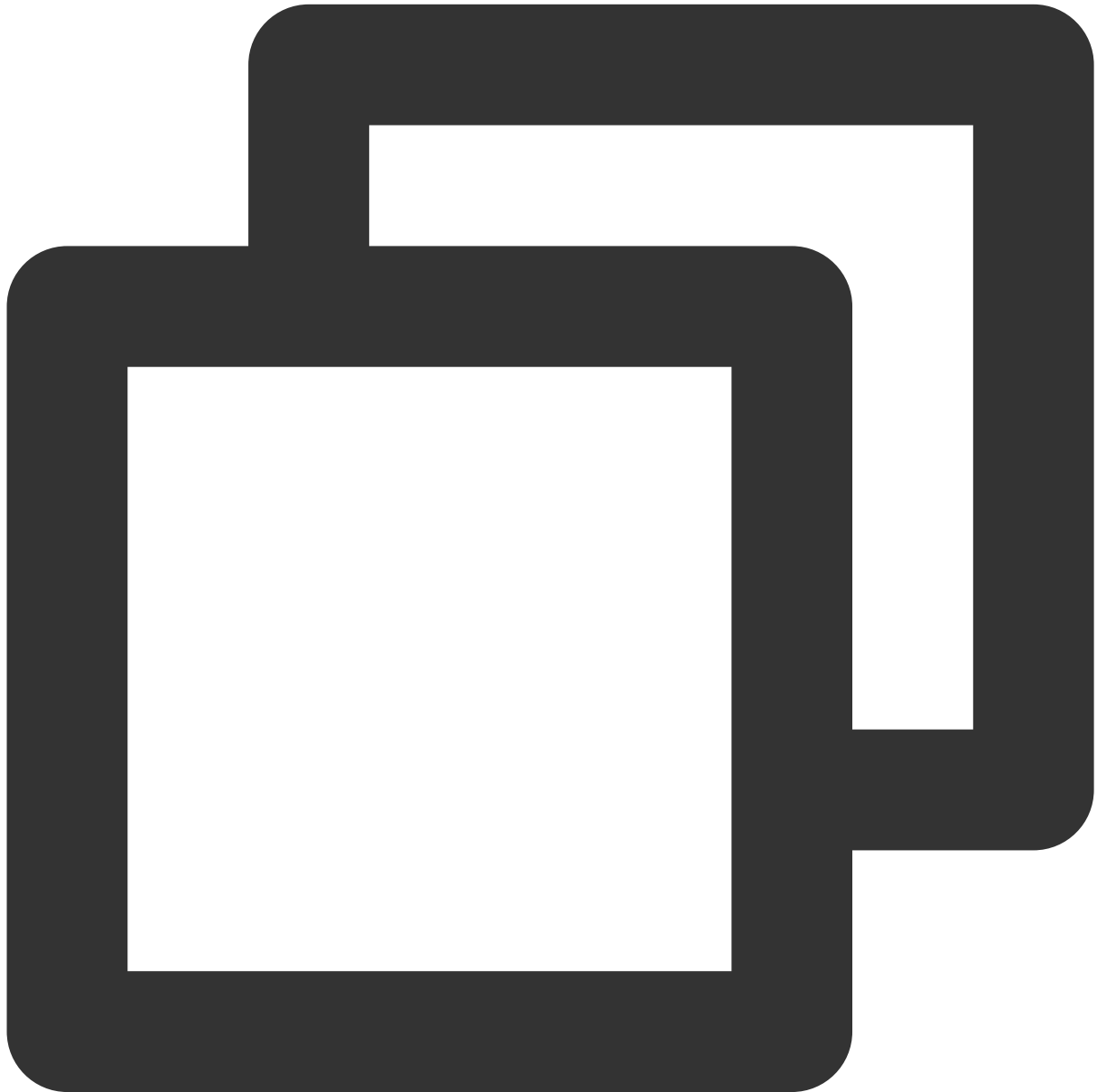


```
-(QAVResult) SetVoiceType:(ITMG_VOICE_TYPE) type
```



```
public abstract class ITMGAudioEffectCtrl{
    public static int VOICE_TYPE_ORIGINAL_SOUND = 0;
    public static int VOICE_TYPE_LOLITA = 1;
    public static int VOICE_TYPE_UNCLE = 2;
    public static int VOICE_TYPE_INTANGIBLE = 3;
    public static int VOICE_TYPE_DEAD_FATBOY = 4;
    public static int VOICE_TYPE_HEAVY_MENTAL = 5;
    public static int VOICE_TYPE_DIALECT = 6;
    public static int VOICE_TYPE_INFLUENZA = 7;
    public static int VOICE_TYPE_CAGED_ANIMAL = 8;
    public static int VOICE_TYPE_HEAVY_MACHINE = 9;
```

```
public static int VOICE_TYPE_STRONG_CURRENT = 10;  
public static int VOICE_TYPE_KINDER_GARTEN = 11;  
public static int VOICE_TYPE_HUANG = 12;  
public abstract int SetVoiceType(int voiceType);  
}
```



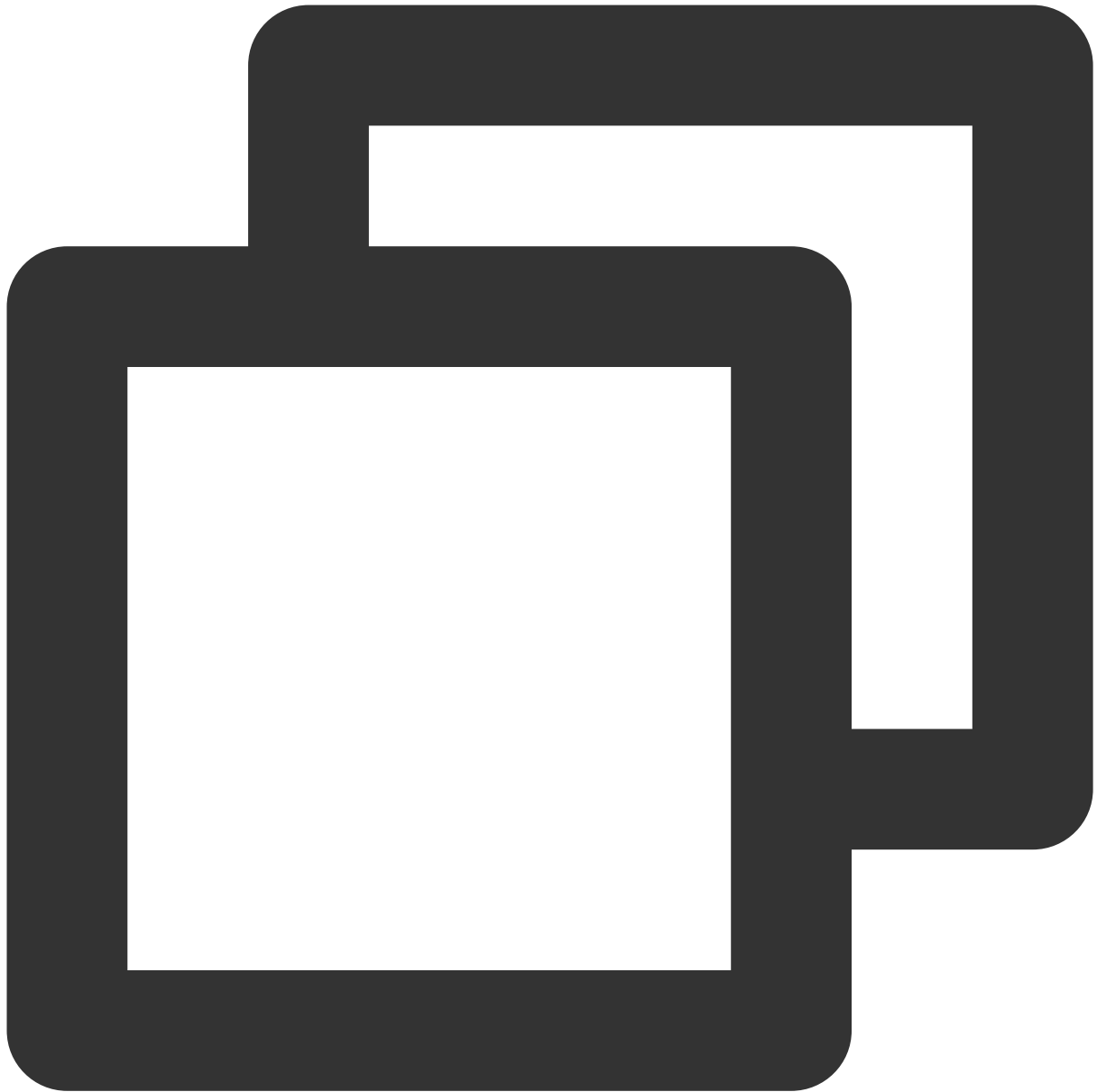
```
class ITMGAudioEffectCtrl {  
public:  
    virtual ~ITMGAudioEffectCtrl(){};  
    virtual int SetVoiceType(ITMG_VOICE_TYPE voiceType) = 0;
```

}

매개변수	유형	설명
type	int	로컬 음성 변조 효과의 종류를 나타냅니다

유형 매개변수	값	설명
ITMG_VOICE_TYPE_ORIGINAL_SOUND	0	오리지널
ITMG_VOICE_TYPE_LOLITA	1	로리타
ITMG_VOICE_TYPE_UNCLE	2	삼촌
ITMG_VOICE_TYPE_INTANGIBLE	3	오묘한
ITMG_VOICE_TYPE_DEAD_FATBOY	4	통통이
ITMG_VOICE_TYPE_HEAVY_MENTA	5	헤비메탈
ITMG_VOICE_TYPE_DIALECT	6	외국인
ITMG_VOICE_TYPE_INFLUENZA	7	감기
ITMG_VOICE_TYPE_CAGED_ANIMAL	8	동물
ITMG_VOICE_TYPE_HEAVY_MACHINE	9	기계
ITMG_VOICE_TYPE_STRONG_CURRENT	10	강한 전류
ITMG_VOICE_TYPE_KINDER_GARTEN	11	유치원
ITMG_VOICE_TYPE_HUANG	12	미니언

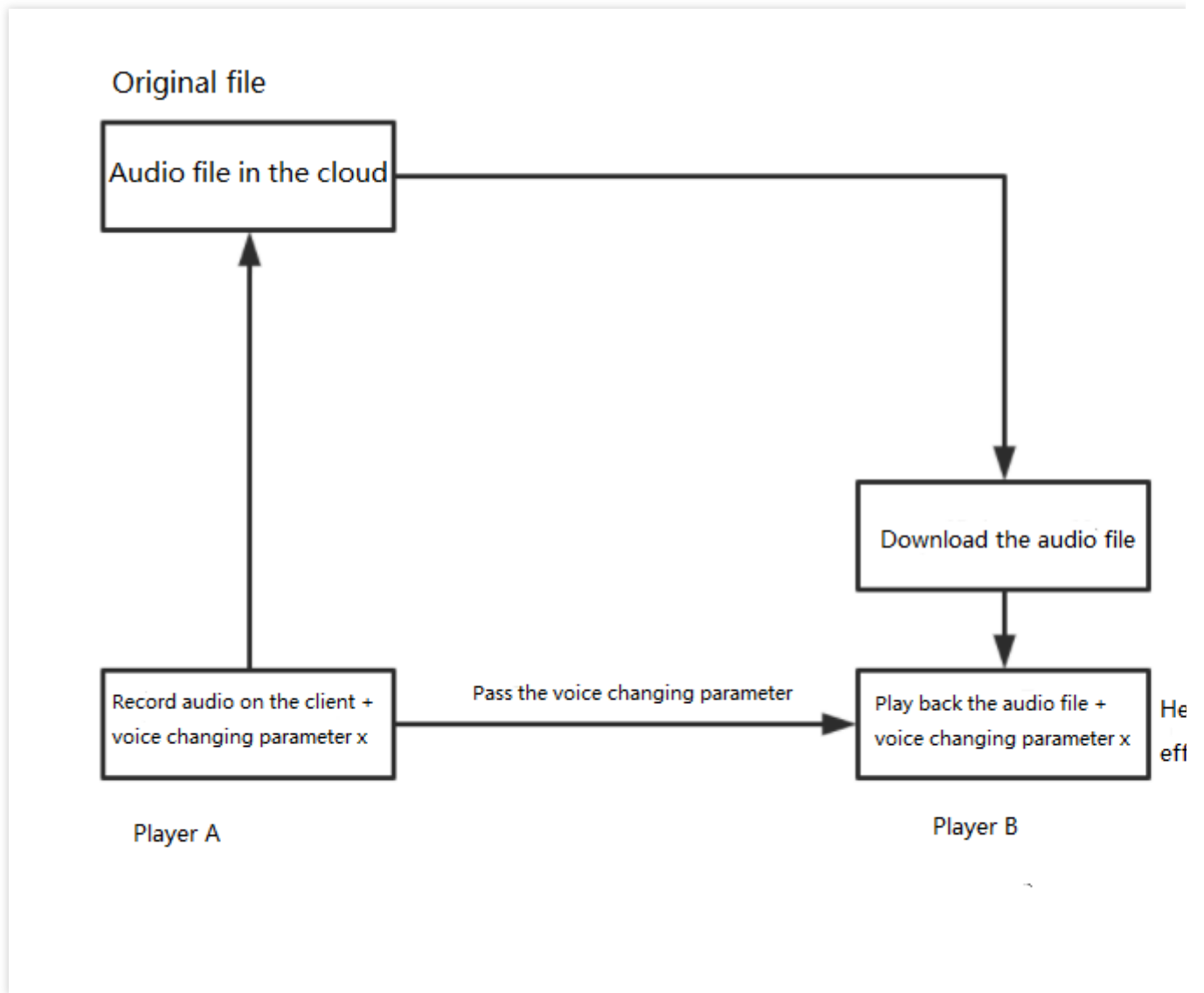
예시 코드



```
ITMGContextGetInstance () ->GetAudioEffectCtrl () ->setVoiceType (0);
```

음성 메시지에 대한 음성 변조 통합

음성 메시지에 대한 음성 변조 프로세스



음성 변조 효과는 재생 중에만 반영되므로 음성 변조는 원본 음성 메시지에는 영향을 미치지 않습니다.

음성 메시지 재생

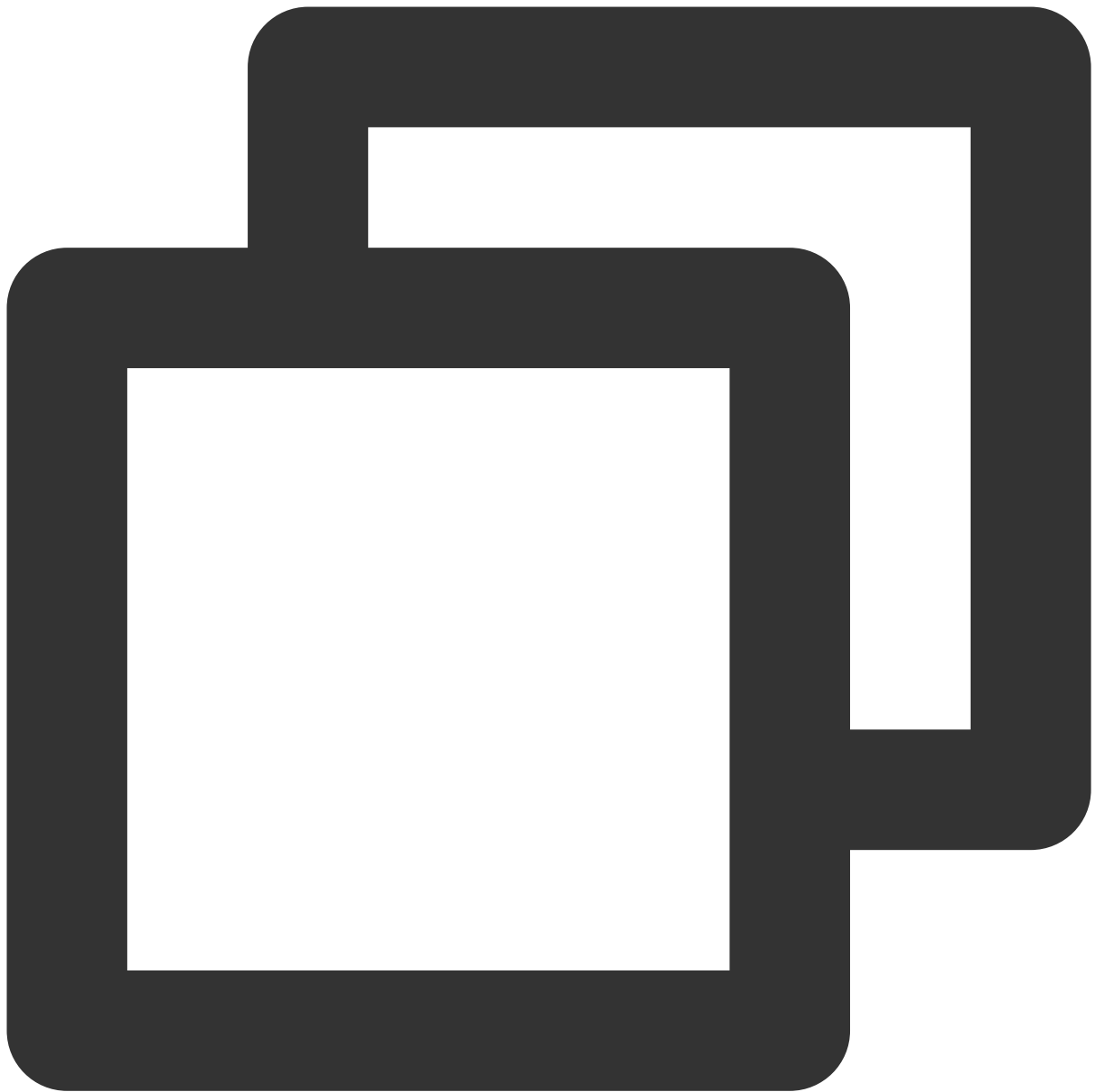
음성 메시지 재생 API 호출 시 음성 변조 매개변수를 추가합니다.

Android

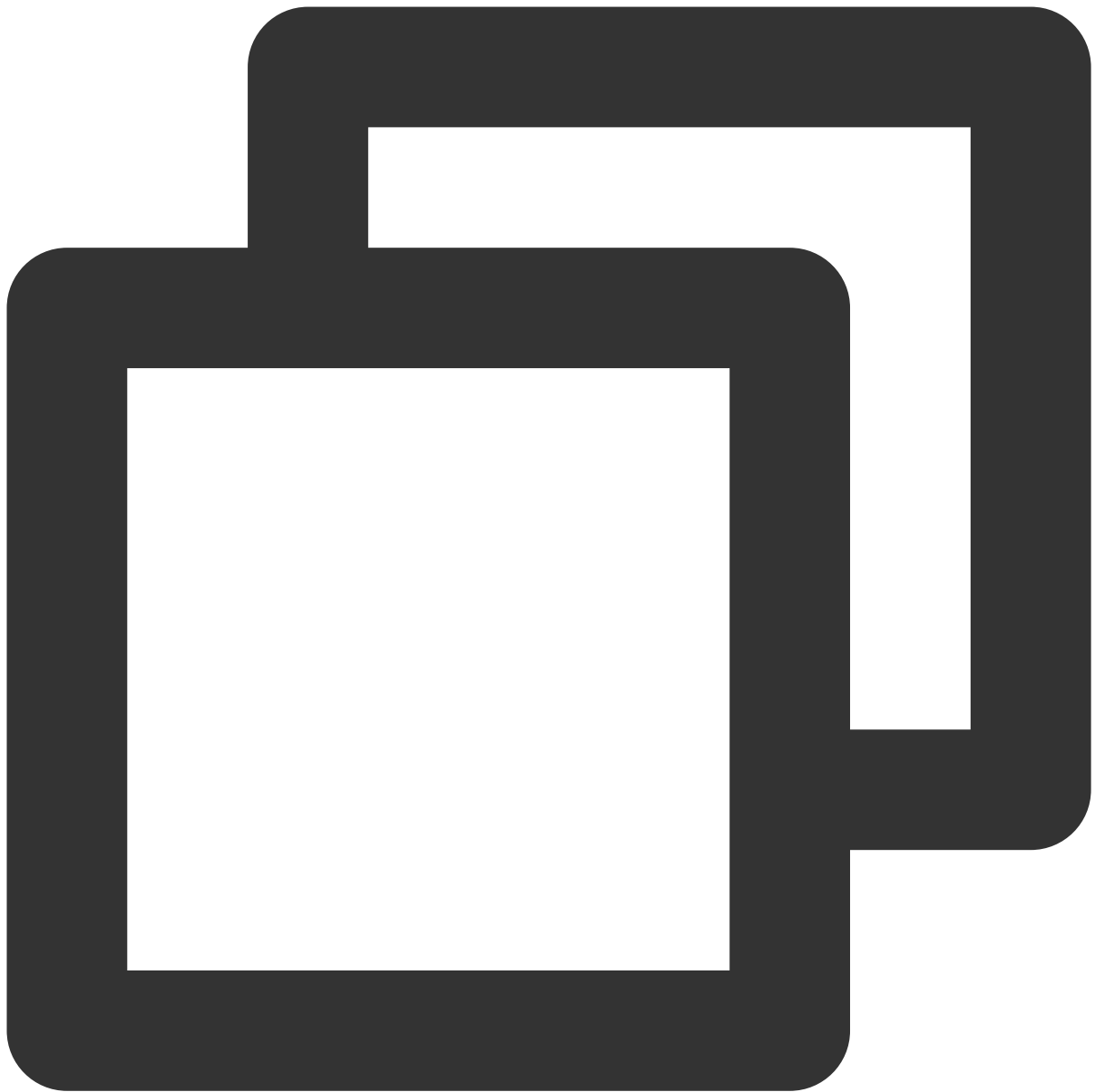
iOS

Unity

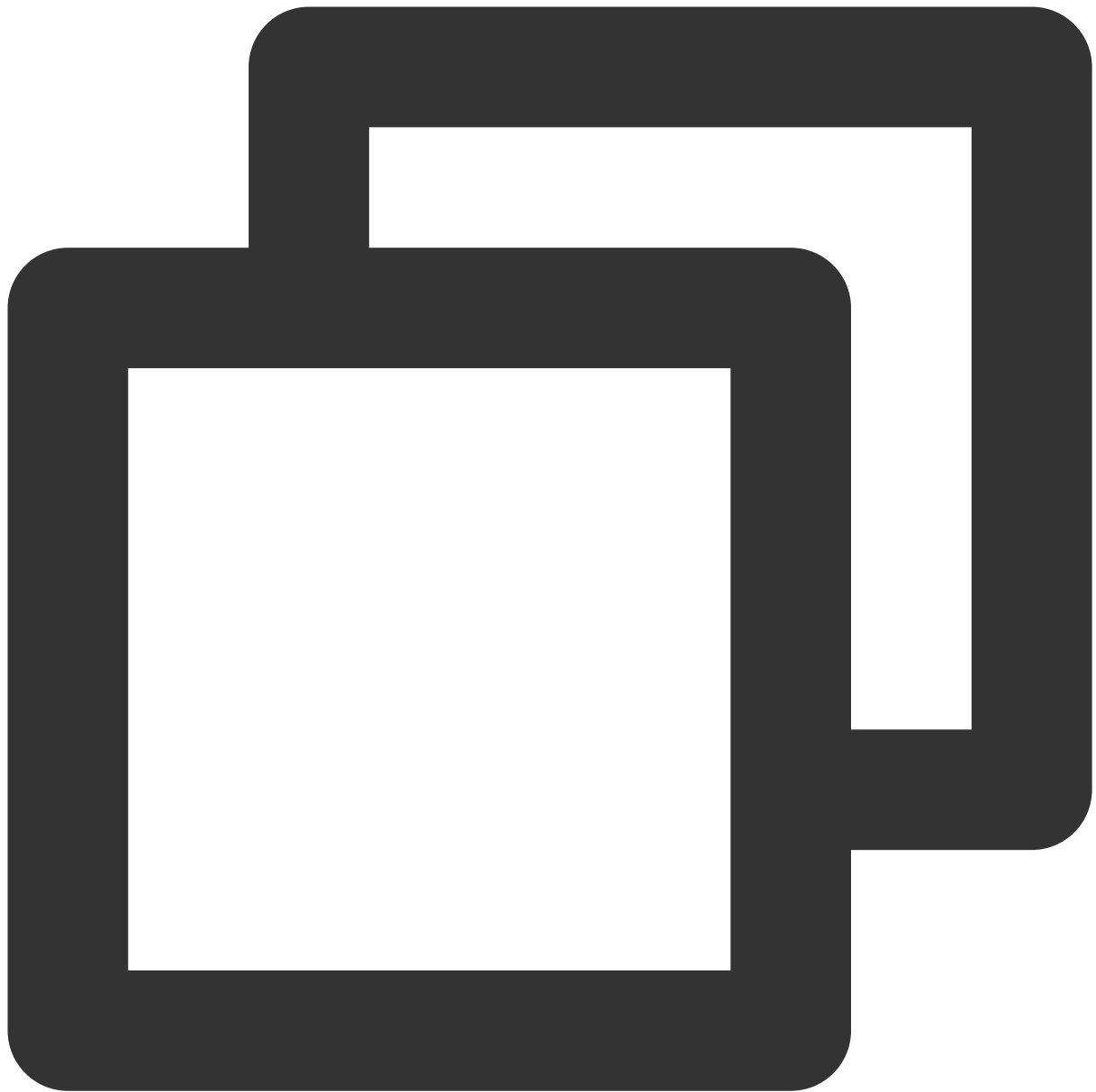
C++



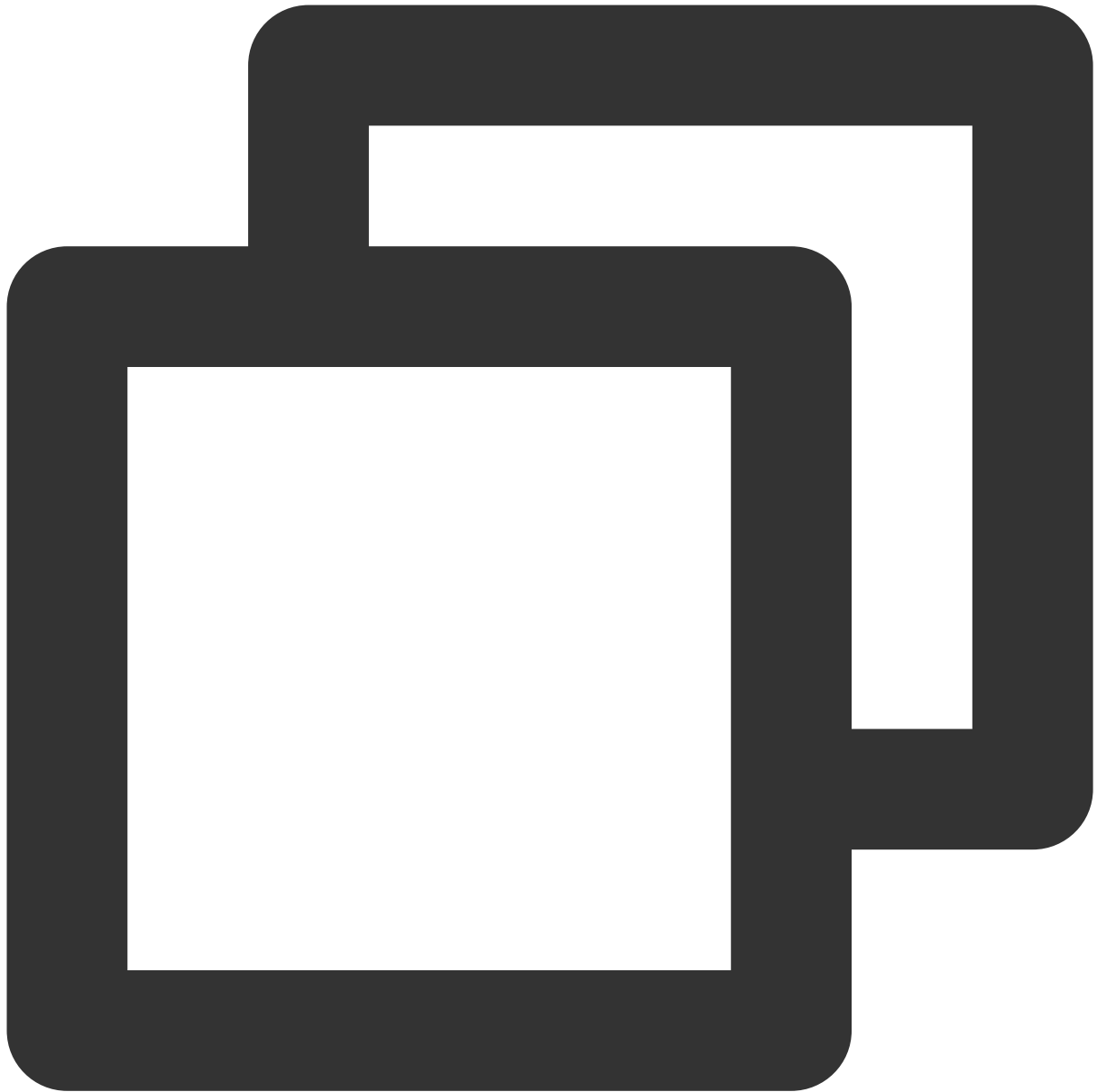
```
public abstract int PlayRecordedFile(String filePath,int voicetype);
```

```
-(int)PlayRecordedFile:(NSString*)filePath VoiceType:(ITMG_VOICE_TYPE) type
```



```
ITMGPTT PlayRecordedFile(string filePath,int voiceType);
```



```
public abstract int PlayRecordedFile(string filePath,int voiceType);
```

매개변수	유형	설명
filePath	string	로컬 오디오 파일 경로
voicetype	int	음성 변조 유형

오류 코드

오류 코드 값	원인	솔루션 제안
20485	재생이 시작되지 않습니다	파일의 존재와 파일 경로의 유효성을 확인하십시오

실시간 음성 반주

최종 업데이트 날짜: : 2024-01-18 16:00:04

개발자가 Tencent Cloud Game Multimedia Engine(GME) 제품 API를 원활히 테스트하고 액세스하는것을 지원드리고자, 게임 멀티미디어 엔진 실시간 음성 반주 관련 액세스 기술 파일을 안내드립니다.

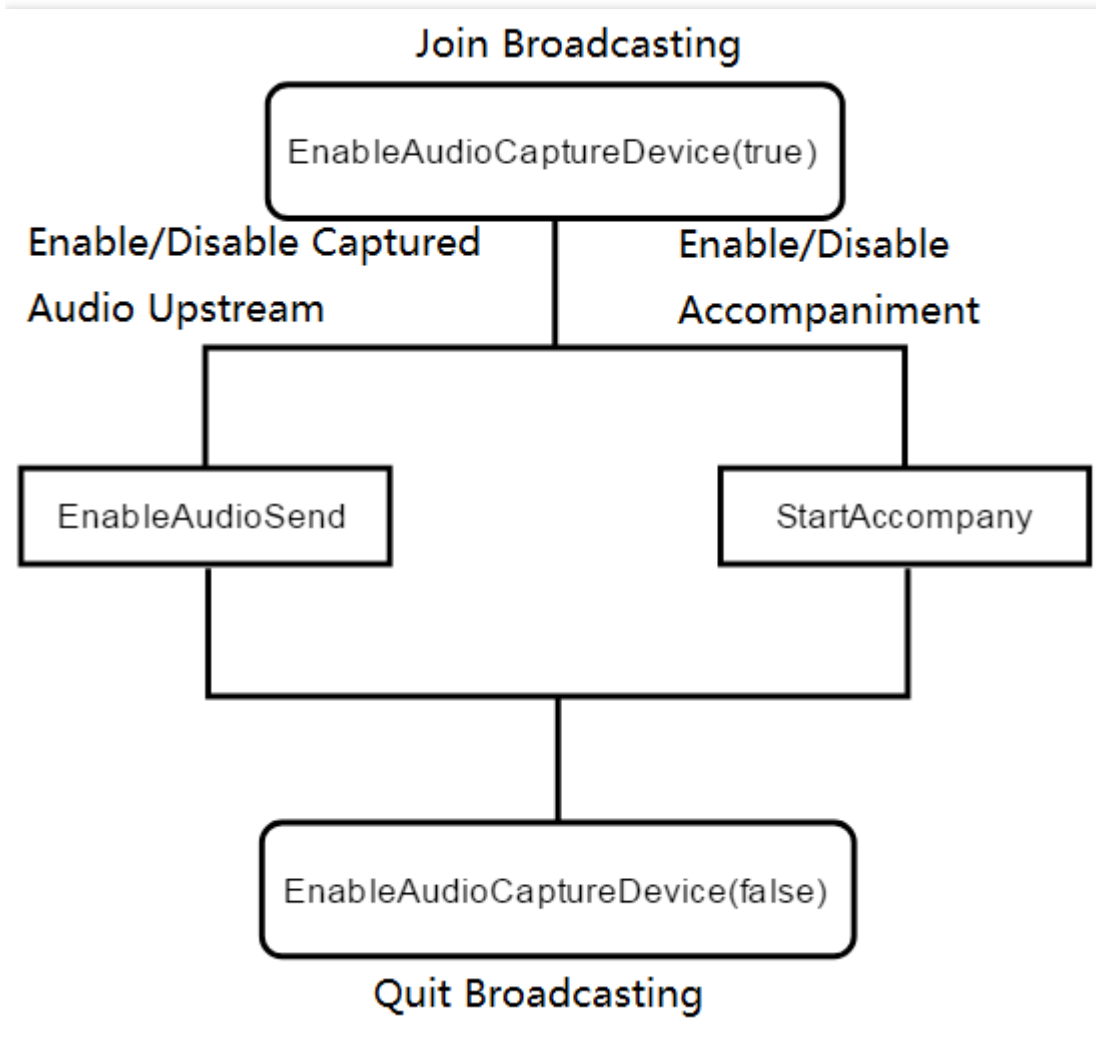
실시간 음성 반주 관련 API

API	API 의미
StartAccompany	반주 플레이 시작.
StopAccompany	반주 플레이 정지.
IsAccompanyPlayEnd	반주 플레이 완료 여부.
PauseAccompany	반주 플레이 일시정지.
ResumeAccompany	반주 리플레이.
SetAccompanyVolume	반주 사운드 설정.
GetAccompanyVolume	반주 플레이 사운드 획득.
SetAccompanyFileCurrentPlayedTimeByMs	플레이 진행 상황 설정.

실시간 음성 반주를 원하시면 GME SDK를 액세스해야 하며, 실시간 음성 통화 시에만 실시간 음성 반주를 사용할 수 있습니다.

프로세스 이미지

소셜타입의 앱 호출 프로세스는 하기 이미지를 참조해주세요.



반주 플레이 시작

StartAccompany API를 호출하여 반주 플레이를 시작합니다. m4a, wav, mp3 등 3종 포맷을 지원합니다. 이 API를 호출하면 사운드는 리셋됩니다.

함수 프로토타입

```
ITMGAudioEffectCtrl virtual int StartAccompany(const char* filePath, bool loopBack, int loopCount, int msTime)
```

파라미터	타입	의미
filePath	char*	반주를 플레이하는 경로.
loopBack	bool	오디오 믹싱 발송 여부, 일반적으로 true로 설정하면 타인도 반주를 들을 수 있습니다.
loopCount	int	리사이클 횟수, 수치가 -1일 경우 무제한 리사이클을 의미합니다

파라미터	타입	의미
msTime	int	지연시간.

예시 코드

```
//Windows 코드
ITMGContextGetInstance()->GetAudioEffectCtrl()->StartAccompany(filePath,true,-1,0);
//Android 코드
ITMGContext.GetInstance(this).GetAudioEffectCtrl().StartAccompany(filePath,true,loopCount,0);
//iOS 코드
[[[ITMGContext GetInstance] GetAudioEffectCtrl] StartAccompany:path loopBack:isLoopBack loopCount:loopCount msTime:0];
```

반주 플레이 콜백

반주 플레이를 완료하면 콜백 함수는 OnEvent를 호출합니다. 이벤트 메시지는 ITMG_MAIN_EVENT_TYPE_ACCOMPANY_FINISH이며, OnEvent함수에서 이벤트 메시지를 판단합니다. 전송된 파라미터 data에 result와 file_path 등 2종 메시지를 포함합니다.

예시 코드

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char* data){
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
            //프로세싱
            break;
        }
        ...
        case ITMG_MAIN_EVENT_TYPE_ACCOMPANY_FINISH:
        {
            //프로세싱
            break;
        }
    }
}
```

반주 플레이 정지

StopAccompany API를 호출하여 반주 플레이를 정지합니다.

함수 프로토타입

```
ITMGAudioEffectCtrl virtual int StopAccompany(int duckerTime)
```

파라미터	타입	의미
duckerTime	int	페이드아웃 시간.

예시 코드

```
ITMGContextGetInstance()->GetAudioEffectCtrl()->StopAccompany(0);
```

반주 플레이 완료 여부

플레이를 완료하면 리턴값은 true이며, 플레이를 완료하지 않았다면 리턴값은 false입니다.

함수 프로토타입

```
ITMGAudioEffectCtrl virtual bool IsAccompanyPlayEnd()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioEffectCtrl()->IsAccompanyPlayEnd();
```

반주 플레이 일시정지

PauseAccompany API를 호출하여 반주 플레이를 일시정지합니다.

함수 프로토타입

```
ITMGAudioEffectCtrl virtual int PauseAccompany()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioEffectCtrl()->PauseAccompany();
```

반주 플레이 복구

ResumeAccompany API를 사용하여 반주 플레이를 복구합니다.

함수 프로토타입

```
ITMGAudioEffectCtrl virtual int ResumeAccompany()
```


예시 코드

```
ITMGContextGetInstance () ->GetAudioEffectCtrl () ->ResumeAccompany ();
```

스스로 반주를 들을 수 있는지 설정합니다.

이 API는 자기 스스로 반주를 들을 수 있는 여부를 설정합니다.

함수 프로토타입

```
ITMGAudioEffectCtrl virtual int EnableAccompanyPlay (bool enable)
```

파라미터	타입	의미
enable	bool	들을 수 있는지 여부.

예시 코드

```
ITMGContextGetInstance () ->GetAudioEffectCtrl () ->EnableAccompanyPlay (false);
```

타인이 반주를 들을 수 있는지 설정합니다.

타인도 반주를 들을 수 있는지 여부를 설정합니다.

함수 프로토타입

```
ITMGAudioEffectCtrl virtual int EnableAccompanyLoopBack (bool enable)
```

파라미터	타입	의미
enable	bool	들을 수 있는지 여부.

예시 코드

```
ITMGContextGetInstance () ->GetAudioEffectCtrl () ->EnableAccompanyLoopBack (false);
```

반주 사운드 설정

SetAccompanyVolume API를 호출하여 반주 사운드를 설정합니다. 기본값은 100이며, 수치가 100보다 클 경우 사운드 증가, 수치가 100보다 작을 경우 사운드 감소, 값범위는 0-200입니다.

함수 프로토타입

```
ITMGAudioEffectCtrl virtual int SetAccompanyVolume(int vol)
```

파라미터	타입	의미
vol	int	사운드 수치.

예시 코드

```
int vol=100;
ITMGContextGetInstance()->GetAudioEffectCtrl()->SetAccompanyVolume(vol);
```

반주 플레이 사운드 획득

GetAccompanyVolume API는 반주 사운드를 읽어옵니다.

함수 프로토타입

```
ITMGAudioEffectCtrl virtual int GetAccompanyVolume()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioEffectCtrl()->GetAccompanyVolume();
```

반주 플레이 진행 상황 획득

다음 2종 API는 반주 플레이 진행 상황을 읽어옵니다. 주의 사항: $Current / Total$ = 기존 리사이클 횟수, $Current \% Total$ = 기존 리사이클 플레이 위치.

함수 프로토타입

```
ITMGAudioEffectCtrl virtual int GetAccompanyFileTotalTimeByMs()
ITMGAudioEffectCtrl virtual int GetAccompanyFileCurrentPlayedTimeByMs()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioEffectCtrl()->GetAccompanyFileTotalTimeByMs();
ITMGContextGetInstance()->GetAudioEffectCtrl()->GetAccompanyFileCurrentPlayedTimeByMs();
```

플레이 진행 상황 설정

SetAccompanyFileCurrentPlayedTimeByMs API는 플레이 진행 상황을 설정할 수 있습니다

함수 프로토타입

```
ITMGAudioEffectCtrl virtual int SetAccompanyFileCurrentPlayedTimeByMs(unsigned int time)
```

파라미터	타입	의미
time	int	플레이 진행 상황, 단위는 밀리초임.

예시 코드

```
ITMGContextGetInstance()->GetAudioEffectCtrl()->SetAccompanyFileCurrentPlayedTimeByMs(time);
```

에러코드 리스트

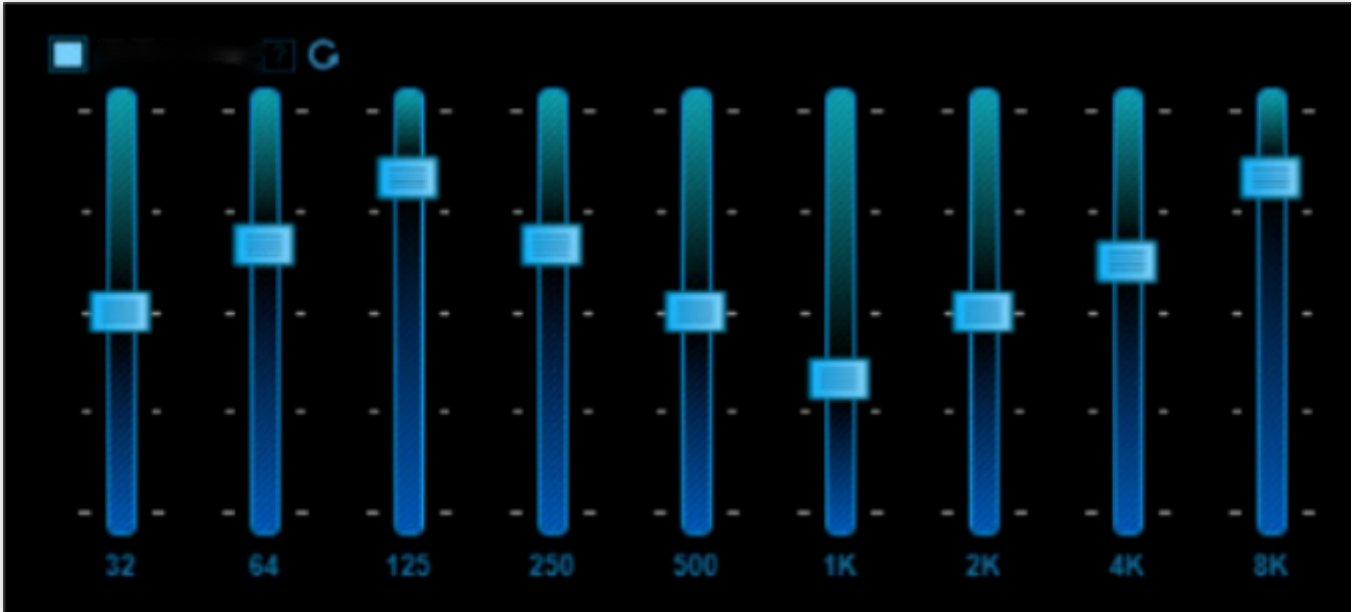
에러코드 명칭	에러코드값	에러코드 의미	솔루션
QAV_ERR_ACC_OPENFILE_FAILED	4001	파일 오픈 실패	파일 경로와 파일 존재 여부를 체크합니다. 파일 액세스 권한이 있는지 체크합니다.
QAV_ERR_ACC_FILE_FORAMT_NOTSUPPORT	4002	지원하지 않는 파일 포맷입니다	파일 포맷이 정확한지 체크합니다.
QAV_ERR_ACC_DECODER_FAILED	4003	디코딩 실패	파일포맷이 정확한지 체크합니다.
QAV_ERR_ACC_BAD_PARAM	4004	파라미터 에러	코드에 입력한 파라미터가 정확한지 체크합니다.
QAV_ERR_ACC_MEMORY_ALLOC_FAILED	4005	메모리 할당 실패	시스템 리소스를 소진하였습니다. 혹 이 에러코드가 계속 존재하면 개발자에게 문의해주세요.

에러코드 명칭	에러코드값	에러코드 의미	솔루션
QAV_ERR_ACC_CREATE_THREAD_FAILED	4006	스레드 구축 실패	시스템 리소스를 소진하였습니다. 혹 이 에러코드가 계속 존재하면 개발자에게 문의해주세요.
QAV_ERR_ACC_STATE_ILLEGAL	4007	불법 상태입니다	어떤 상태가 아니며, 이 상태에 있을 시에만 허용되는 API를 호출해야 합니다. 그러지 않을 경우 이 에러가 생성합니다.

실시간 사운드 이퀄라이저

최종 업데이트 날짜: : 2024-01-18 16:00:04

본문은 개발자가 쉽게 통합하고 디버깅할 수 있도록 음성 채팅의 이퀄라이저용 Game Multimedia Engine(GME) API에 대해 설명합니다.



시나리오

GME 이퀄라이저 기능은 GME SDK에서 실시간으로 캡처한 오디오 스트림의 이퀄라이저를 조정할 수 있습니다. 이 기능은 온라인 노래방 시나리오에 적용할 수 있습니다. 플레이어가 노래를 시작한 후 **GME SDK의 이퀄라이저 API**를 호출하여 플레이어의 실시간 오디오 스트림을 조정합니다.

전제 조건

음성 채팅 서비스 활성화: [서비스 활성화](#)를 참고하십시오.

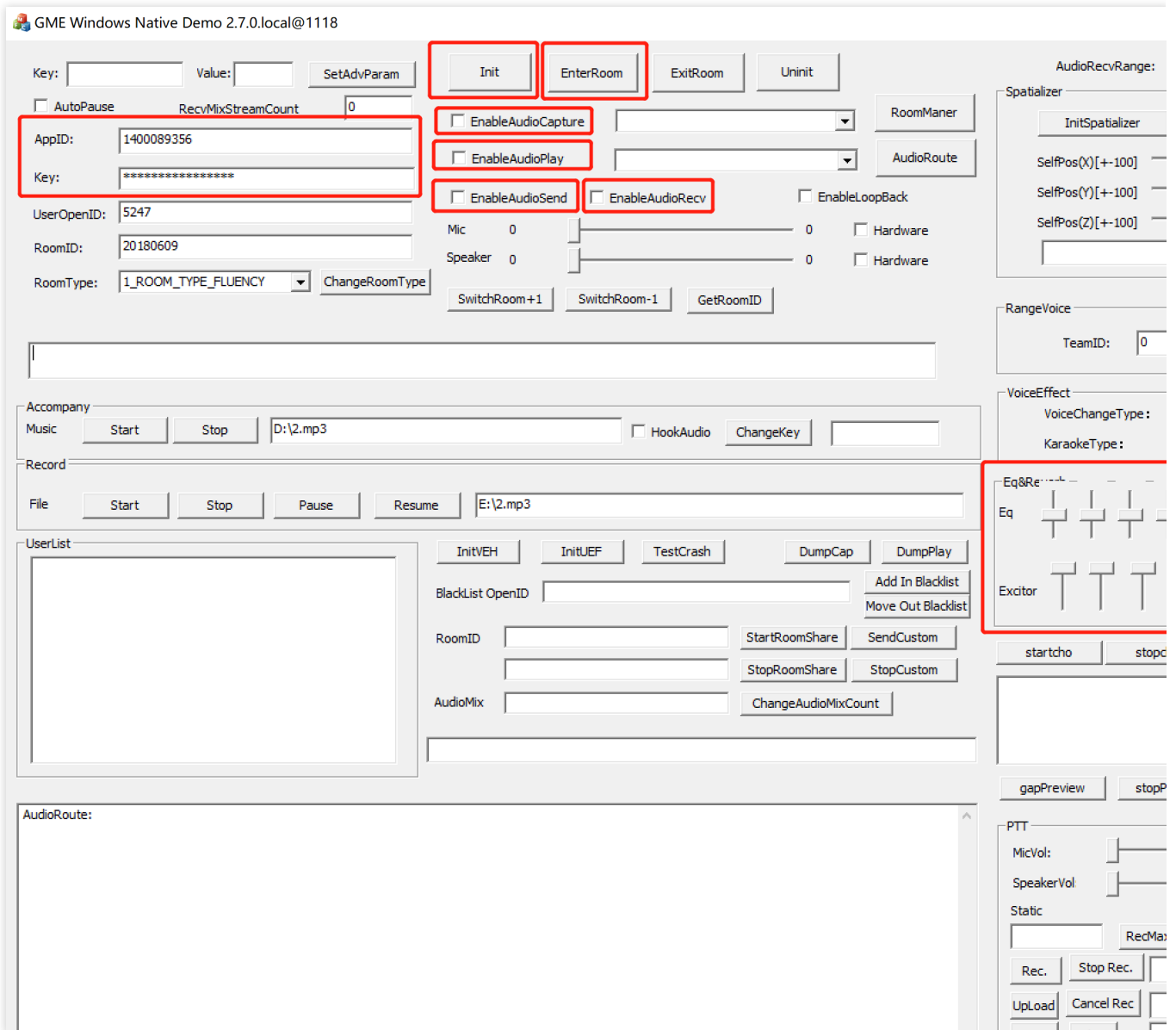
GME SDK에 액세스: 핵심 API 및 음성 채팅 API에 대한 액세스를 포함합니다. 자세한 내용은 [Quick Integration of Native SDK](#), [Quick Integration of SDK for Unity](#), [Quick Integration of SDK for Unreal Engine](#)을 참고하십시오.

Demo

Demo 다운로드

[다운로드 주소 >>](#)

이 무료 Demo는 Windows 실행 프로그램입니다.



매개변수 구성

입력 상자에 GME AppID 및 Key를 입력할 수 있습니다.

대상 방 ID와 OpenID를 입력할 수도 있습니다.

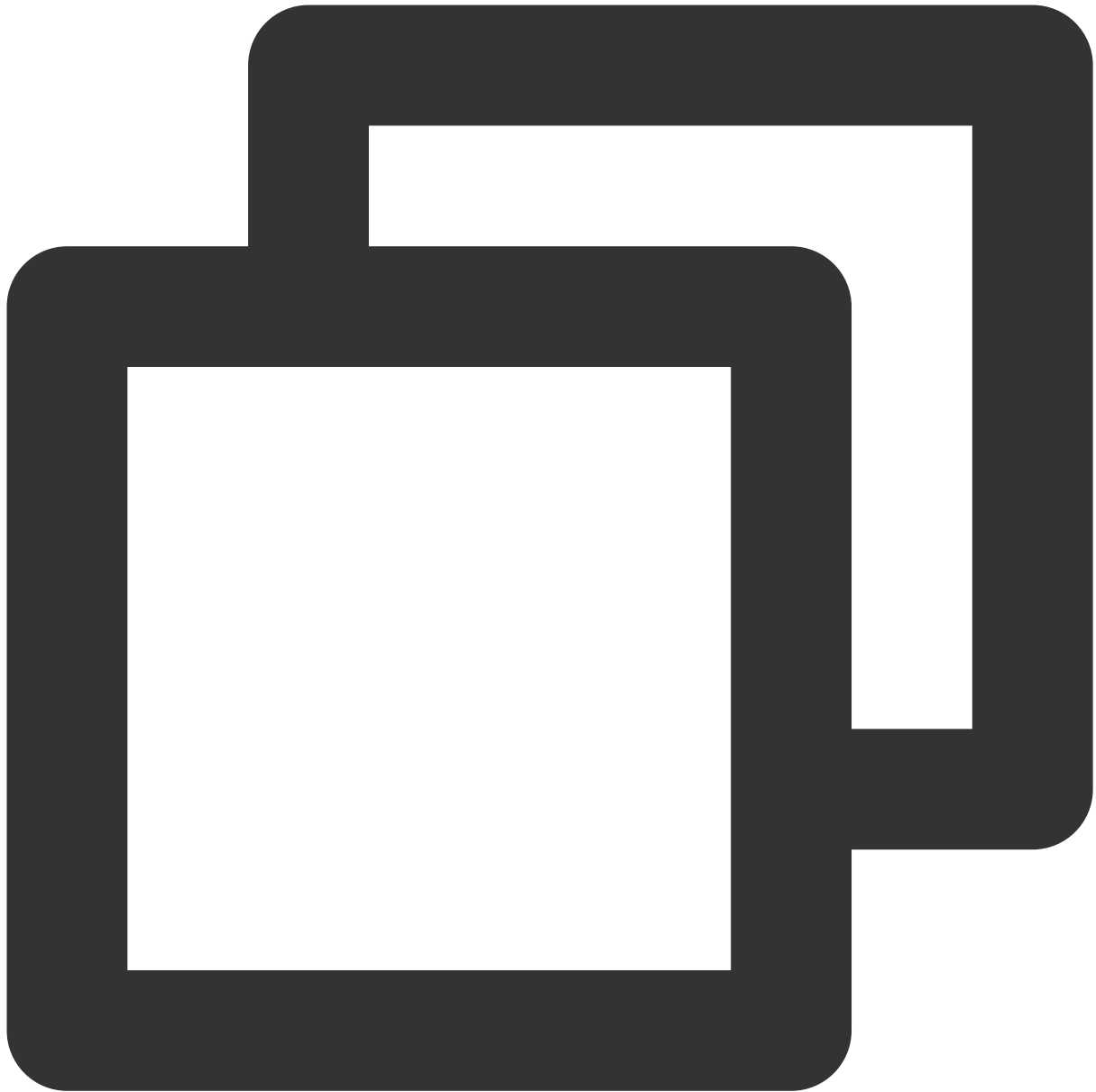
사용 방법

1. 다음 단계에 따라 마이크와 스피커를 엽니다. Init > EnterRoom > EnableCapture > EnablePlay > EnableSend > EnableRecv.
2. 방에 성공적으로 입장한 후 **EnableLoopBack**을 활성화하면 음성을 들을 수 있습니다.
3. **EQ**를 조정합니다. (EQ는 주파수 구간 게인, Exditor 및 Reverb는 리버브레이터)

이퀄라이저 기능 통합

방에 성공적으로 입장한 후에야 API가 로컬에서 캡처한 사운드에 이퀄라이저를 사용할 수 있습니다.

함수 프로토타입



```
int SetKaraokeType(ITMG_VOICE_TYPE_EQUALIZER* pEqualizer, ITMG_VOICE_TYPE_REVERB* p
```

매개변수	유형	설명
pEqualizer	ITMG_VOICE_TYPE_EQUALIZER	주파수 밴드 게인
pReverb	ITMG_VOICE_TYPE_REVERB	HARMONIC 및 REVERB 커버

구조 세부 정보

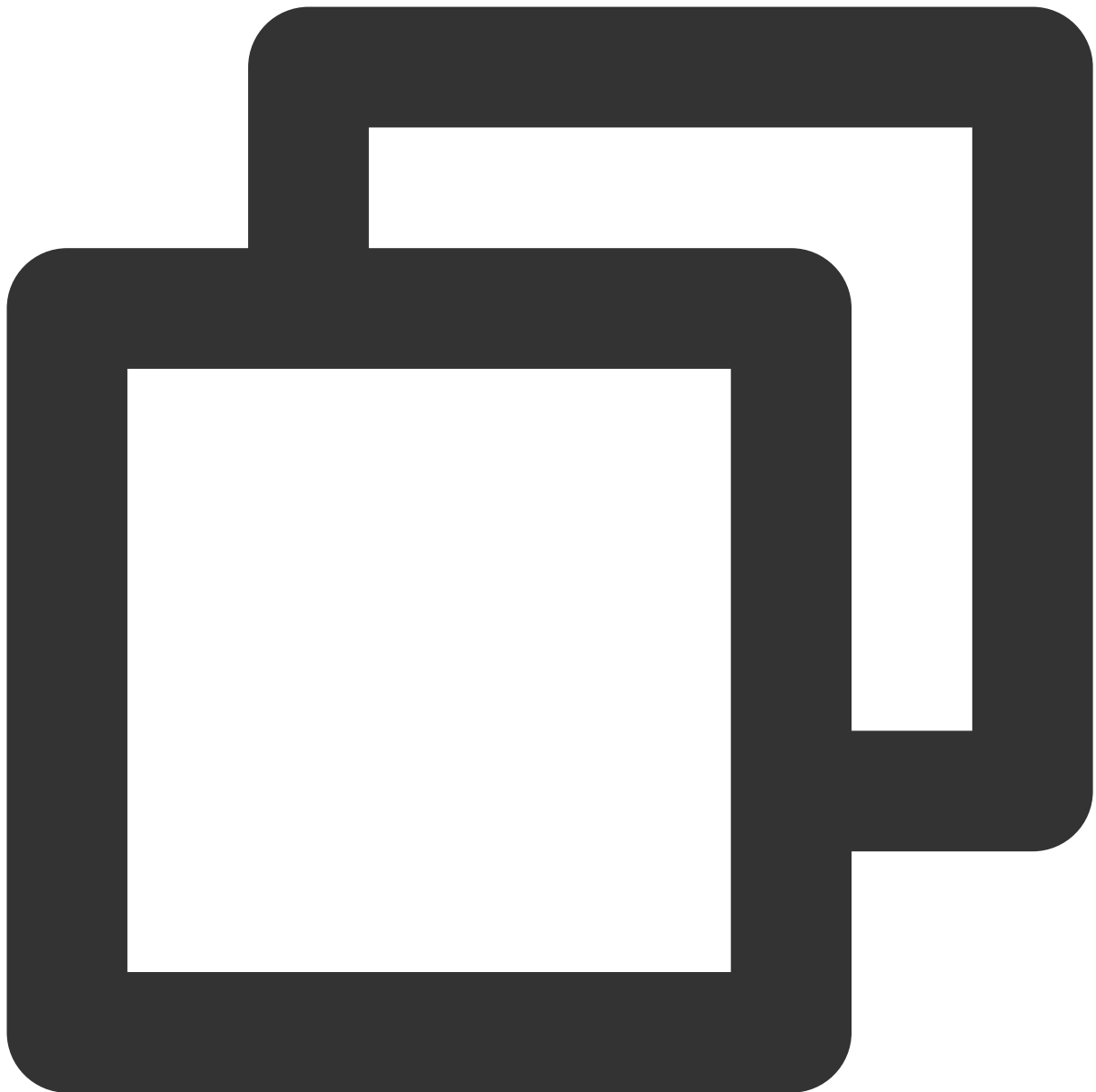
ITMG_VOICE_TYPE_EQUALIZER의 구조 구성원은 float 유형입니다. 값 범위는 -12에서 12까지입니다.

ITMG_VOICE_TYPE_EQUALIZER	설명
EQUALIZER_32HZ	32HZ 밴드에 적용되는 게인
EQUALIZER_64HZ	64HZ 밴드에 적용되는 게인
EQUALIZER_128HZ	128HZ 밴드에 적용되는 게인
EQUALIZER_250HZ	250HZ 밴드에 적용되는 게인
EQUALIZER_500HZ	500HZ 밴드에 적용되는 게인
EQUALIZER_1KHZ	1KHZ 밴드에 적용되는 게인
EQUALIZER_2KHZ	2KHZ 밴드에 적용되는 게인
EQUALIZER_4KHZ	4KHZ 밴드에 적용되는 게인
EQUALIZER_8KHZ	8KHZ 밴드에 적용되는 게인
EQUALIZER_16KHZ	16KHZ 밴드에 적용되는 게인
EQUALIZER_MASTER_GAIN	전체 볼륨

ITMG_VOICE_TYPE_REVERB의 구조 구성원은 float 유형입니다. 값 범위는 0에서 1까지입니다.

ITMG_VOICE_TYPE_REVERB
HARMONIC_GAIN
HARMONIC_START_FREQUENCY
HARMONIC_BASS_CONTROL
REVERB_SIZE
REVERB_DEPTH
REVERB_GAIN
REVERB_ECHO_DEPTH

예시 코드



```
void CTMGSDK_For_AudioDlg::OnVScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
{
    if ((CWnd*)pScrollBar == (CWnd*)&m_SliderEQ1 ||
        (CWnd*)pScrollBar == (CWnd*)&m_SliderEQ2 ||
        (CWnd*)pScrollBar == (CWnd*)&m_SliderEQ3 ||
        (CWnd*)pScrollBar == (CWnd*)&m_SliderEQ4 ||
        (CWnd*)pScrollBar == (CWnd*)&m_SliderEQ5 ||
        (CWnd*)pScrollBar == (CWnd*)&m_SliderEQ6 ||
        (CWnd*)pScrollBar == (CWnd*)&m_SliderEQ7 ||
        (CWnd*)pScrollBar == (CWnd*)&m_SliderEQ8 ||
        (CWnd*)pScrollBar == (CWnd*)&m_SliderEQ9 ||
```

```

(CWnd*)pScrollBar == (CWnd*)&m_SliderEQ10 ||
(CWnd*)pScrollBar == (CWnd*)&m_SliderEQ11 ||
(CWnd*)pScrollBar == (CWnd*)&m_SliderExGain ||
(CWnd*)pScrollBar == (CWnd*)&m_SliderExStartFrequency ||
(CWnd*)pScrollBar == (CWnd*)&m_SliderExBaseCtrl ||
(CWnd*)pScrollBar == (CWnd*)&m_SliderReverbSize ||
(CWnd*)pScrollBar == (CWnd*)&m_SliderReverbDepth ||
(CWnd*)pScrollBar == (CWnd*)&m_SliderReverbGain ||
(CWnd*)pScrollBar == (CWnd*)&m_SliderReverbEchoDepth
)
{
ITMG_VOICE_TYPE_EQUALIZER equalizer = {
    (m_SliderEQ1.GetPos() - 50) * 24.0f / 100,
    (m_SliderEQ2.GetPos() - 50) * 24.0f / 100,
    (m_SliderEQ3.GetPos() - 50) * 24.0f / 100,
    (m_SliderEQ4.GetPos() - 50) * 24.0f / 100,
    (m_SliderEQ5.GetPos() - 50) * 24.0f / 100,
    (m_SliderEQ6.GetPos() - 50) * 24.0f / 100,
    (m_SliderEQ7.GetPos() - 50) * 24.0f / 100,
    (m_SliderEQ8.GetPos() - 50) * 24.0f / 100,
    (m_SliderEQ9.GetPos() - 50) * 24.0f / 100,
    (m_SliderEQ10.GetPos() - 50) * 24.0f / 100,
    (m_SliderEQ11.GetPos() - 50) * 24.0f / 100
};

ITMG_VOICE_TYPE_REVERB reverb = {
    (m_SliderExGain.GetPos()) * 1.0f / 100.0f,
    (m_SliderExStartFrequency.GetPos()) * 1.0f / 100.0f,
    (m_SliderExBaseCtrl.GetPos()) * 1.0f / 100.0f,
    (m_SliderReverbSize.GetPos()) * 1.0f / 100.0f,
    (m_SliderReverbDepth.GetPos()) * 1.0f / 100.0f,
    (m_SliderReverbGain.GetPos()) * 1.0f / 100.0f,
    (m_SliderReverbEchoDepth.GetPos()) * 1.0f / 100.0f
};

m_pTmgContext->GetAudioEffectCtrl()->SetKaraokeType(&equalizer, &reverb);

}
CDialogEx::OnVScroll(nSBCode, nPos, pScrollBar);
}

```

이퀄라이저 사용 가이드

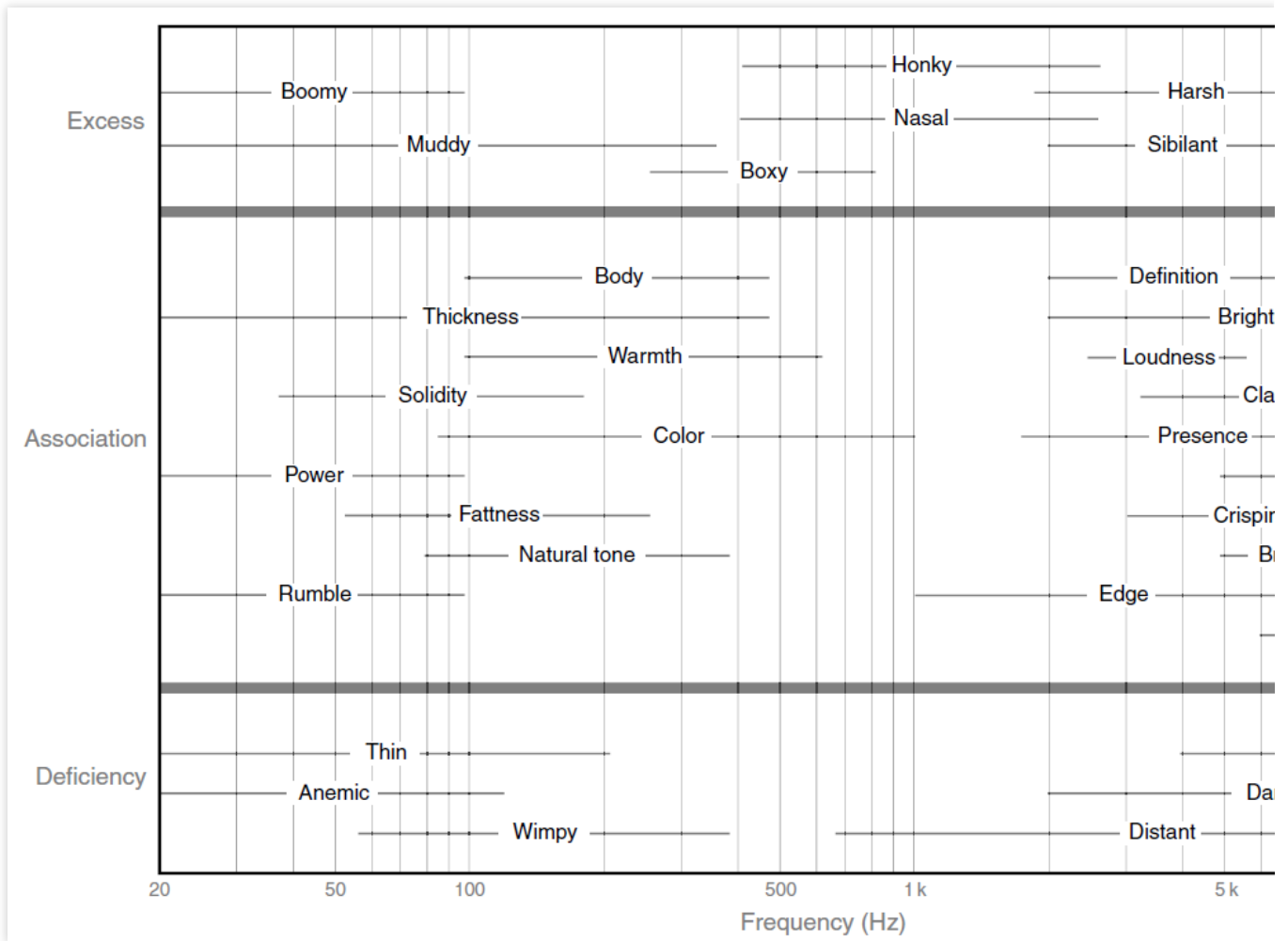
주의사항 :

이 문서는 가이드용으로만 제공됩니다.

사람이 들을 수 있는 소리의 범위는 대략 20HZ에서 20KHZ 사이입니다. 각 주파수 밴드의 인식은 로그 관계를 나타내므로 믹싱 프로젝트는 종종 사람이 들을 수 있는 주파수 대역을 10옥타브로 나누고 다음 영역으로 나눕니다.

밴드	영역	설명
20HZ - 32HZ	초저주파 및 서브우퍼 영역	이 영역에 있는 대부분의 주파수 대역은 사람이 들을 수 있는 하한선보다 낮습니다. 초대형 파이프 오르간의 음악과 폭발 및 영화의 천둥 음향 효과는 이 주파수에 도달할 수 있지만 사람의 목소리는 도달할 수 없습니다. 일반적으로 VOIP 통화는 DC 간섭을 제거하고 다른 주파수 대역에 대한 신호 에너지를 남기기 위해 z 최소로 조정하는 것이 좋습니다.
32HZ - 64HZ	헤비 베이스 영역	주로 드럼과 베이스의 다이브를 조절하는데 사용합니다. 톤이 명확하지 않아 바리톤의 일부가 도달할 수 있습니다. 산업용 주파수 간섭을 제거하고 다른 주파수 대역에 에너지를 남겨두기 위해 VOIP 통화를 낮추는 것이 좋습니다.
64HZ - 125HZ	베이스 영역	타악기의 강도를 결정하는 대부분의 오케스트라 악기의 기본 클럭 범위입니다.
125HZ - 250HZ	베이스 영역	사람 목소리 톤의 인식을 결정하는 것은 사람 목소리의 기본 클럭 범위입니다. 너무 무거우면 뭉개지는 소리가 납니다.
250HZ - 500HZ	미드베이스 영역	주파수 대역에서 톤의 중요한 하모닉스를 포함하고 있어 톤을 조절하는 데 사용할 수 있으며 이 섹션의 강화는 톤을 따뜻하고 두껍게 만드는 반면 과도한 강화는 톤을 뭉개지게 만듭니다.
500HZ - 1KHZ	미드레인지	여성의 목소리 톤을 조절하여 더 풍성하게 만들 때 사용합니다. 너무 높으면 무거운 비음으로 이어지니 주의하십시오. 휴대 전화 및 기타 모바일 재생 장치의 공진 피크는 이 대역에서 최대이므로 너무 높게 설정하지 마십시오.
1KHZ - 2KHZ	미드레인지	인간 귀의 민감한 부분입니다. 이 값은 음량과 현장감에 영향을 미치므로 강화하는 것이 좋습니다.
2KHZ - 4KHZ	소프라노 영역	인간의 귀에서 가장 민감한 부분입니다. 향상은 목소리의 크기를 증가시키고 명료도를 향상시킬 수 있습니다. 너무 높으면 치찰음이 무거울 수 있습니다.
4KHZ - 8KHZ	소프라노 영역	예를 들어 심벌즈 및 현악기와 같은 고주파 악기의 세부 정보를 표시할 수 있습니다. 게다가, 순치음과 마찰음과 같은 음성의 세부 사항을 결정할 수 있습니다. 조정은 권장하지 않습니다.
8KHZ - 16KHZ	초고음 및 초음성 영역	악기의 고주파 오버톤 영역입니다. 조정은 음성에 거의 영향을 미치지 않습니다.

자세한 내용은 다음을 참고하십시오.



실시간 노래방 기능

최종 업데이트 날짜: : 2024-01-18 16:00:04

본문은 개발자가 쉽게 디버깅하고 통합할 수 있도록 실시간 노래방 기능용 Game Multimedia Engine(GME) API에 대해 설명합니다.

전제 조건

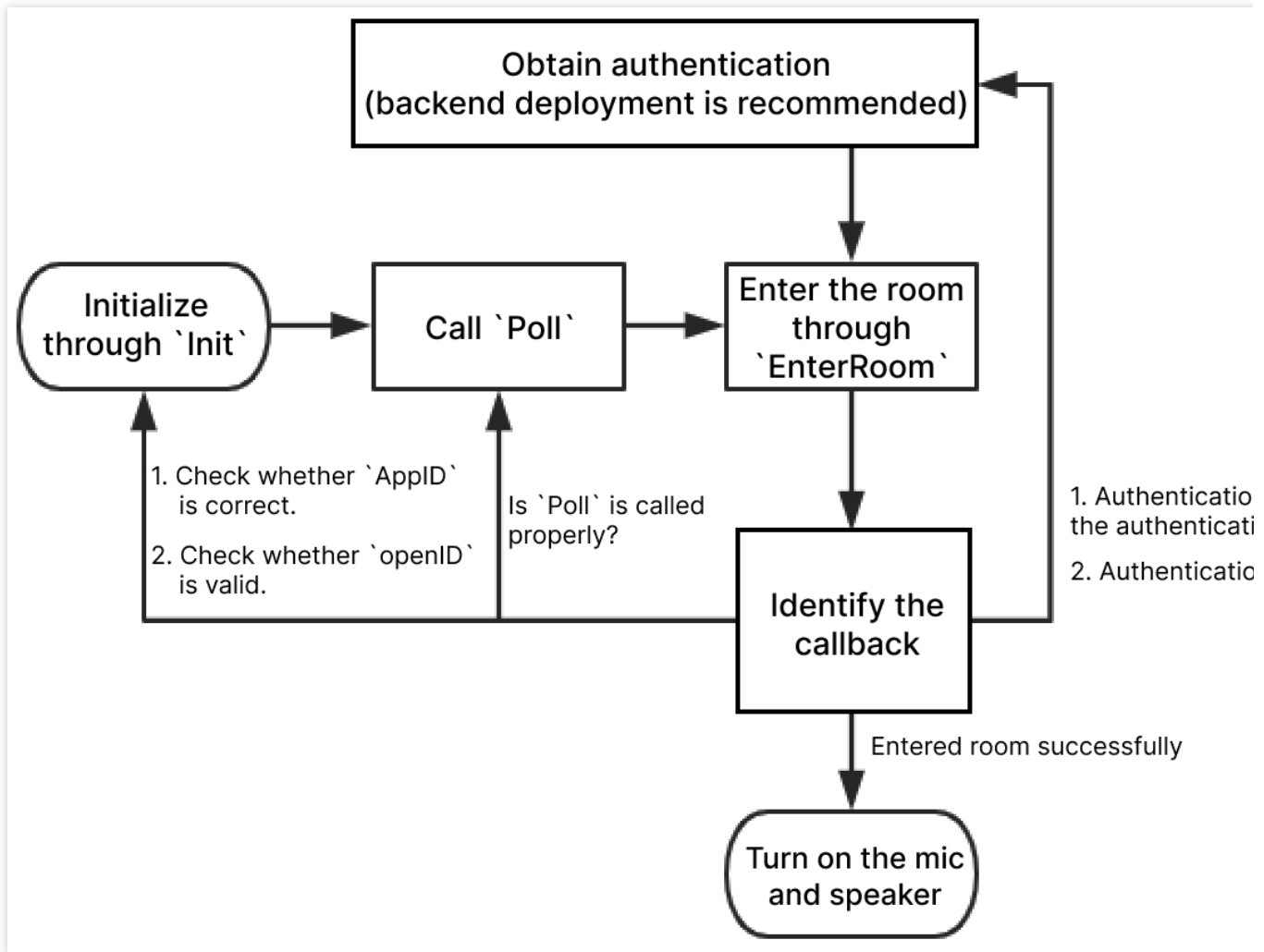
실시간 노래방 기능을 사용하려면 GME SDK를 통합해야 하며, 실시간 음성 통화가 가능해야 이 기능을 사용할 수 있습니다.

방 입장 시 방 유형 매개변수를 입력해야 하며 방에 들어갈 때 RoomType = 2를 사용하는 것이 좋습니다(2 또는 3을 사용하십시오).

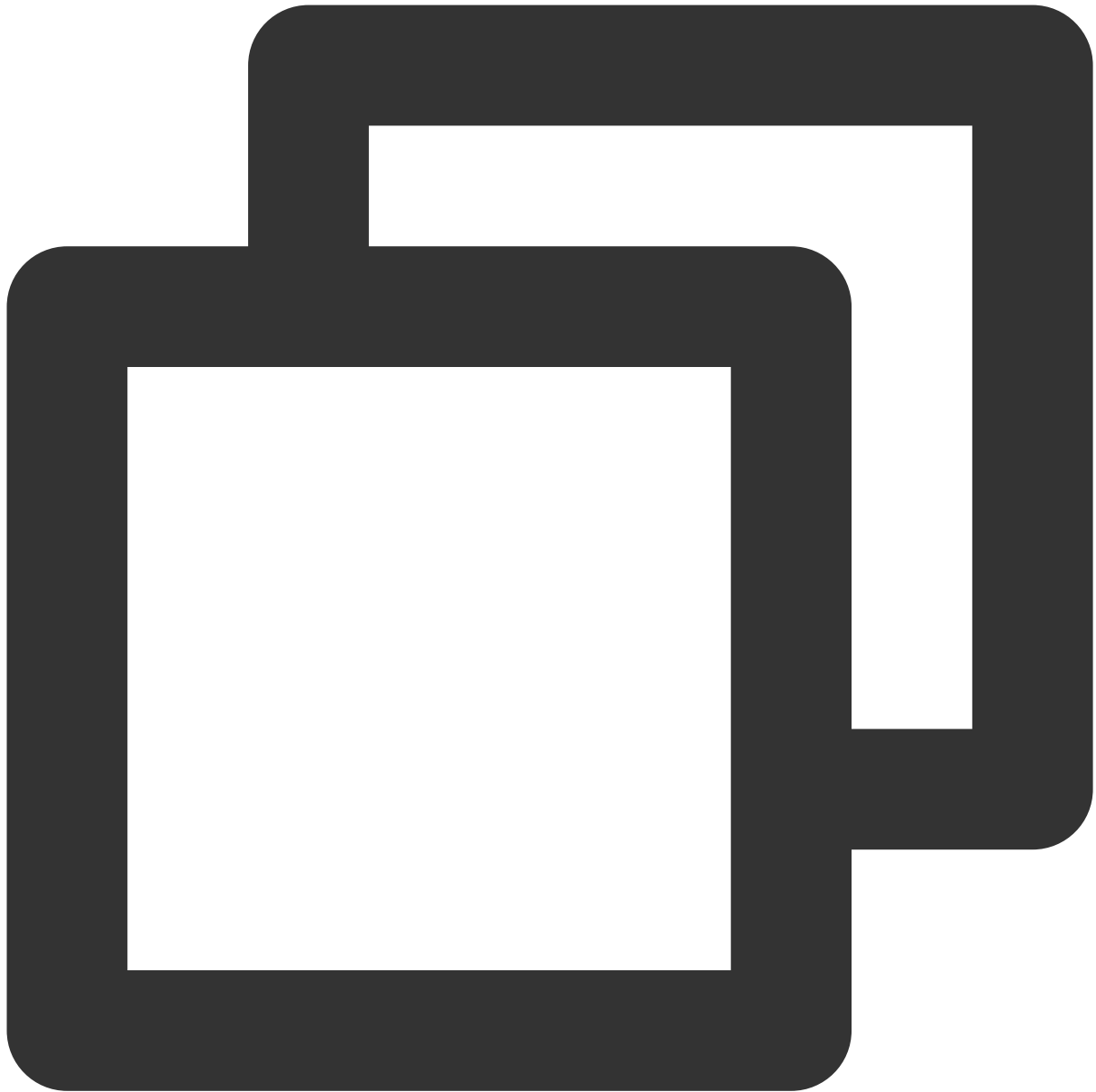
사용 중 오류 코드 프롬프트가 표시되면 [예러 코드](#)를 참고하여 해결할 수 있습니다.

순서도

방 입장 프로세스는 다음 이미지를 참고하십시오.



실시간 음성 채팅방 입장 API:



```
ITMGContext.GetInstance(this).Init(String.valueOf(mAppId), mUserId);//sdk 초기화
ITMGContext.GetInstance(this).SetTMGDelegate(new MyDelegate());//다양한 콜백 및 이벤트
EnginePollHelper.createEnginePollHelper();//콜백을 트리거하기 위해 Poll 함수를 주기적으로

byte[] authbuff = AuthBuffer.getInstance().genAuthBuffer(mAppId, mRoomId, mUserId,m
ITMGContext.GetInstance(this).EnterRoom(mRoomId, 2, authbuff);//방 입장
```

설명 :

자세한 호출 프로세스 및 API 세부 정보는 [각 SDK API 문서](#)를 참고하십시오.

노래방 API 가져오기

1. [다운로드 가이드](#)에서 표준 SDK 파일을 다운로드해야 합니다.
2. 플랫폼별 [노래방 기능 API](#)를 다운로드하고 해당 API 파일을 가져옵니다.

설명 :

이 기능은 mp3와 ogg의 두 가지 형식을 지원합니다.

음악 파일이 ogg 형식인 경우 [클릭하여 ogg 동적 라이브러리를 다운로드](#)하십시오(iOS에는 이미 ogg 동적 라이브러리가 포함되어 있으므로 추가로 가져올 필요가 없습니다).

음악 파일이 mp3 형식인 경우 [클릭하여 mp3 동적 라이브러리를 다운로드](#)하십시오(iOS 플랫폼만 이 동적 라이브러리를 가져오면 되며, 다른 플랫폼은 추가로 가져올 필요가 없습니다).

Android 구성

Android에 해당하는 API는 표준 jar 패키지에 이미 포함되어 있으므로 추가로 API 파일을 다운로드할 필요가 없습니다.

iOS 구성

1. iOS에서 노래방 기능을 사용하려면 관련 동적 라이브러리를 프로젝트에 가져와야 합니다. [mp3 동적 라이브러리를 다운로드하려면 클릭](#)하십시오.
2. 다운로드한 파일을 프로젝트 파일로 가져옵니다. Link Binary With Libraries에 이 동적 라이브러리를 추가합니다.
3. 다른 SDK 헤더 파일과 동일한 디렉터리에 있는 프로젝트에 헤더 파일 TMGEngine_adv.h를 추가합니다.

Windows 구성

Windows에서 노래방 기능을 사용하려면 헤더 파일을 다운로드한 후 헤더 파일 tmg_sdk_adv.h 및 tmg_type_adv.h를 프로젝트로 가져옵니다.

Unity 엔진 구성

Unity 엔진에서 노래방 기능을 사용하려면 헤더 파일을 다운로드하고 Unity 폴더에 있는 코드 파일 TMGEngine_Adv.cs 및 ITMEngine_Adv.cs를 복사하여 프로젝트로 가져와야 합니다.

iOS 플랫폼으로 내보내려면 상기 내용을 참고하여 mp3 동적 라이브러리를 가져오십시오.

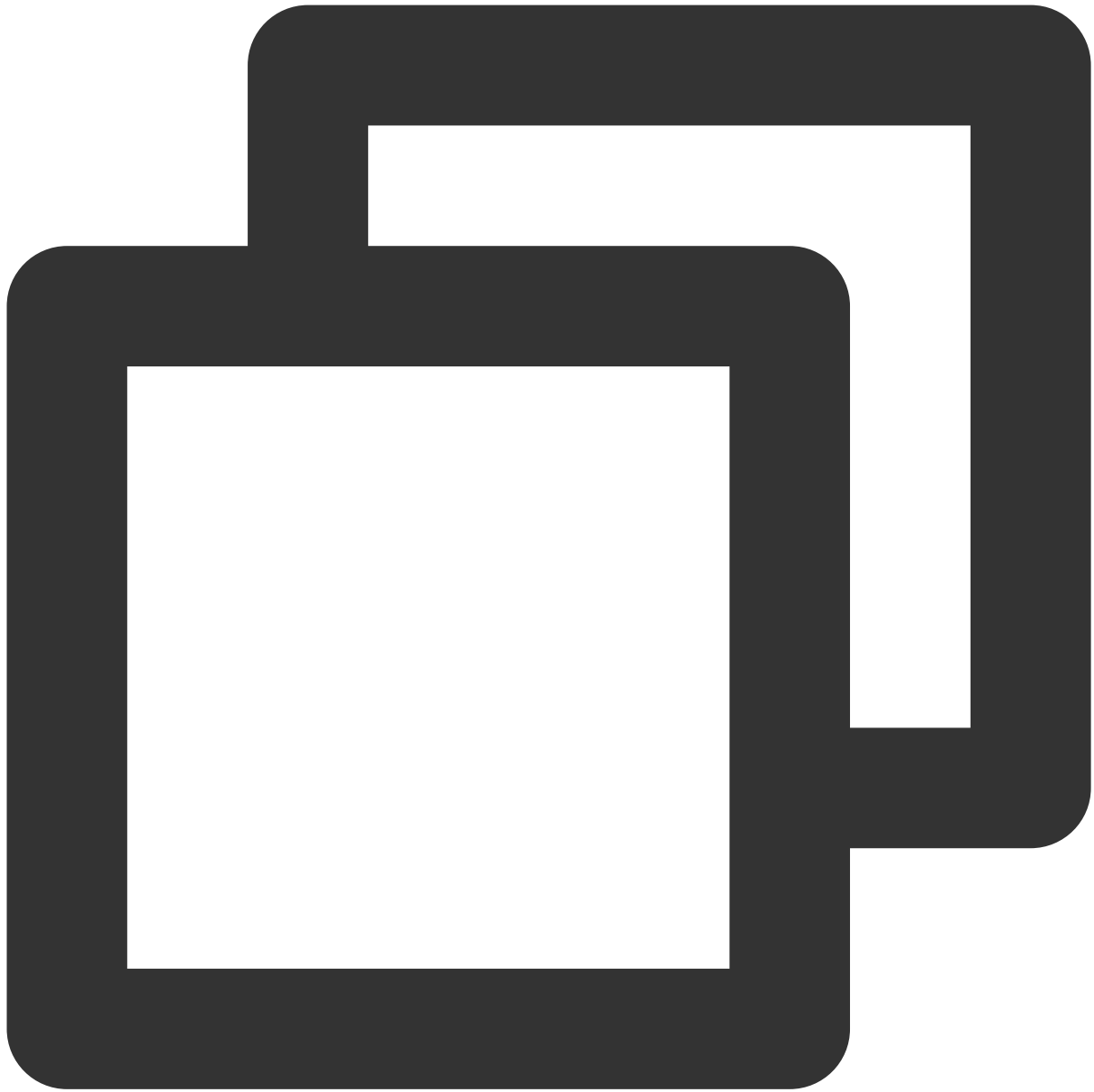
녹음 관련 API

녹음 시작

StartRecord API를 호출하여 녹음을 시작합니다. 녹음이 완료되면 콜백 함수가 있으며 ITMG_MAIN_EVENT_TYPE_RECORD_COMPLETED를 수신해야 합니다.

녹음할 때 마이크가 켜져 있는지 확인하고(장치와 업스트림 모두 켜져 있어야 함) 파일 경로에 액세스할 수 있는지 확인하고 SDK가 적극적으로 폴더를 생성하지 않도록 합니다.

함수 프로토타입

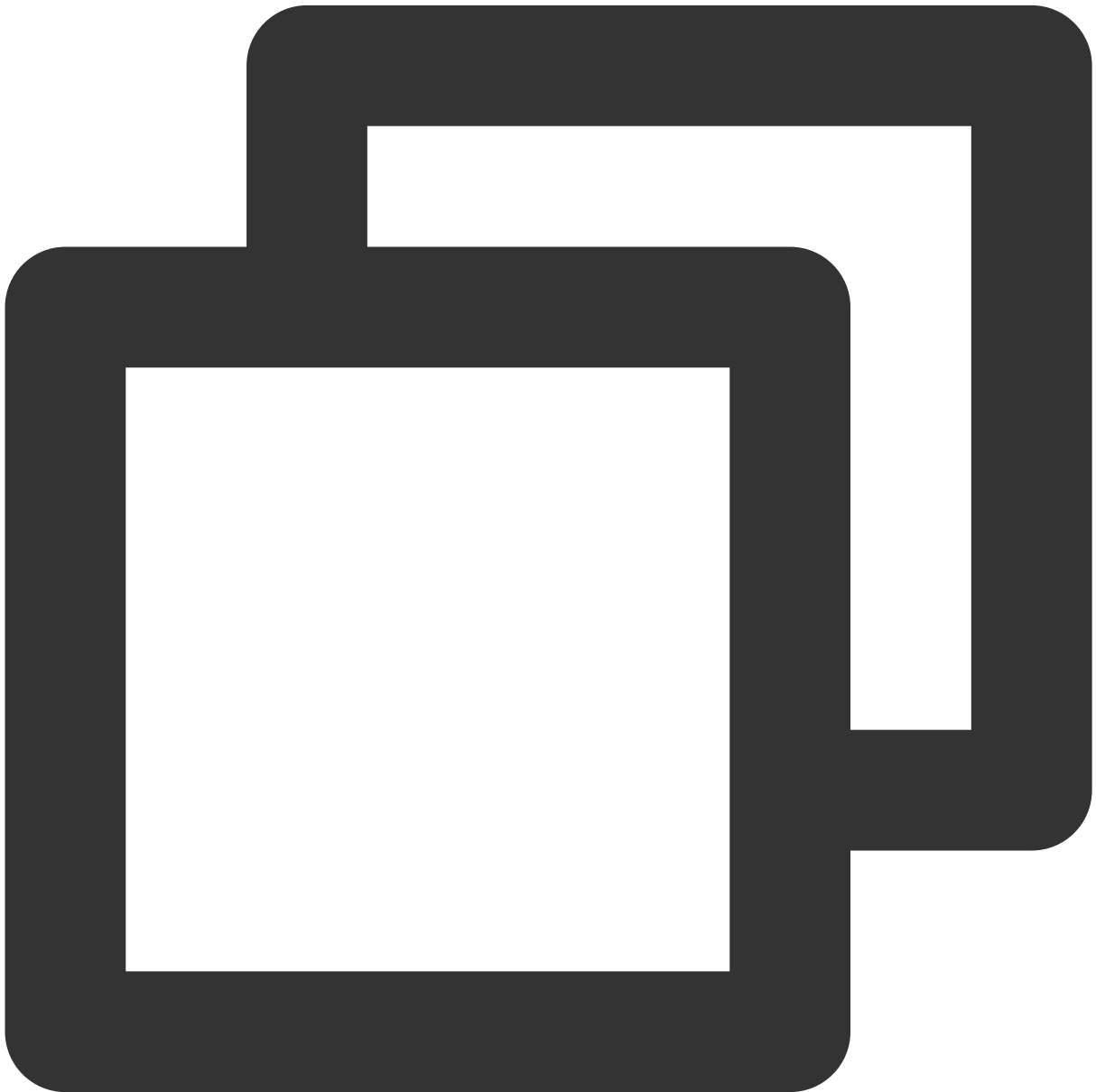


```
int StartRecord(int type, String dstFile, String accMixFile, String accPlayFile)
```

매개변수	유형	설명
type	int	노래방 시나리오에서 이 매개변수는 ITMG_AUDIO_RECORDING_KTV로 전달됩니다. MP3 녹음 파일인 경우에는 ITMG_AUDIO_RECORDING_SELF를 사용하십시오.
dstFile	String	녹음된 음악을 저장하는 데 사용되는 대상 파일 경로입니다.

accMixFile	String	보통 사람 목소리와 합성하여 음악 파일을 만들기 위해 원음이 없는 반주를 사용합니다.
accPlayFile	String	재생에 사용되는 음악 파일은 일반적으로 accMixFile과 동일한 파일입니다. 그러나 사용자가 노래를 잘 모를 경우, 원곡이 포함된 음악 파일 경로를 입력할 수 있습니다. 이 경우 원곡이 포함된 음악이 재생되지만, 원곡이 제외된 반주로 합성됩니다.

예시 코드



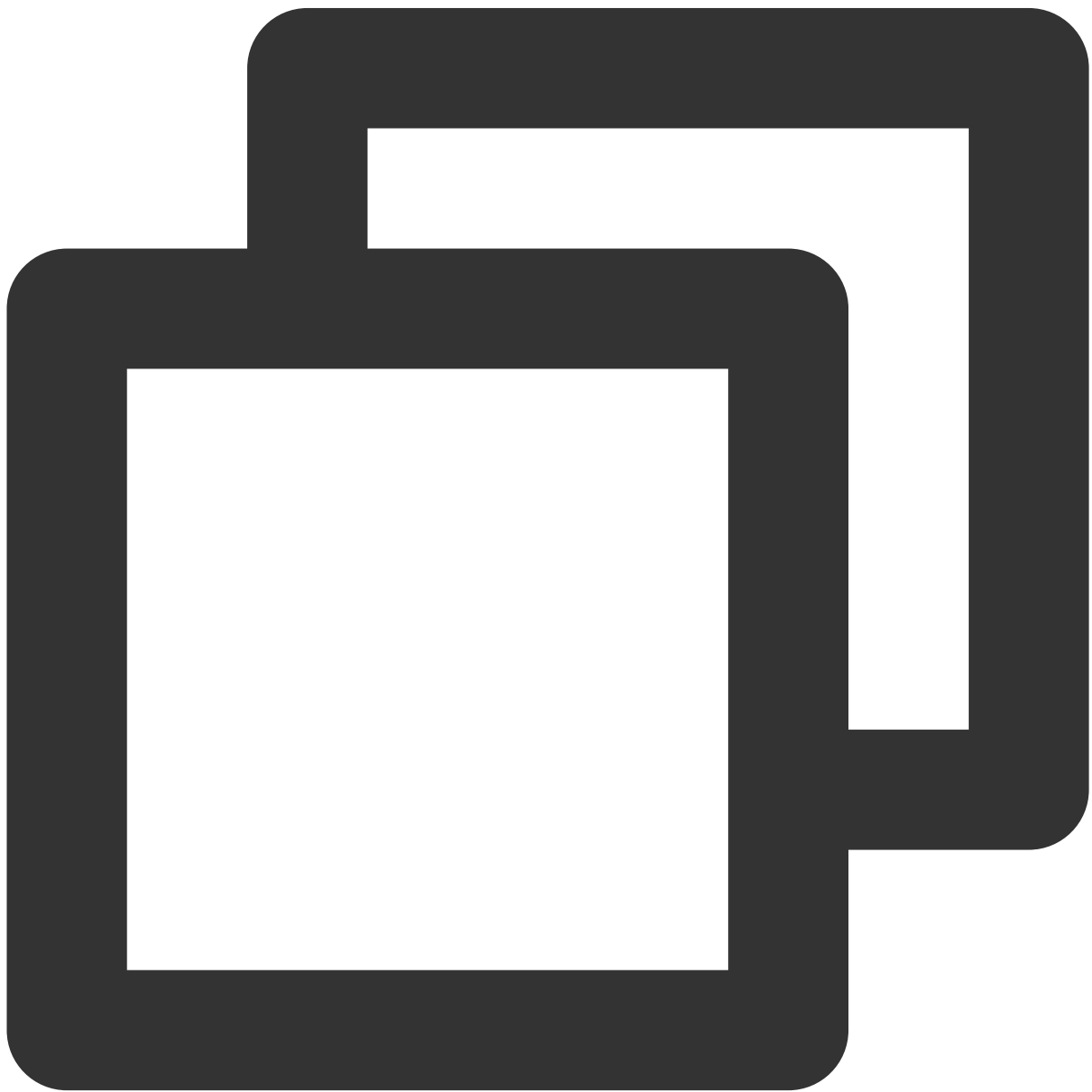
```
//Android
```

```
ITMGAudioRecordCtrl.GetInstance().StartRecord(ITMGAudioRecordCtrl.ITMG_AUDIO_RECORD
//iOS
#import "GMESDK/TMGEngine_adv.h"
[[ITMGAudioRecordCtrl GetInstance]StartPreview]
```

녹음 중지

녹음을 중지하려면 StopRecord API를 호출합니다.

함수 프로토타입

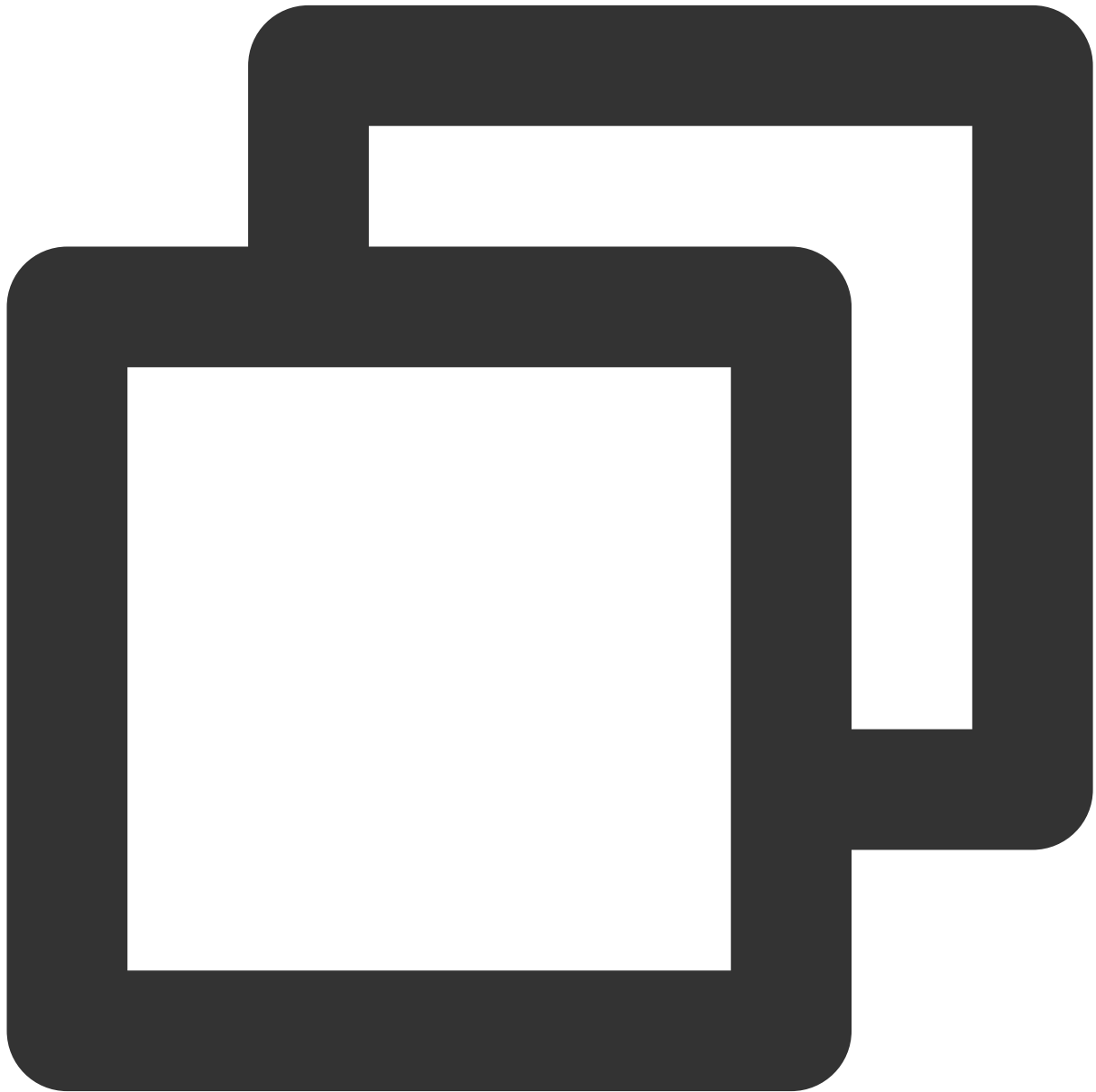


```
int StopRecord()
```

녹음 일시 정지

녹음을 일시 중지하려면 PauseRecord API를 호출합니다.

함수 프로토타입

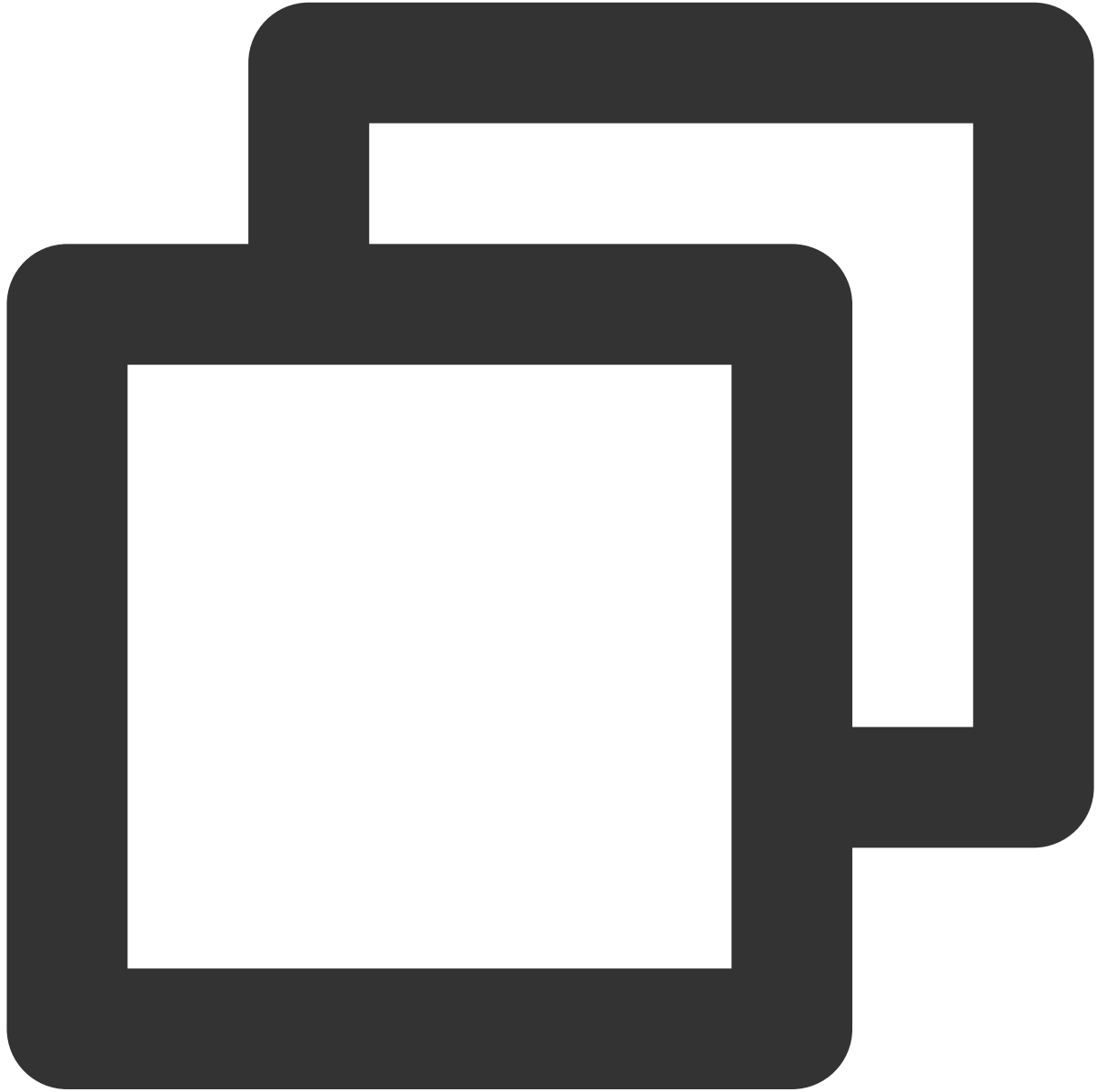


```
int PauseRecord()
```

녹음 재개

녹음을 재개하려면 ResumeRecord API를 호출합니다.

함수 프로토타입



```
int ResumeRecord()
```

녹음 콜백

ITMG_MAIN_EVENT_TYPE_RECORD_COMPLETED 녹음 완료 콜백입니다. 이 콜백은 반주가 재생되거나 StopRecord가 호출될 때 트리거됩니다.

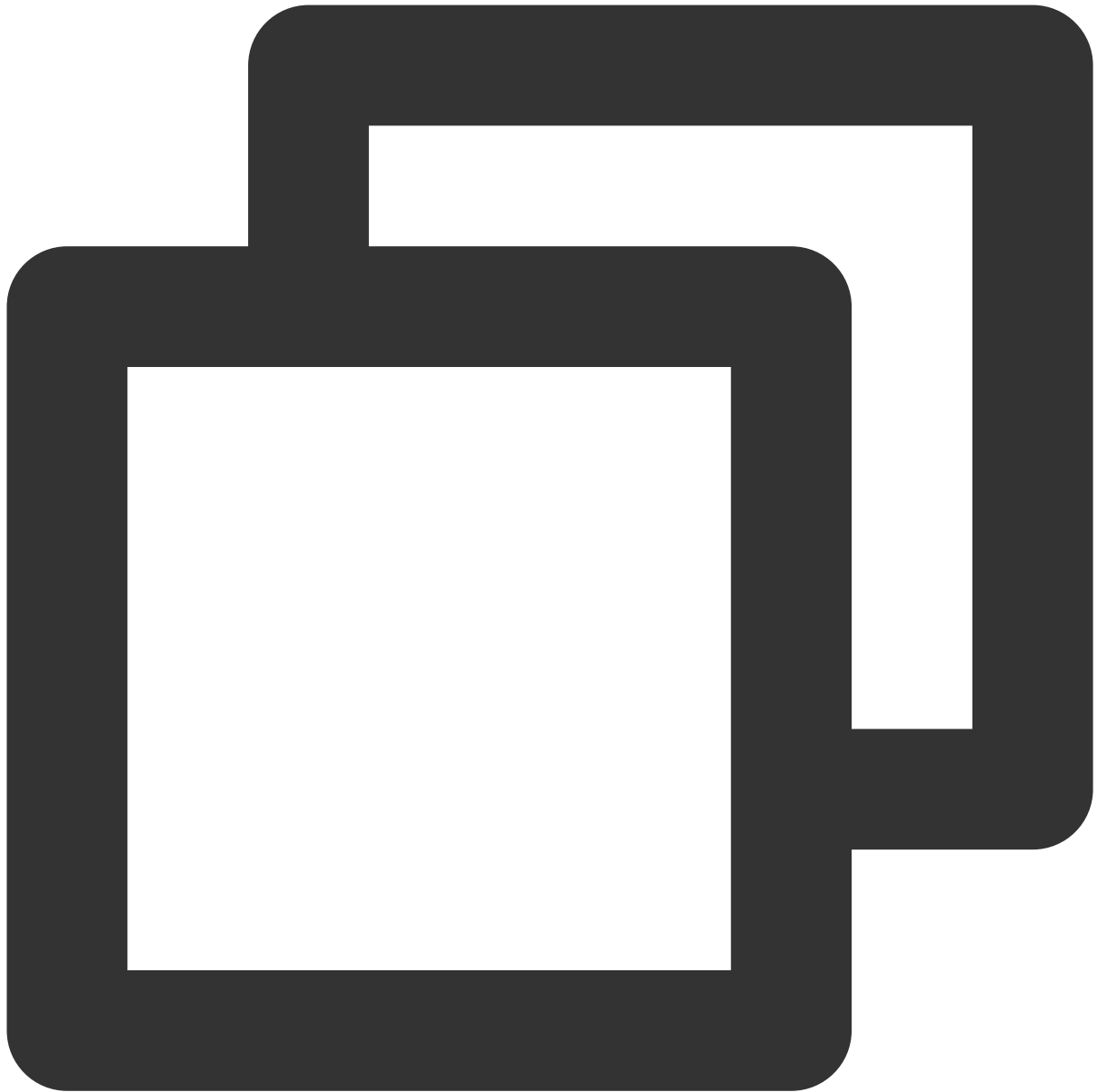
콜백 매개변수

매개변수	유형	설명
result	int	녹음 결과, 0은 성공을 의미합니다. 다른 에러 코드가 있는 경우 에러 코드 문서 를 참고하여 해결할 수 있습니다.
filepath	String	대상 파일의 경로는 StartRecord에 전달된 매개변수인 dstFile입니다.
duration	String	녹음 파일의 길이(밀리초)입니다.

재생 파일 설정

StartRecord API를 호출하여 녹음을 할 때, 재생할 음악 파일이 설정됩니다. 다시 설정하려면 이 API를 호출하여 재생 파일을 재설정할 수 있습니다. 이것은 일반적으로 원곡과 반주 간의 전환에 사용됩니다.

함수 프로토타입



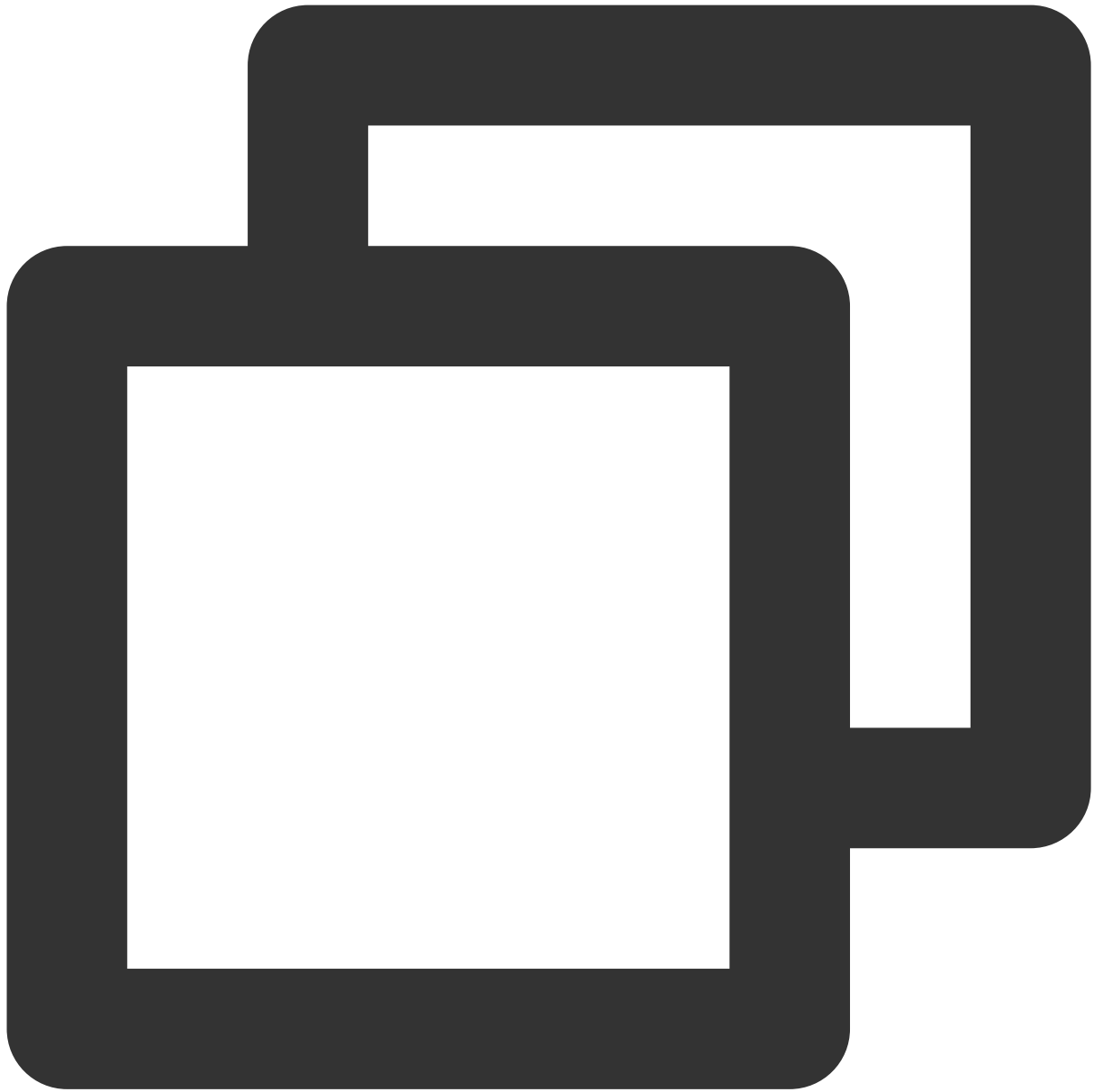
```
int SetAccompanyFile(String accPlayFile)
```

매개변수	유형	설명
accPlayFile	String	재생에 사용되는 음악 파일입니다.

반주 길이 가져오기

이 매개변수를 호출하여 반주 파일인 `accMixFile`의 길이를 가져옵니다. 단위는 밀리초입니다.

함수 프로토타입

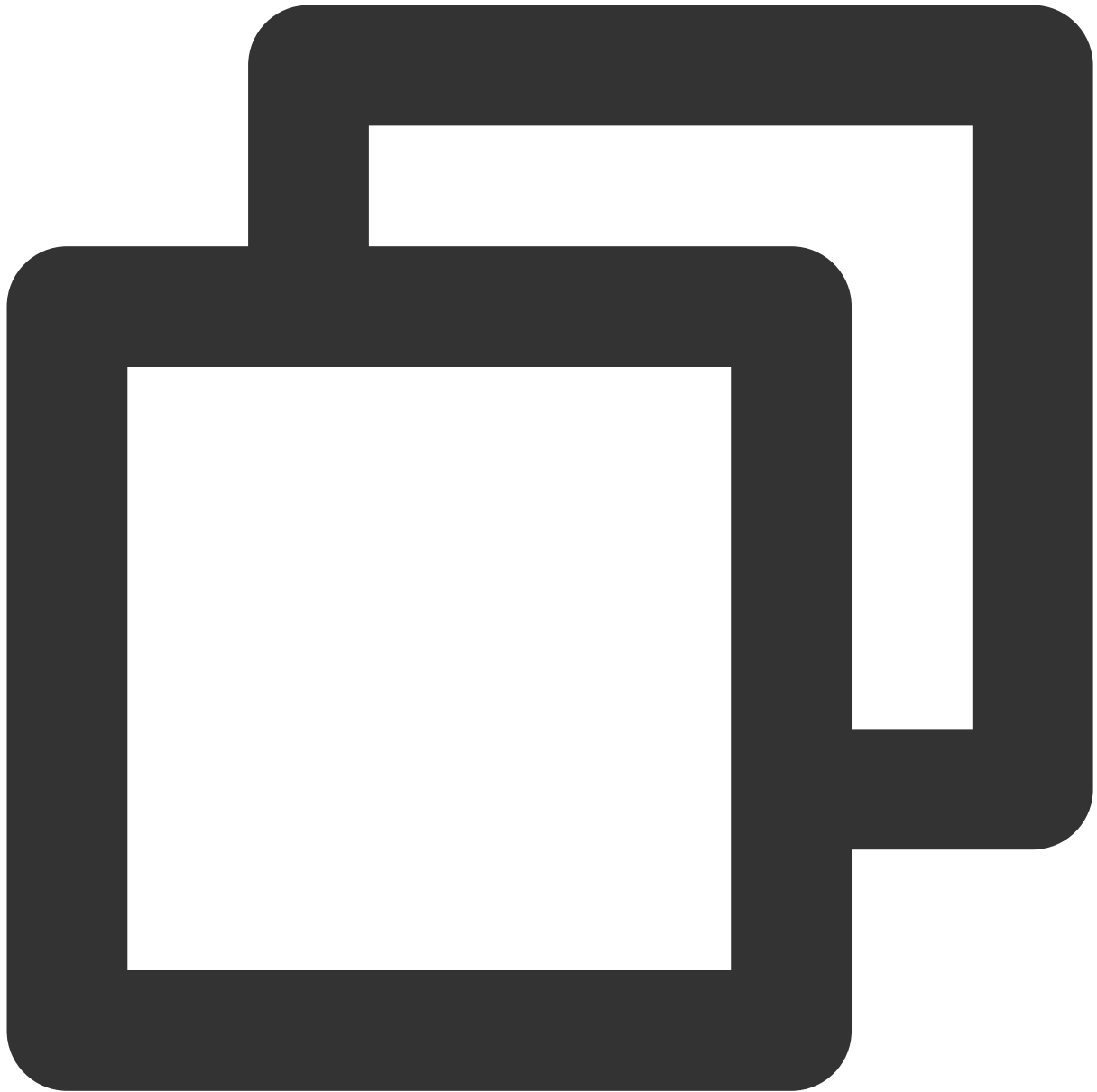


```
int GetAccompanyTotalTimeByMs ()
```

현재의 녹음 길이 가져오기

이 매개변수를 호출하여 현재의 녹음 길이를 가져옵니다. 단위는 밀리초입니다.

함수 프로토타입

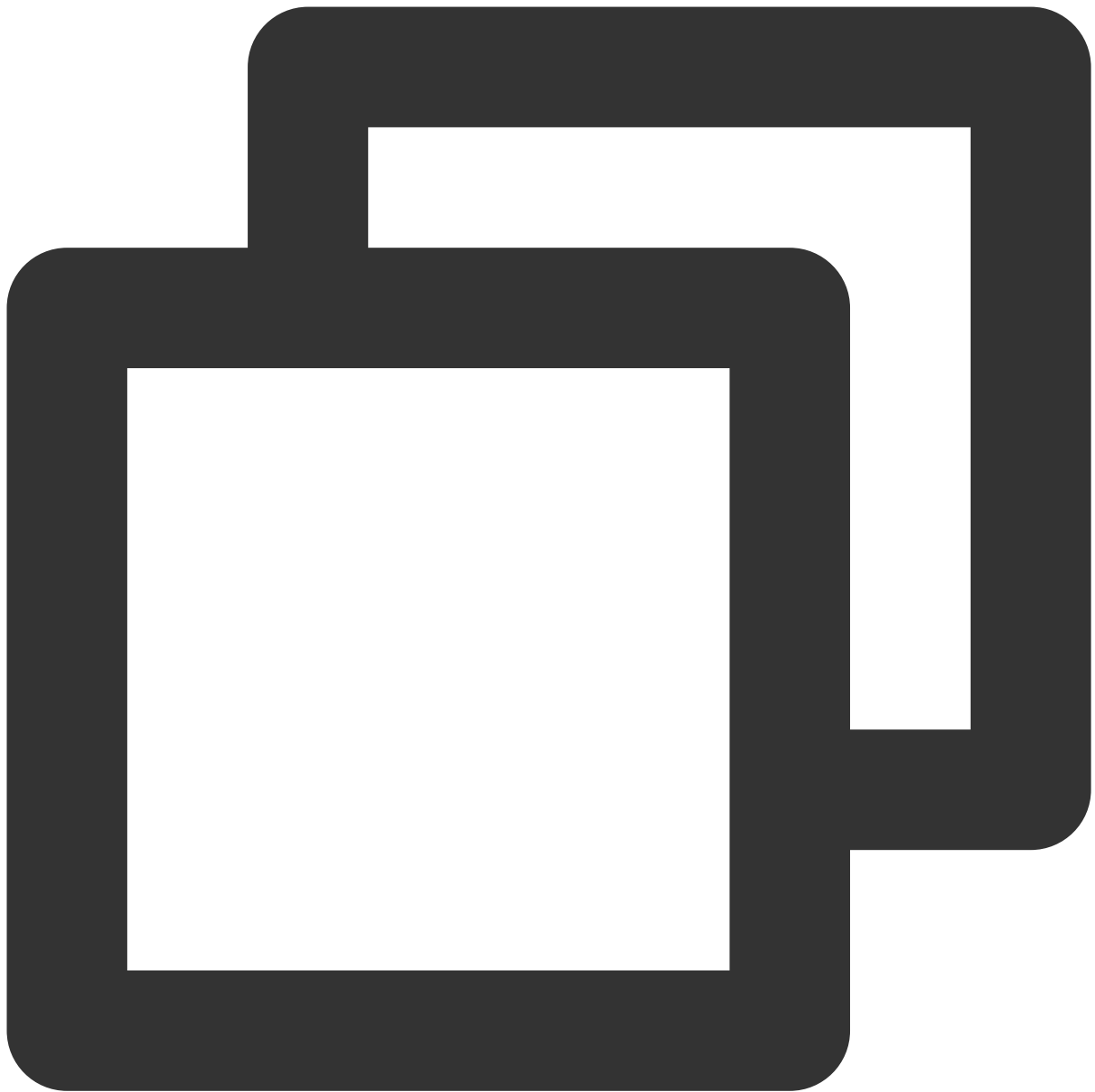


```
int GetRecordTimeByMs ()
```

녹음 리디렉션

녹음 시간을 지정된 시점으로 리디렉션합니다. 매개변수가 현재 시간보다 더 앞쪽인 경우, 반복되는 부분을 다시 녹음합니다. 현재 시간보다 더 뒤쪽인 경우, 녹음되지 않은 부분을 음소거 데이터로 채웁니다.

함수 프로토타입



```
int SetRecordTimeByMs (int timeMs)
```

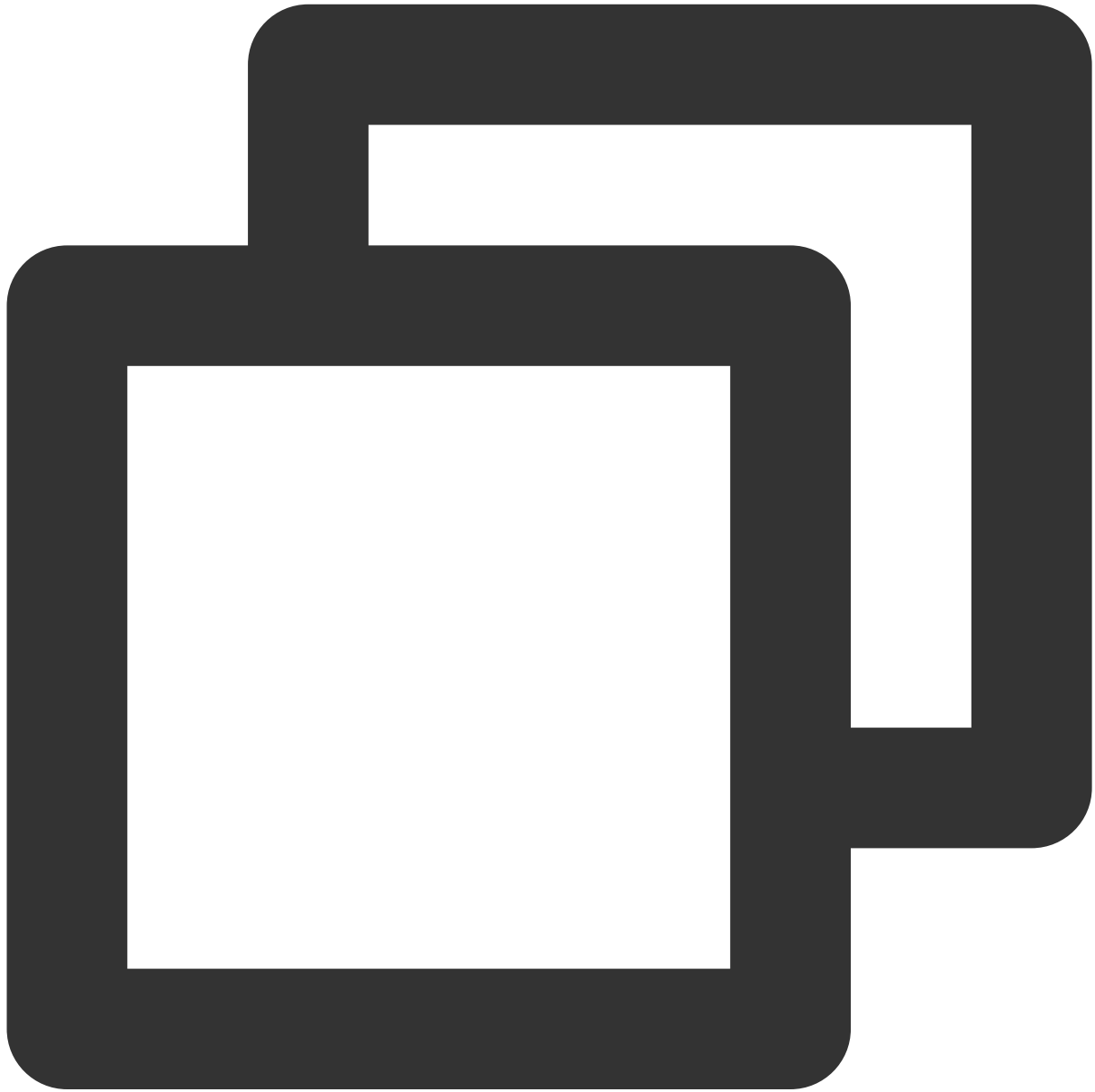
매개변수	유형	설명
timeMs	int	리디렉션 시점이며, 단위는 밀리초입니다.

노래방 파일 미리보기

녹음 파일 길이 가져오기

이 매개변수를 호출하여 녹음 파일의 지속 시간을 가져옵니다.

함수 프로토타입

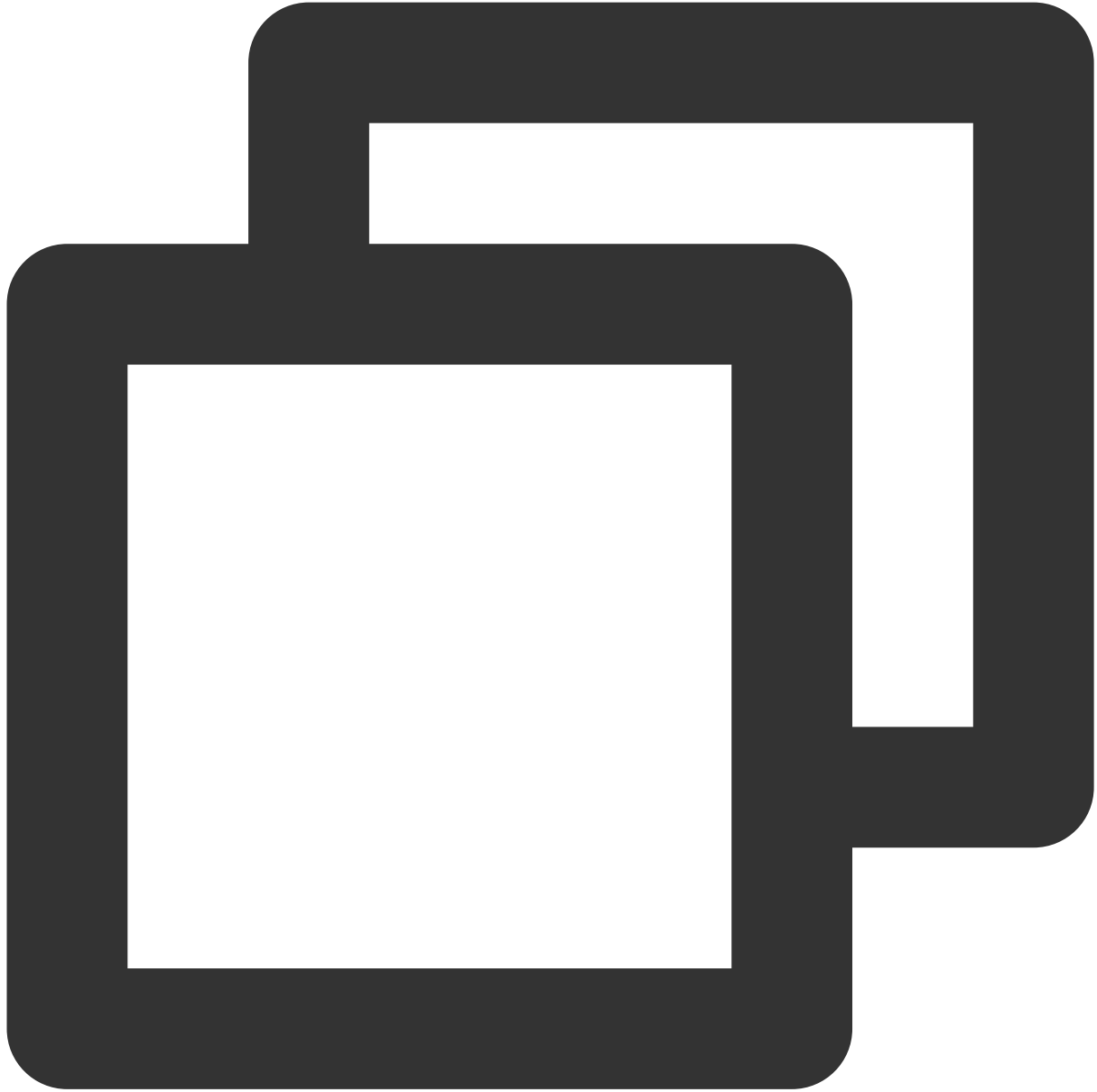


```
int GetRecordFileDurationByMs ()
```

녹음 파일 미리보기 시작

이 매개변수를 호출하여 녹음 파일 미리보기를 시작합니다.

함수 프로토타입

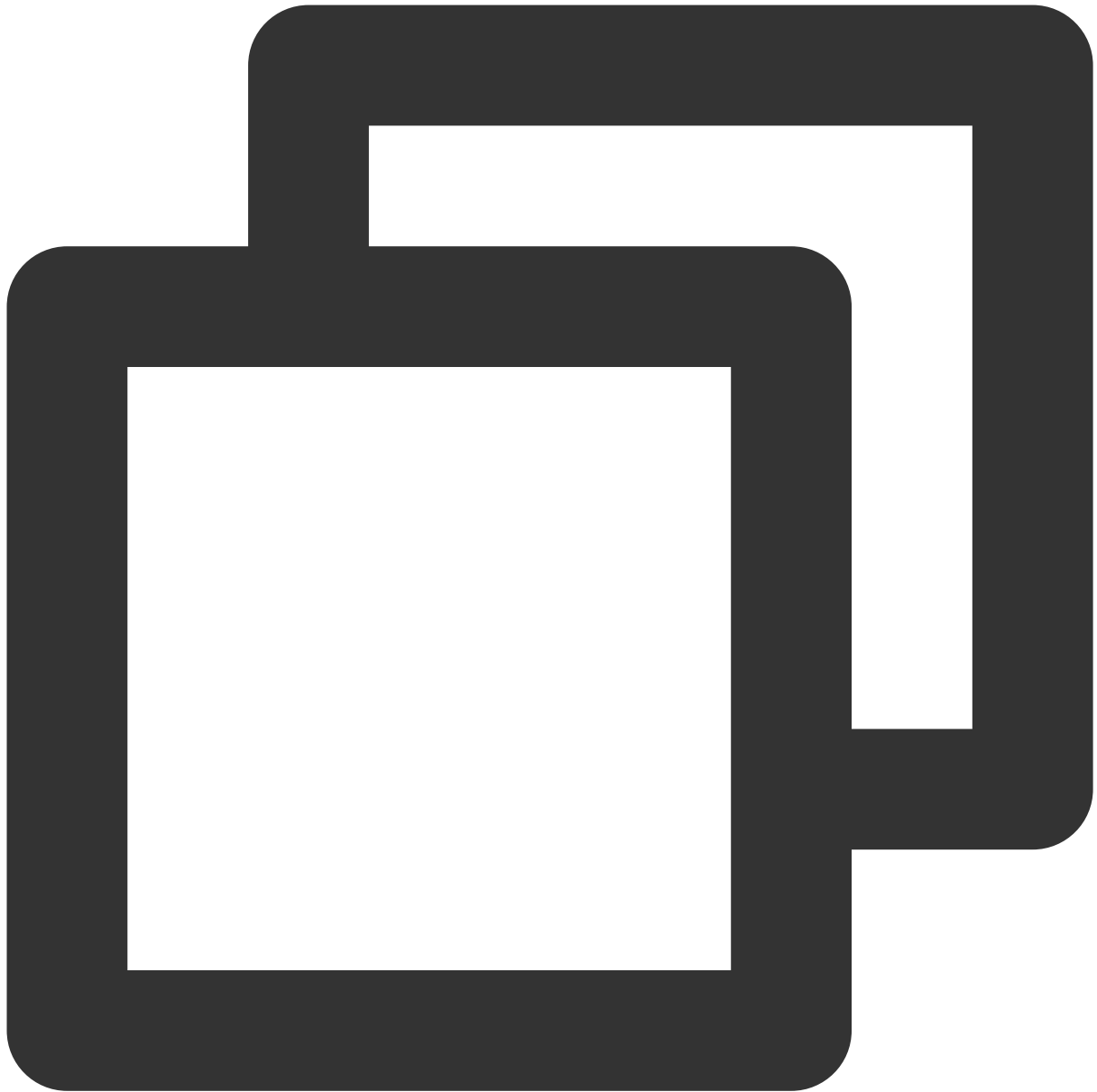


```
int StartPreview()
```

녹음 파일 미리보기 중지

이 매개변수를 호출하여 녹음 파일 미리보기를 중지합니다.

함수 프로토타입

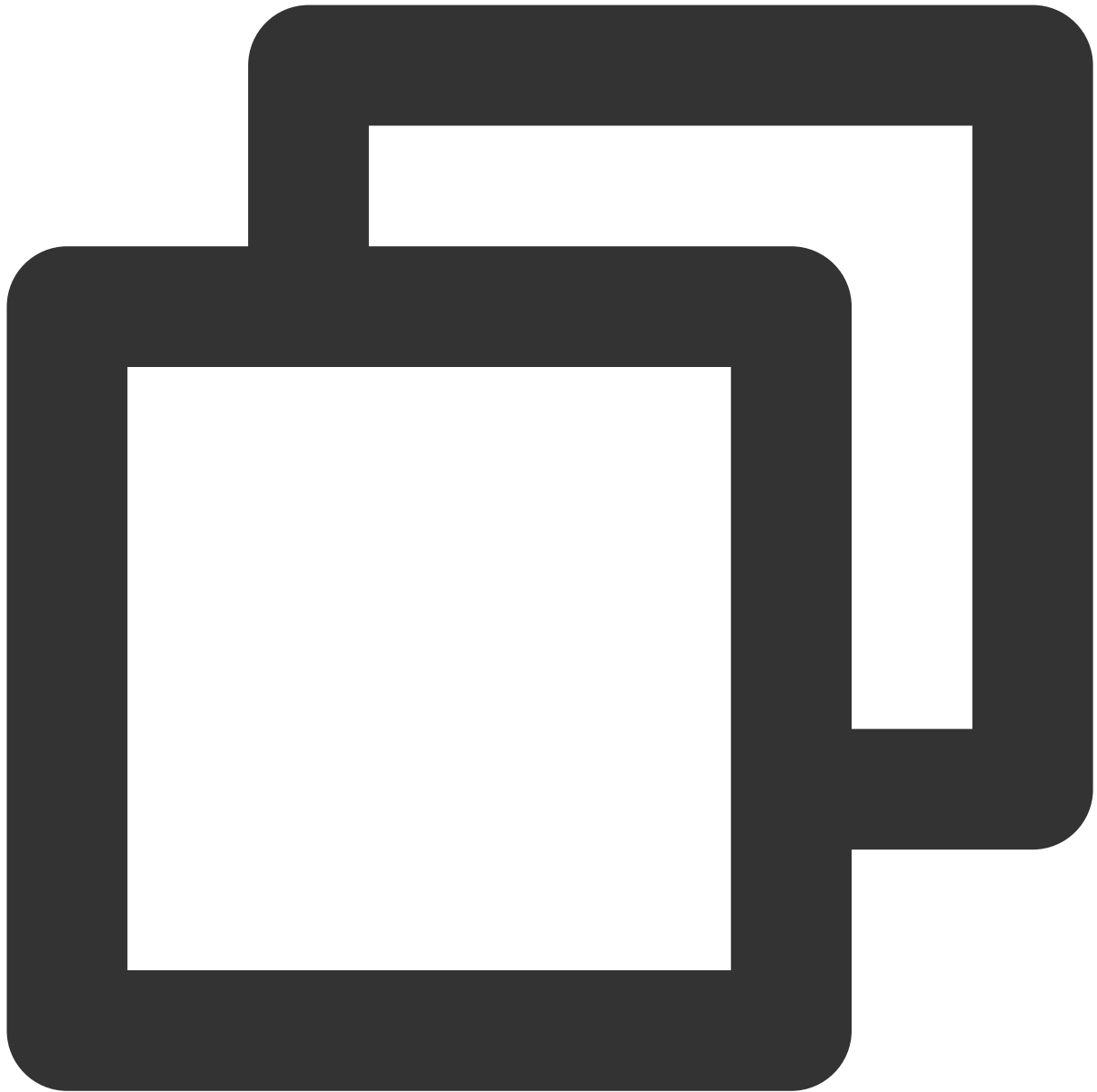


```
int StopPreview()
```

녹음 파일 미리보기 일시 중지

이 매개변수를 호출하여 녹음 파일 미리보기를 일시 중지합니다.

함수 프로토타입

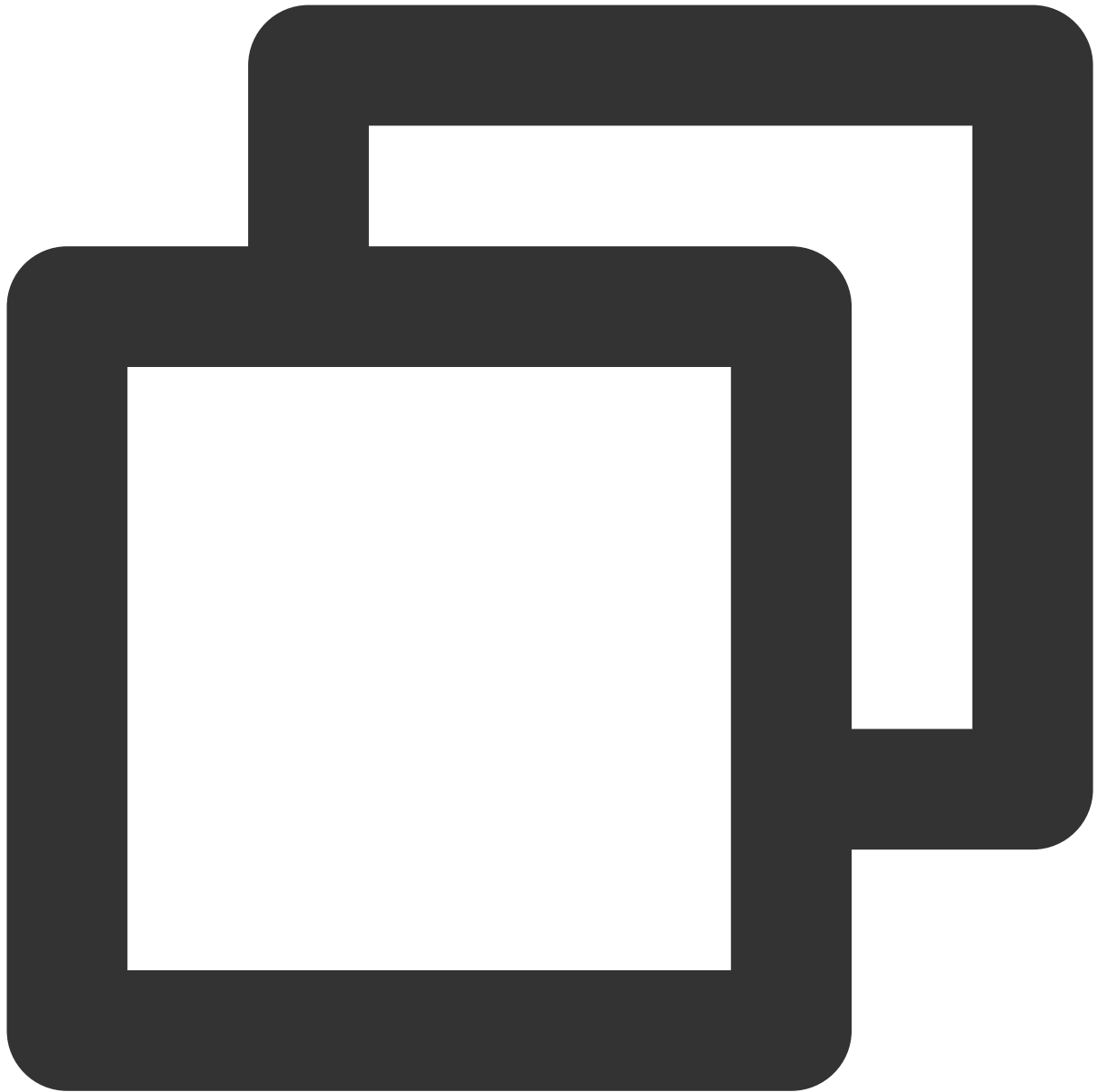


```
int PausePreview()
```

녹음 파일 미리보기 재개

이 매개변수를 호출하여 녹음 파일 미리보기를 재개합니다.

함수 프로토타입

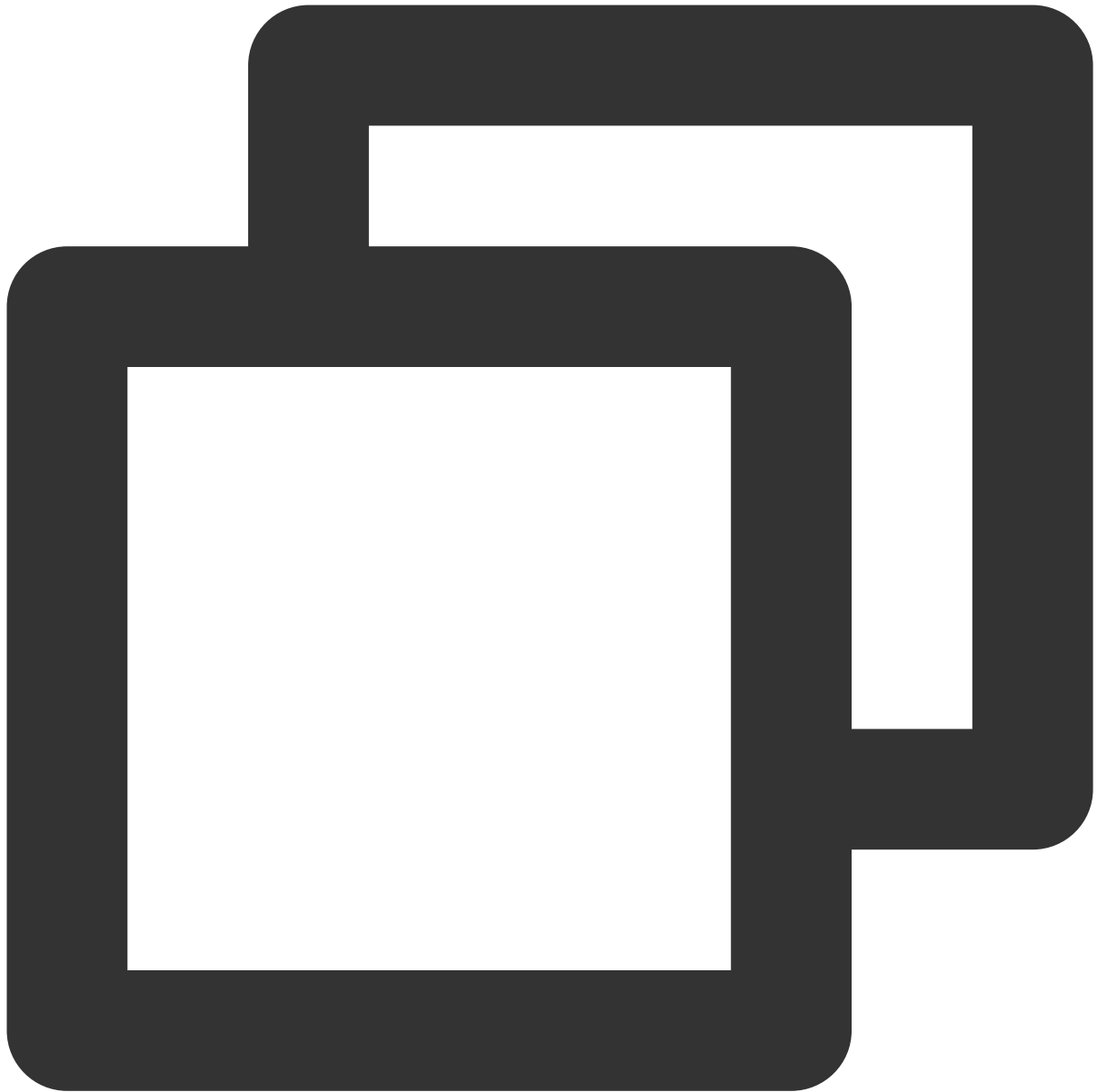


```
int ResumePreview()
```

현재의 미리보기 시점 설정

이 매개변수를 호출하여 현재의 미리보기 시점을 설정합니다.

함수 프로토타입



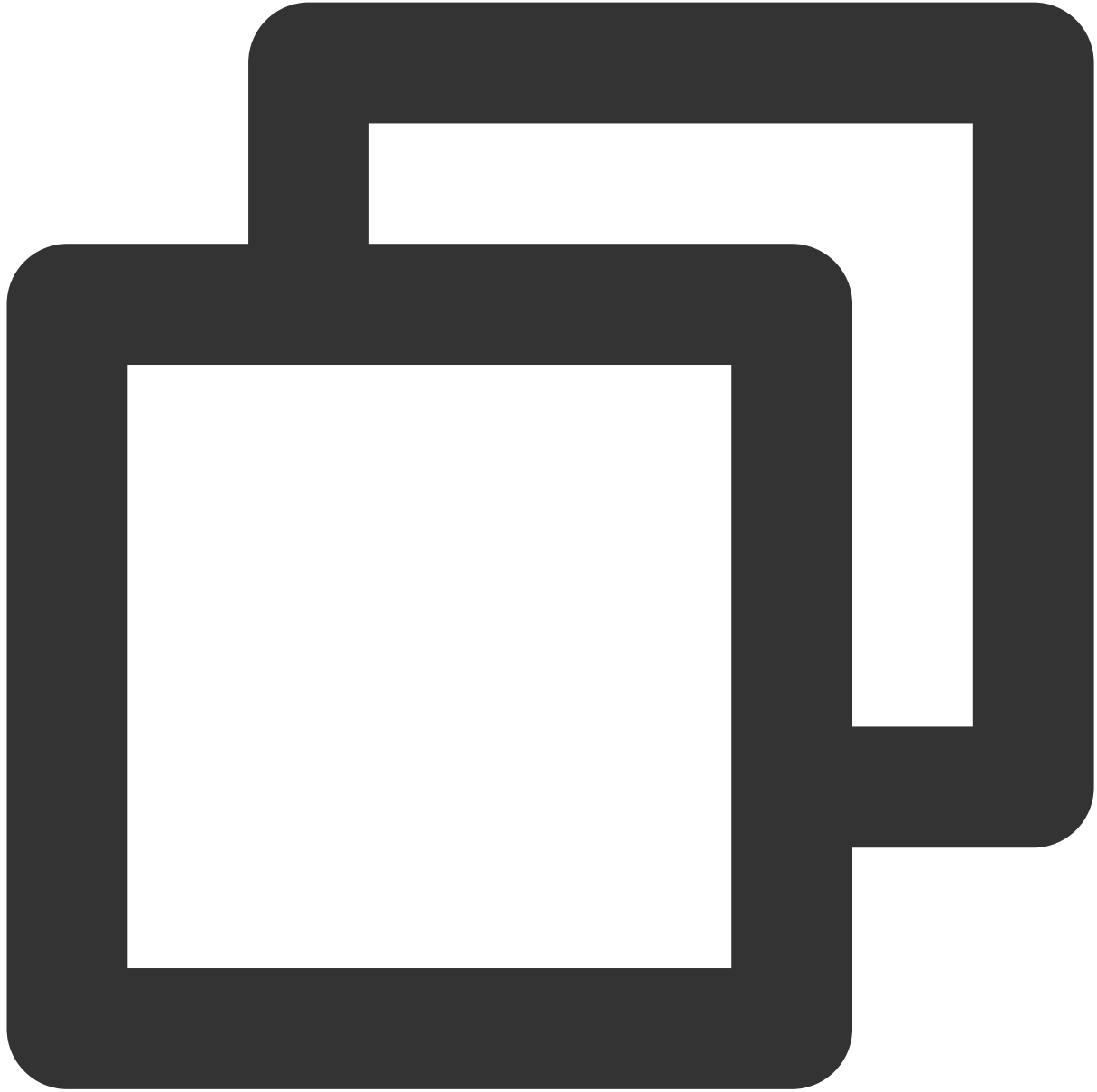
```
int SetPreviewTimeByMs (int time)
```

매개변수	유형	설명
time	int	미리보기 파일의 시점이며 밀리초 단위입니다.

현재의 미리보기 시점 가져오기

이 매개변수를 호출하여 현재 미리보기 시점을 가져옵니다.

함수 프로토타입



```
int GetPreviewTimeByMs ()
```

미리보기 재생 콜백

ITMG_MAIN_EVENT_TYPE_RECORD_PREVIEW_COMPLETED: 미리보기 완료 콜백입니다. 미리보기 파일 재생이 종료되거나 StopPreview API가 호출될 때 트리거됩니다.

콜백 매개변수

--	--	--

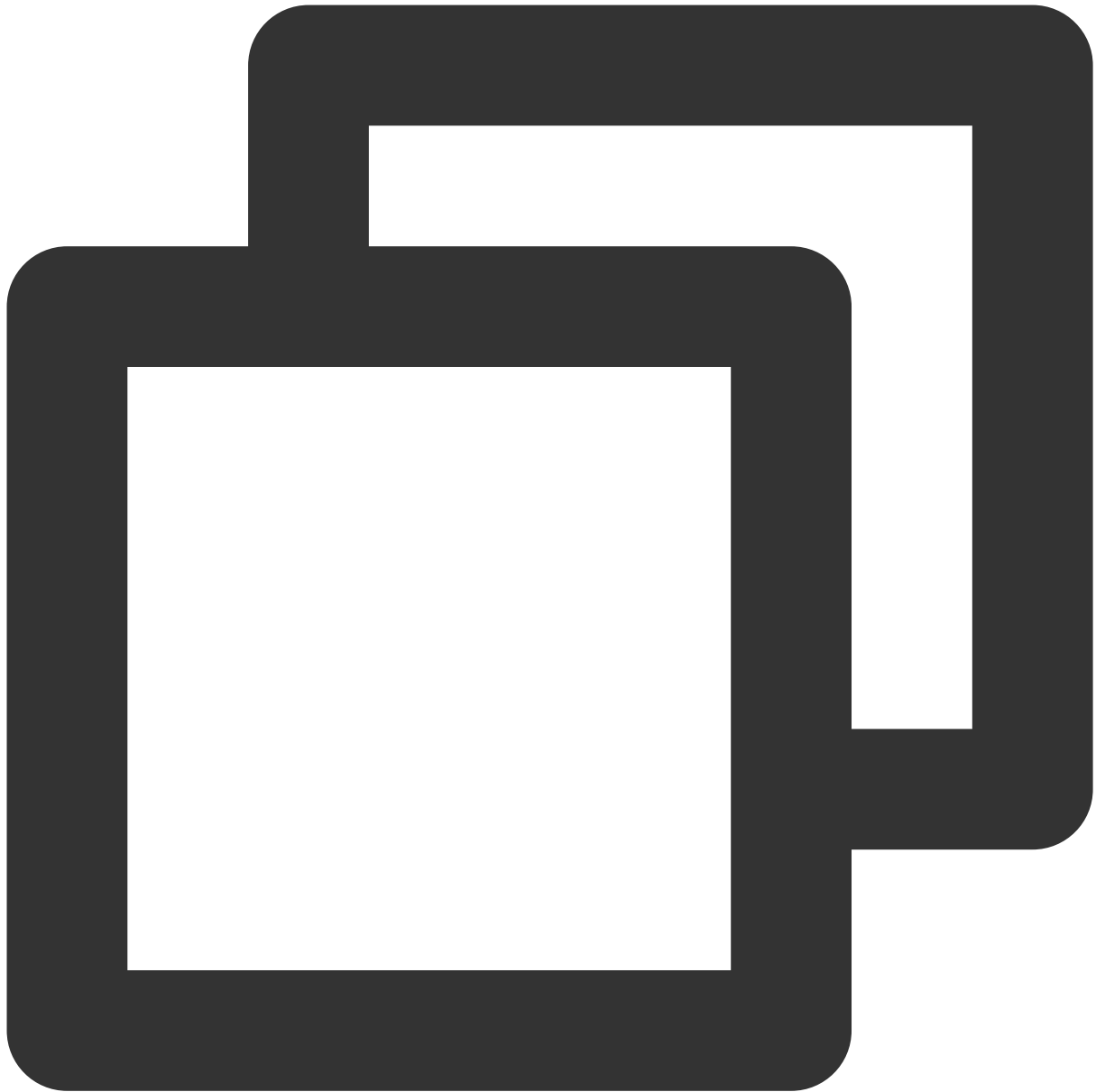
매개변수	유형	설명
result	int	재생 결과. 0은 성공을 의미합니다.

파일 합성 API

파일 병합

이 매개변수를 호출하여 녹음된 음성과 반주를 하나의 파일로 병합합니다.

함수 프로토타입

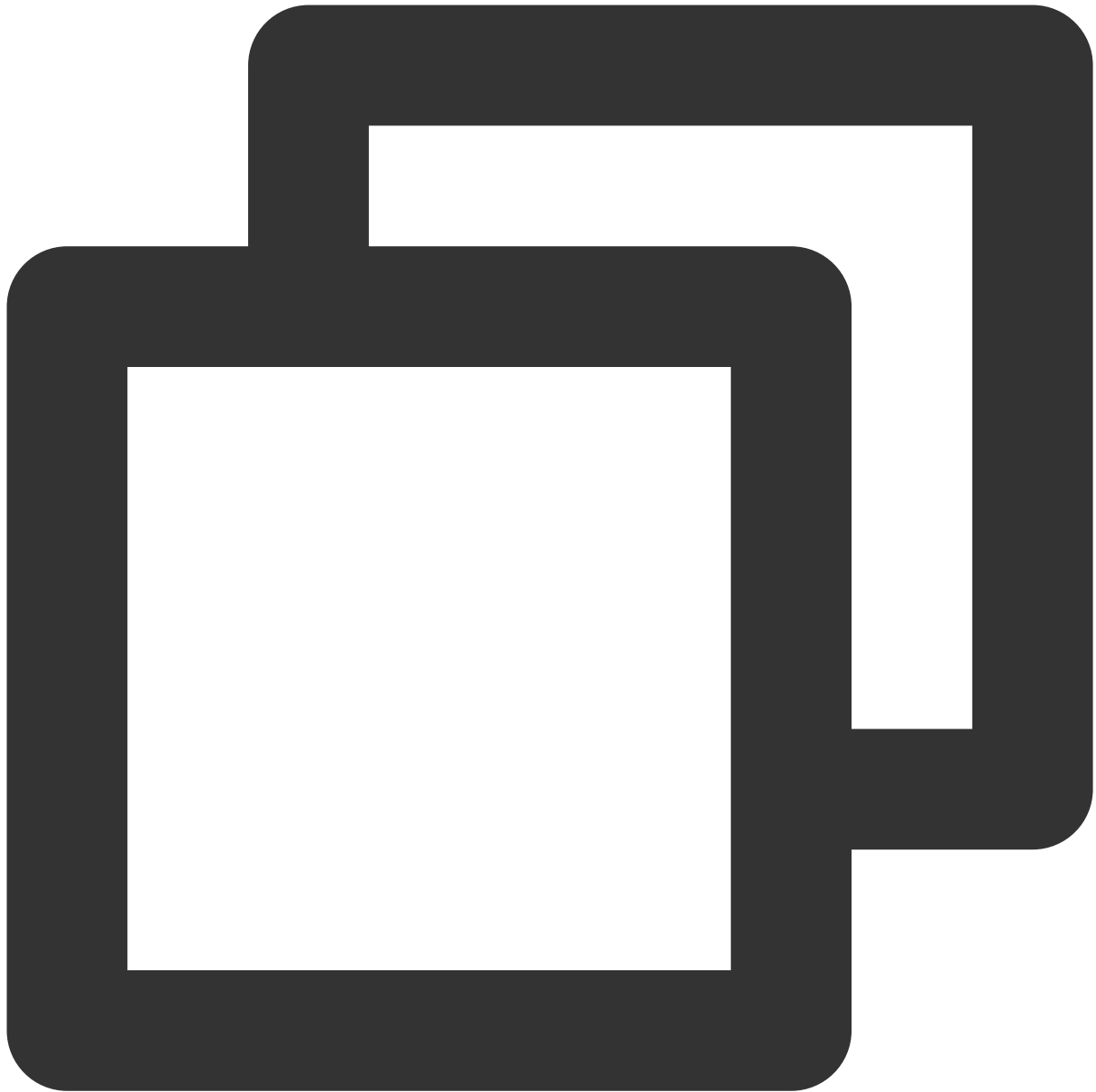


```
int MixRecordFile();
```

병합 취소

이 매개변수를 호출하면 병합 작업이 취소됩니다.

함수 프로토타입



```
int CancelMixRecordFile();
```

병합 파일 콜백

ITMG_MAIN_EVENT_TYPE_RECORD_MIX_COMPLETED: 미리보기 완료 콜백입니다. 미리보기 파일의 재생이 종료되거나 StopPreview API가 호출될 때 트리거됩니다.

콜백 매개변수

매개변수	유형	설명
------	----	----

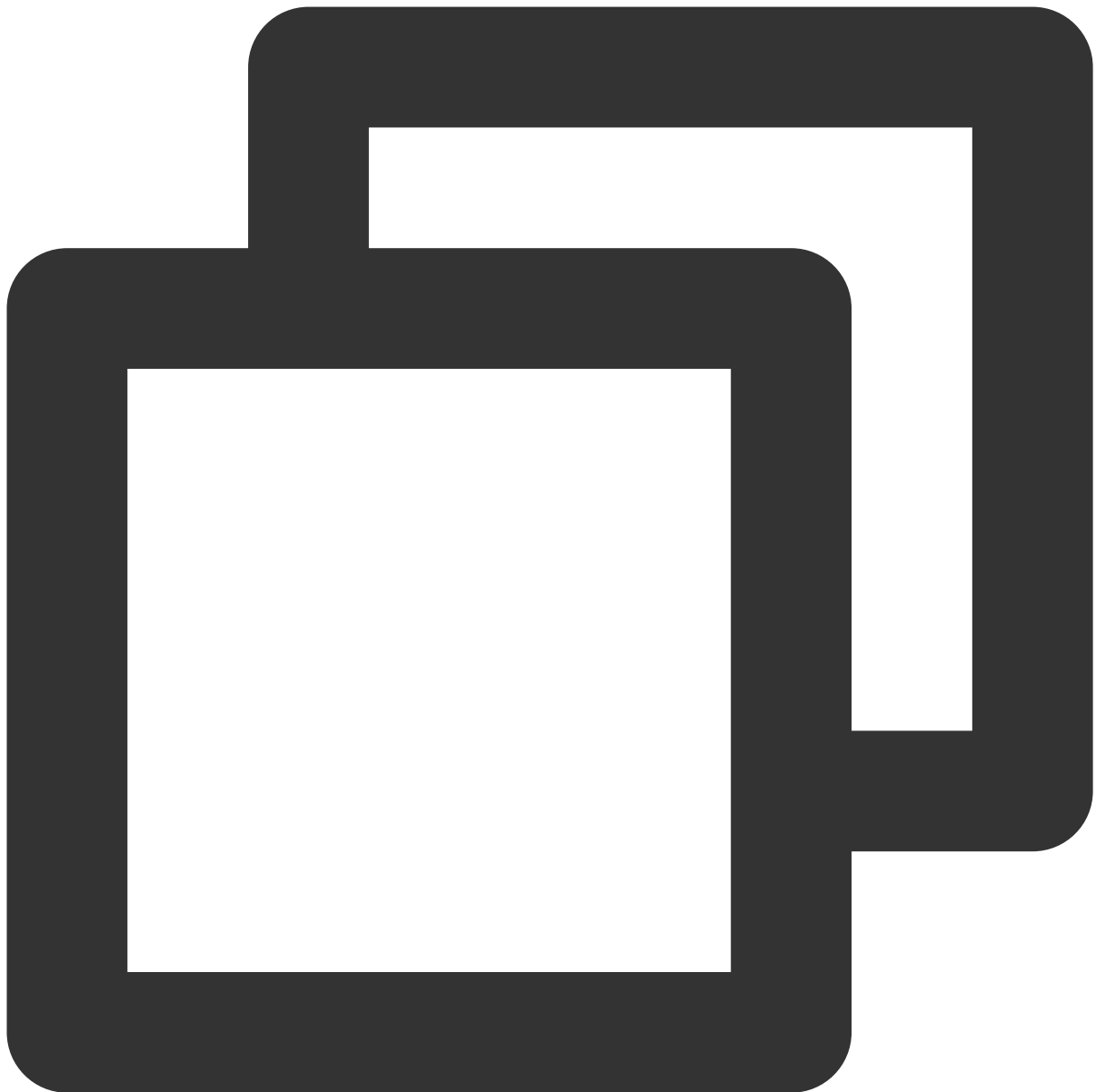
result	int	합성 결과. 0은 성공을 의미합니다.
filepath	String	대상 파일 경로, StartRecord API에서 전달된 dstFile입니다.
duration	String	녹음 파일의 길이, 단위는 밀리초입니다.

고급 설정

반주 크기 조정 설정

이 API를 호출하여 음성과 반주의 조정 비율을 설정합니다. 녹음이 완료된 후 조정할 수 있습니다.

함수 프로토타입



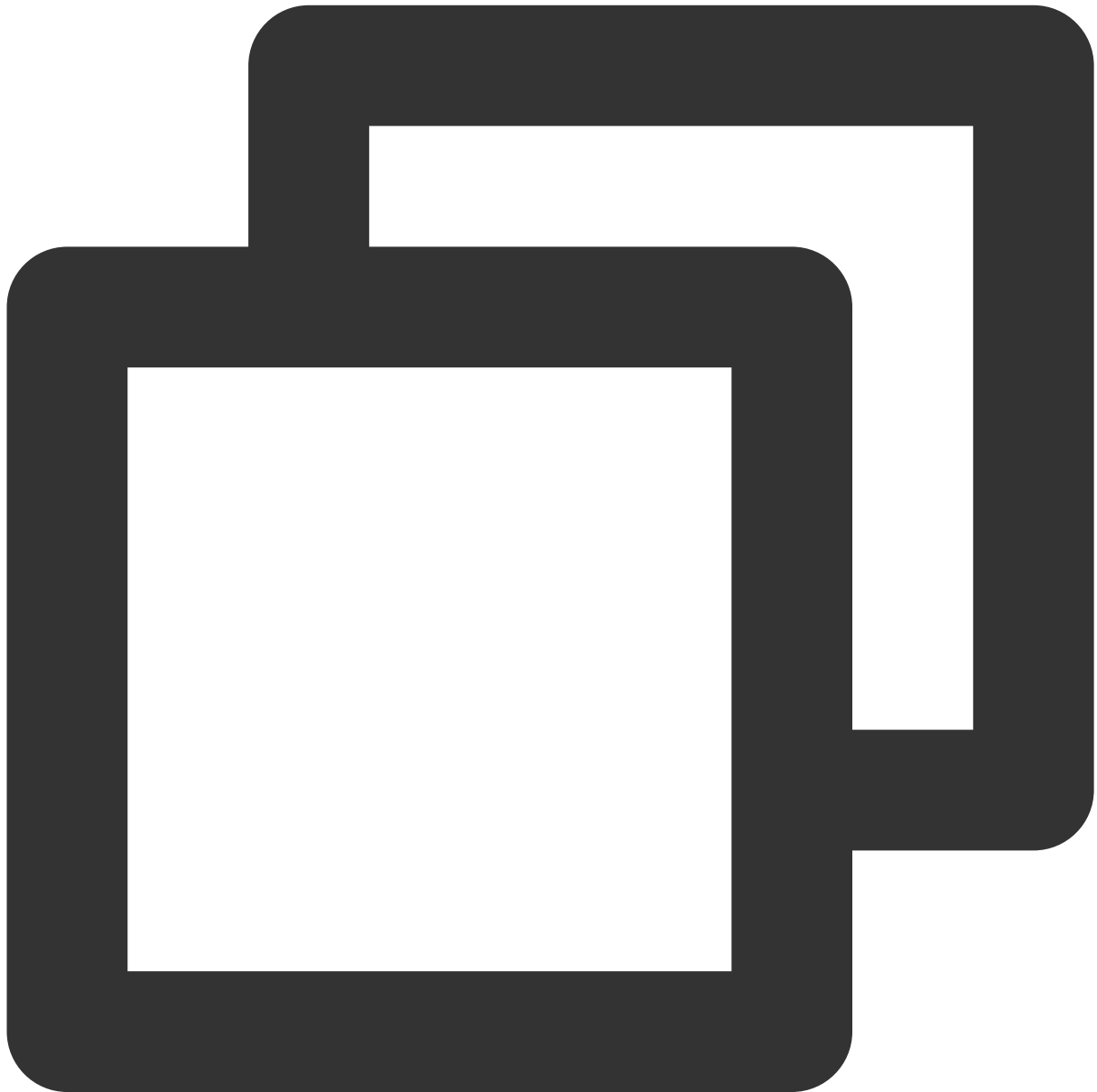
```
int SetMixWieghts(float mic, float acc)
```

매개변수	유형	설명
mic	float	음성 조정 비율입니다. 1.0은 기존 볼륨이며, 1.0보다 작으면 축소, 1.0보다 크면 확대됩니다. 범위는 0에서 2까지입니다.
acc	float	반주의 크기 조정 비율입니다. 1.0은 기존 볼륨이며, 1.0보다 작으면 축소, 1.0보다 크면 확대됩니다. 범위는 0에서 2까지입니다.

오프셋 설정

이 API를 호출하여 반주에 대비해 음성의 오프셋을 설정합니다. 일반적으로 음성과 박자가 맞지 않는 경우 조정하는데 사용됩니다. 녹음이 완료된 후 조정할 수 있습니다.

함수 프로토타입



```
int AdjustAudioTimeByMs(int time)
```

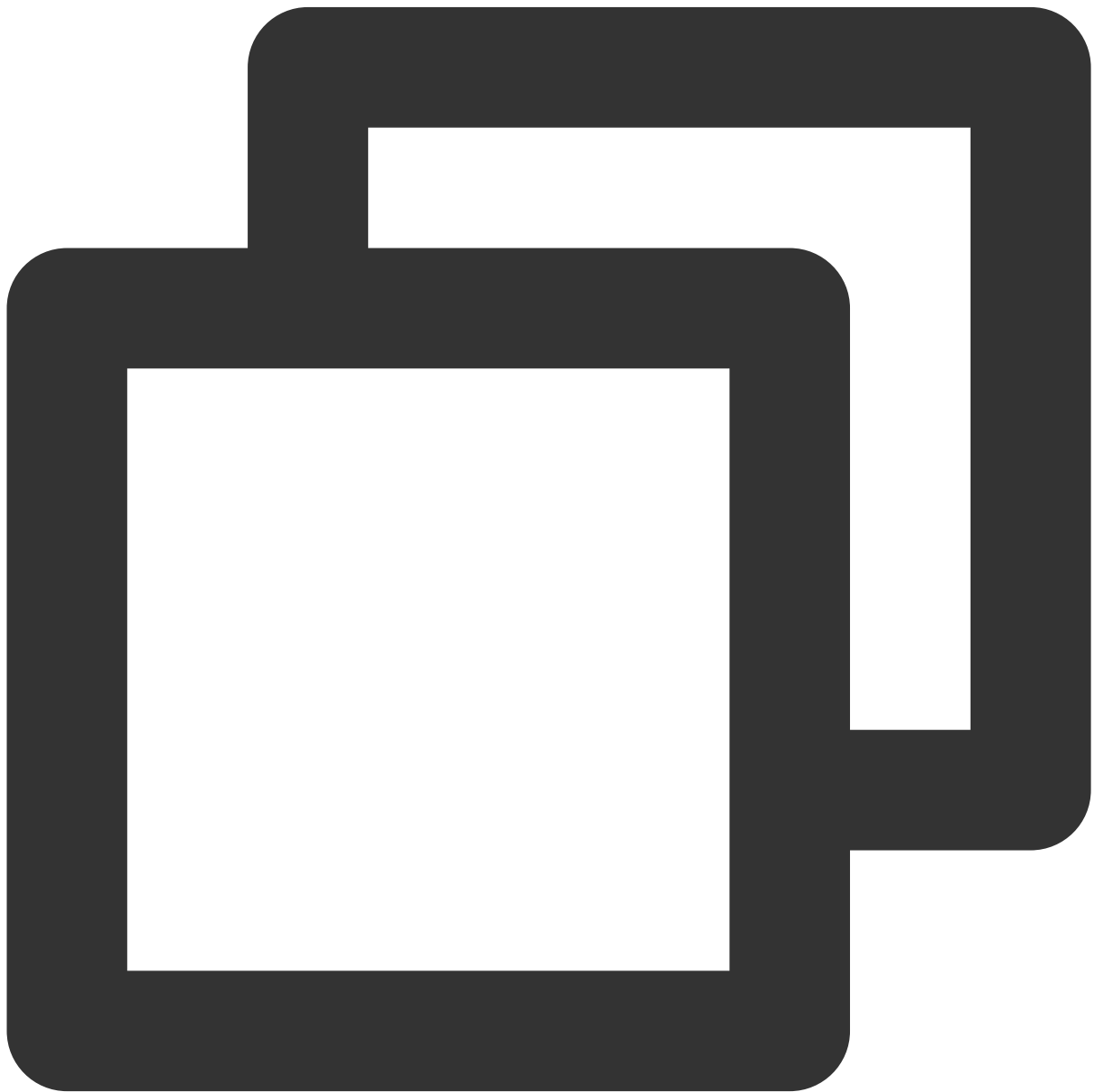
매개변수	유형	설명

time	int	음성의 반주에 대한 오프셋 시간이며, 단위는 밀리초입니다. 값이 0보다 큰 경우 뒤로 이동하고, 0보다 작은 경우 앞으로 이동합니다.
------	-----	--

음향 효과 설정

음향 효과 유형을 설정합니다. 노래방 음향 효과에 대한 내용은 [실시간 음성 채팅 음향 효과 문서](#)를 참고하십시오. 녹음이 완료된 후에 조정할 수 있으며, 녹음 중에도 사용할 수 있습니다.

함수 프로토타입



```
int SetRecordKaraokeType(int type)
```

매개변수	유형	설명
type	int	이 유형은 실시간 음향 효과에서 노래방 음향 효과 유형과 동일합니다. 자세한 내용은 노래방 음향 효과 를 참고하십시오.

네트워크 오디오 스트림 포워딩 라우팅

최종 업데이트 날짜: : 2024-01-18 16:00:04

본문은 개발자가 Game Multimedia Engine(GME)용 API를 쉽게 디버그하고 통합할 수 있도록 해주는 사용자 정의 오디오 포워딩 라우팅 사용에 대한 자세한 설명을 제공합니다.

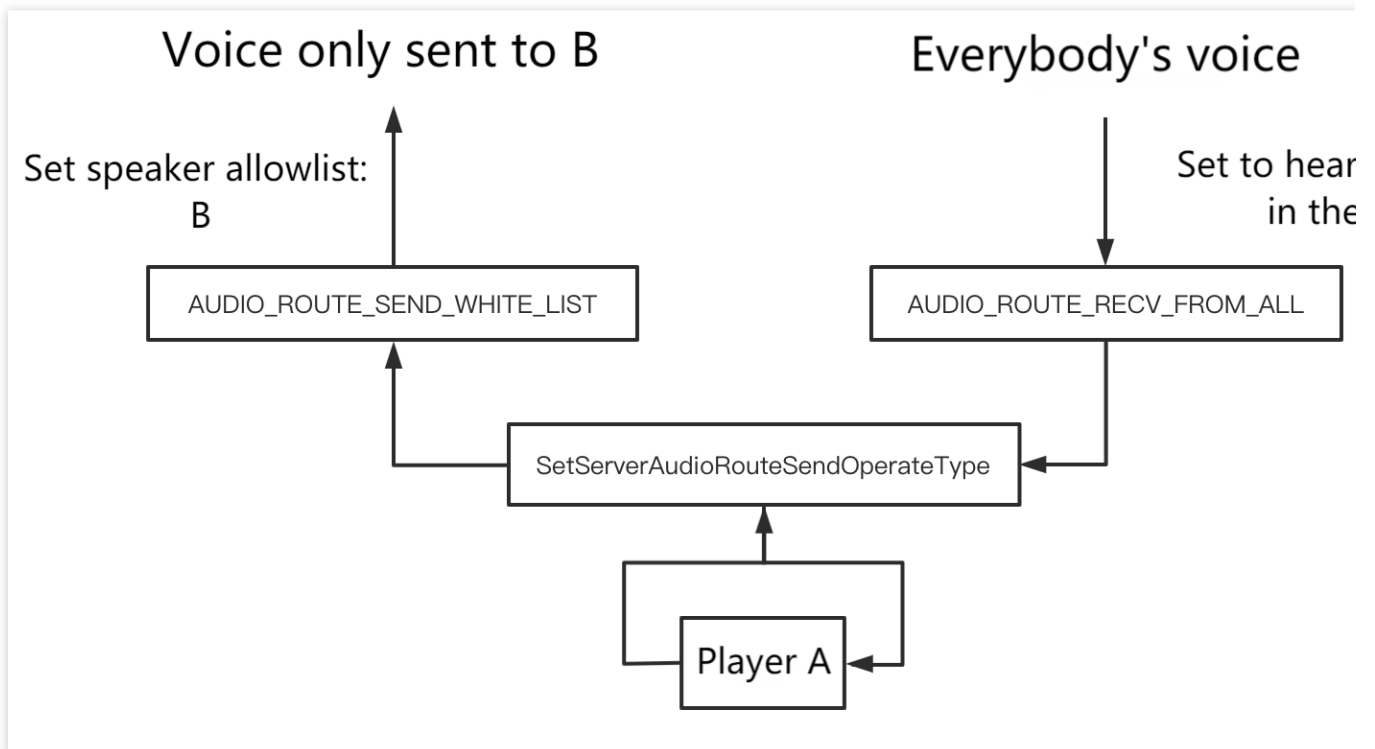
사용 사례

시나리오 설명: 두 명의 친구가 팀을 이루어 작은 팀을 구성한 후 그들은 세 명의 낯선 사람을 연결하여 큰 팀을 구성합니다. 그들은 큰 팀에서는 모든 구성원의 목소리를 듣고 싶어하지만 작은 팀에서는 서로만 이야기합니다.

이는 맞춤형 오디오 라우팅 기능으로 달성할 수 있습니다. 여기서는 5명의 사용자가 모두 동일한 음성 대화방에 입장합니다. 그런 다음 오디오 라우팅 API 설정을 통해 플레이어가 2인 팀 또는 전체 방에서 음성을 듣도록 설정할 수 있을 뿐만 아니라 2인 팀 또는 전체 방에서 들리도록 설정할 수 있습니다.

오디오 규칙 거리: `SetServerAudioRouteSendOperateType(AUDIO_ROUTE_SEND_WHITE_LIST,"2인 팀 list",ITMG_SERVER_AUDIO_ROUTE_RECV_TYPE,"2인 팀 list");`

이런 식으로 오디오는 목록에 있는 플레이어에게만 전송되고 3인 팀에서만 수신됩니다.



전제 조건

실시간 음성 채팅 서비스 **활성화 완료**: [서비스 활성화](#)를 참고하십시오.

GME SDK 통합 완료: 핵심 API 및 실시간 음성 채팅 API 통합을 포함합니다. 자세한 내용은 [Native SDK 빠른 통합](#), [Unity SDK 빠른 통합](#), [Unreal SDK 빠른 통합](#)을 참고하십시오.

GME의 음성 채팅 기능을 이용하여 성공적으로 음성방에 입장하였고, 마이크(EnableMic)와 스피커(EnableSpeaker)를 켜었습니다.

오디오 포워딩 라우팅에 액세스

오디오 포워딩 규칙 설정

이 API는 오디오 포워딩 규칙을 설정하는 데 사용되며 성공적인 방 입장 콜백에서 호출됩니다. 호출된 후 이 API는 이 방 입장에 적용되며 방을 나간 후에는 유효하지 않습니다.

주의사항 :

음소거 기능인 AddBlackList는 로컬에서 적용되며 사용자 지정 오디오 라우팅보다 우선 순위가 높습니다. 예를 들어 A가 SetServerAudioRouteSendOperateType을 통해 B의 음성만 들리도록 설정했지만 AddBlackList를 호출하여 B를 음소거하면 A는 B의 음성을 듣지 않습니다.

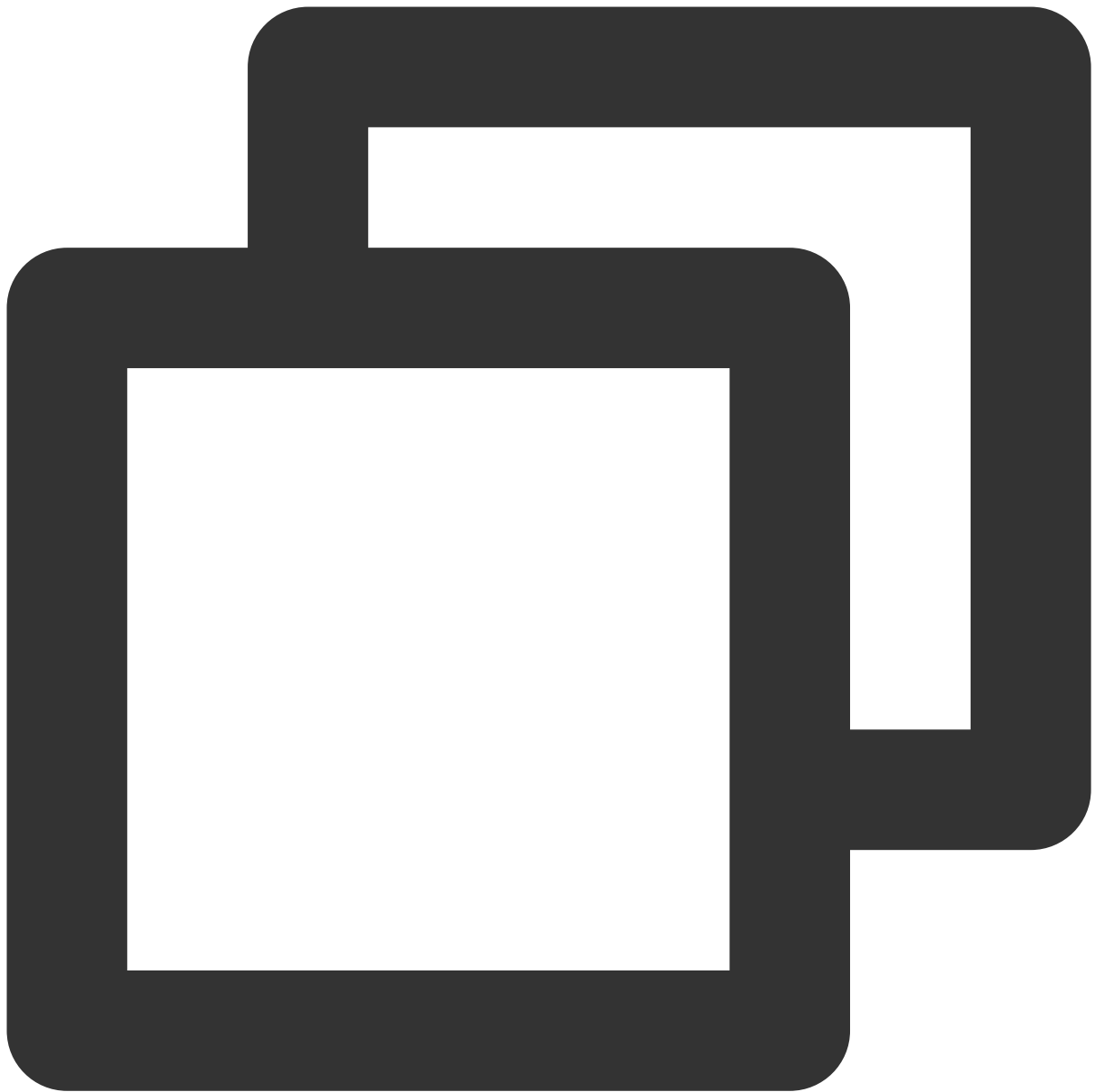
API 프로토타입

Unity

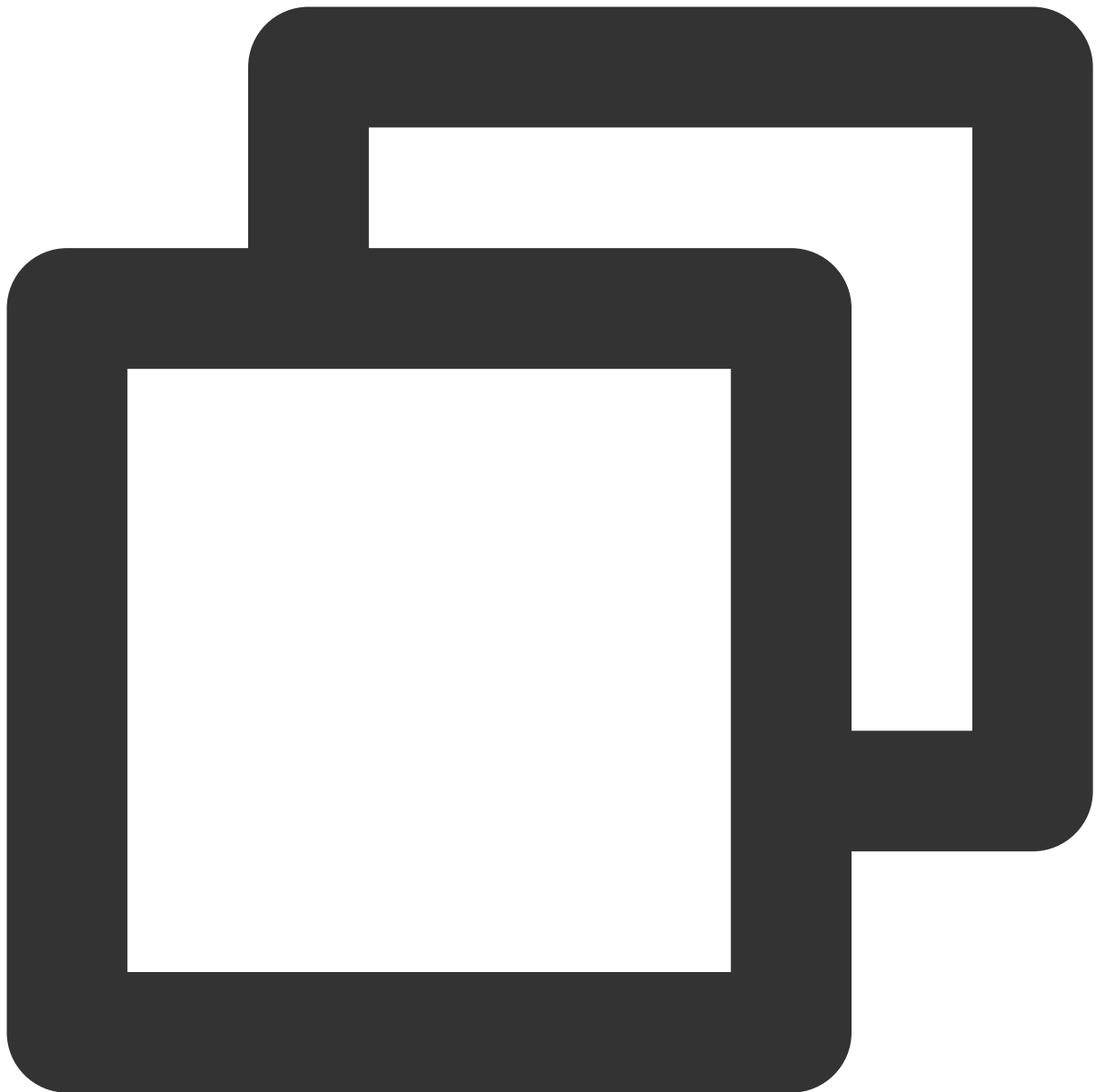
C++

Android

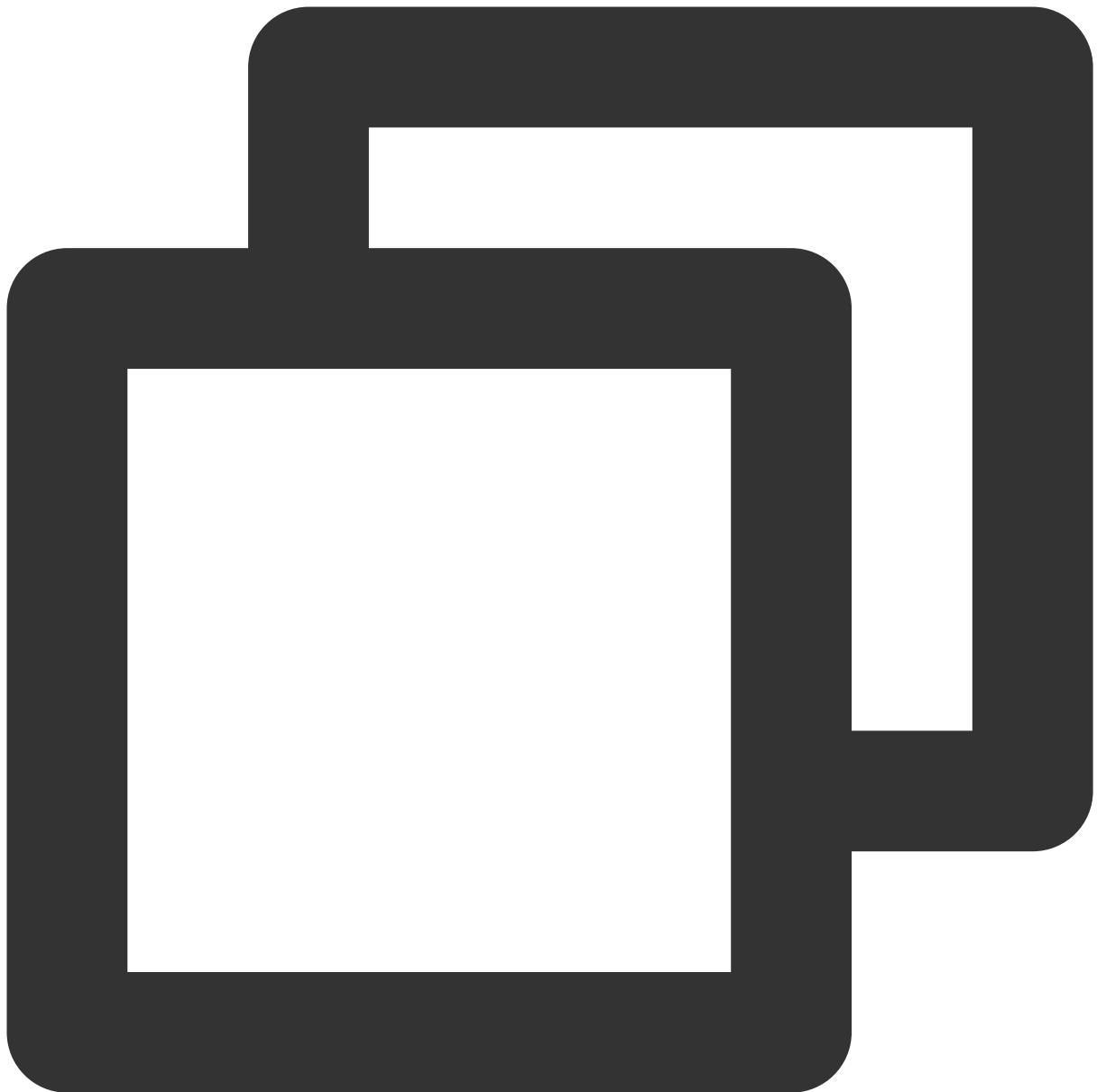
iOS



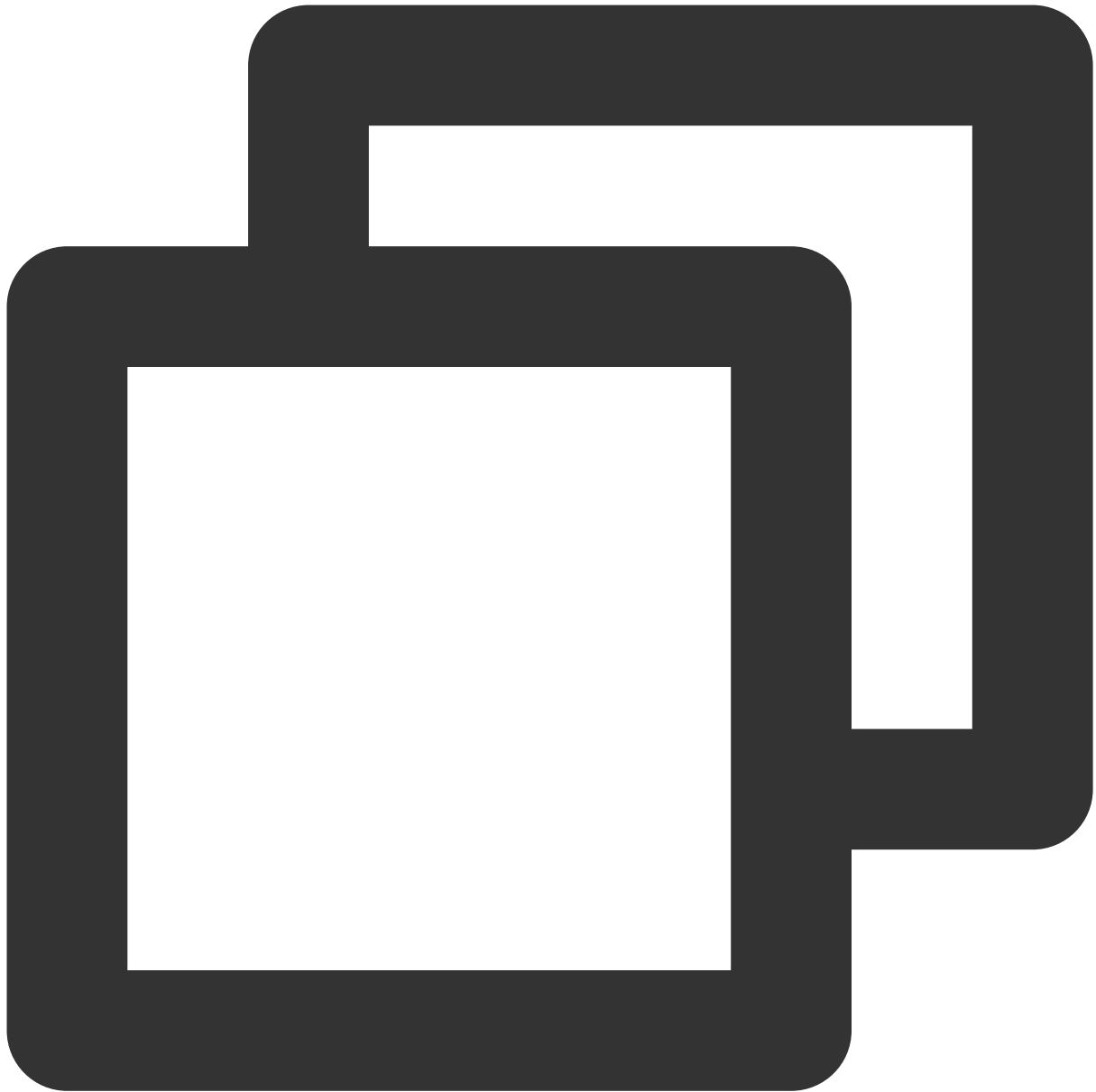
```
public abstract class ITMGRoom{  
    public abstract int SetServerAudioRouteSendOperateType(ITMG_SERVER_AUDIO_ROUTE_  
}
```



```
virtual int SetServerAudioRoute(ITMG_SERVER_AUDIO_ROUTE_SEND_TYPE SendType, const c
```



```
public abstract int SetServerAudioRoute(ITMGContext.ITMG_SERVER_AUDIO_ROUTE_SEND_TY
```



```
-(int) SetServerAudioRouteSendOperateType: (ITMG_SERVER_AUDIO_ROUTE_SEND_TYPE) Sendty
```

유형 설명

ITMG_SERVER_AUDIO_ROUTE_SEND_TYPE

오디오 전송 규칙을 설정합니다. 전송 규칙에 따라 전송 효과가 달라집니다.

수신 유형	효과
AUDIO_ROUTE_NOT_SEND_TO_ANYONE	로컬 오디오는 백엔드로 업스트림으로 전송되지만 백엔드는 아무에게도 포워딩하지 않습니다. 이는 로컬을 음소거하는 것

	과 같습니다. 현재 OpenIDForSend 매개변수는 유효하지 않습니다. null만 입력하면 됩니다.
AUDIO_ROUTE_SEND_TO_ALL	로컬 오디오는 모든 사람에게 업스트림으로 전송됩니다. 현재 OpenIDForSend 매개변수는 유효하지 않습니다. null만 입력하면 됩니다.
AUDIO_ROUTE_SEND_BLACK_LIST	로컬 오디오는 업스트림으로 전송되며 매개변수 OpenIDForSend가 제공하는 블록리스트에 있는 사람에게 포워딩되지 않습니다.
AUDIO_ROUTE_SEND_WHITE_LIST	로컬 오디오는 업스트림으로 전송되고 OpenIDForSend 매개변수가 제공하는 얼로우리스트에 있는 사람에게 포워딩됩니다.

설명 :

전달된 유형이 AUDIO_ROUTE_NOT_SEND_TO_ANYONE 또는 AUDIO_ROUTE_SEND_TO_ALL인 경우 OpenIDForSend 매개변수가 적용되지 않습니다. null만 입력하면 됩니다.

전달된 유형이 AUDIO_ROUTE_SEND_BLACK_LIST인 경우 OpenIDForSend 매개변수가 블록리스트가 됩니다. 최대 10명까지 지원됩니다.

전달된 유형이 AUDIO_ROUTE_SEND_WHITE_LIST인 경우 OpenIDForSend 매개변수가 얼로우리스트가 됩니다. 최대 10명까지 지원됩니다.

ITMG_SERVER_AUDIO_ROUTE_RECV_TYPE

오디오 수신 규칙을 설정합니다. 다른 수신 규칙에 대해 다른 수신 효과가 있습니다.

수신 유형	효과
AUDIO_ROUTE_NOT_RECV_FROM_ANYONE	로컬은 오디오를 수신하지 않으며 이는 방에서 스피커를 비활성화하는 것과 같습니다. 현재 OpenIDForSend 매개변수는 유효하지 않습니다. null만 입력하면 됩니다.
AUDIO_ROUTE_RECV_FROM_ALL	로컬은 모든 사람의 오디오를 수신합니다. 현재 OpenIDForSend 매개변수는 유효하지 않습니다. null만 입력하면 됩니다.
AUDIO_ROUTE_RECV_BLACK_LIST	로컬은 블록리스트에 있는 사람들의 오디오를 수신하지 않습니다. 블록리스트는 OpenIDForSend 매개변수로 제공됩니다.
AUDIO_ROUTE_RECV_WHITE_LIST	로컬은 얼로우리스트에 있는 사람들의 오디오만 수신합니다. 얼로우리스트는 OpenIDForSend 매개변수로 제공됩니다.

설명 :

전달된 유형이 AUDIO_ROUTE_NOT_RECV_FROM_ANYONE 또는 AUDIO_ROUTE_RECV_FROM_ALL인 경우 OpenIDForSend 매개변수가 적용되지 않습니다.

전달된 유형이 AUDIO_ROUTE_RECV_BLACK_LIST인 경우 OpenIDForSend 매개변수가 블랙리스트가 됩니다. 최대 10명까지 지원됩니다.

전달된 유형이 AUDIO_ROUTE_RECV_WHITE_LIST인 경우 OpenIDForSend 매개변수가 얼로우리스트가 됩니다. 최대 10명까지 지원됩니다.

반환된 값

반환된 QAV_OK 값은 호출이 성공했음을 나타냅니다.

콜백이 1004를 반환하면 매개변수가 잘못된 것입니다. 매개변수를 확인하십시오.

콜백이 1001을 반환하면 작업이 반복되었음을 의미합니다.

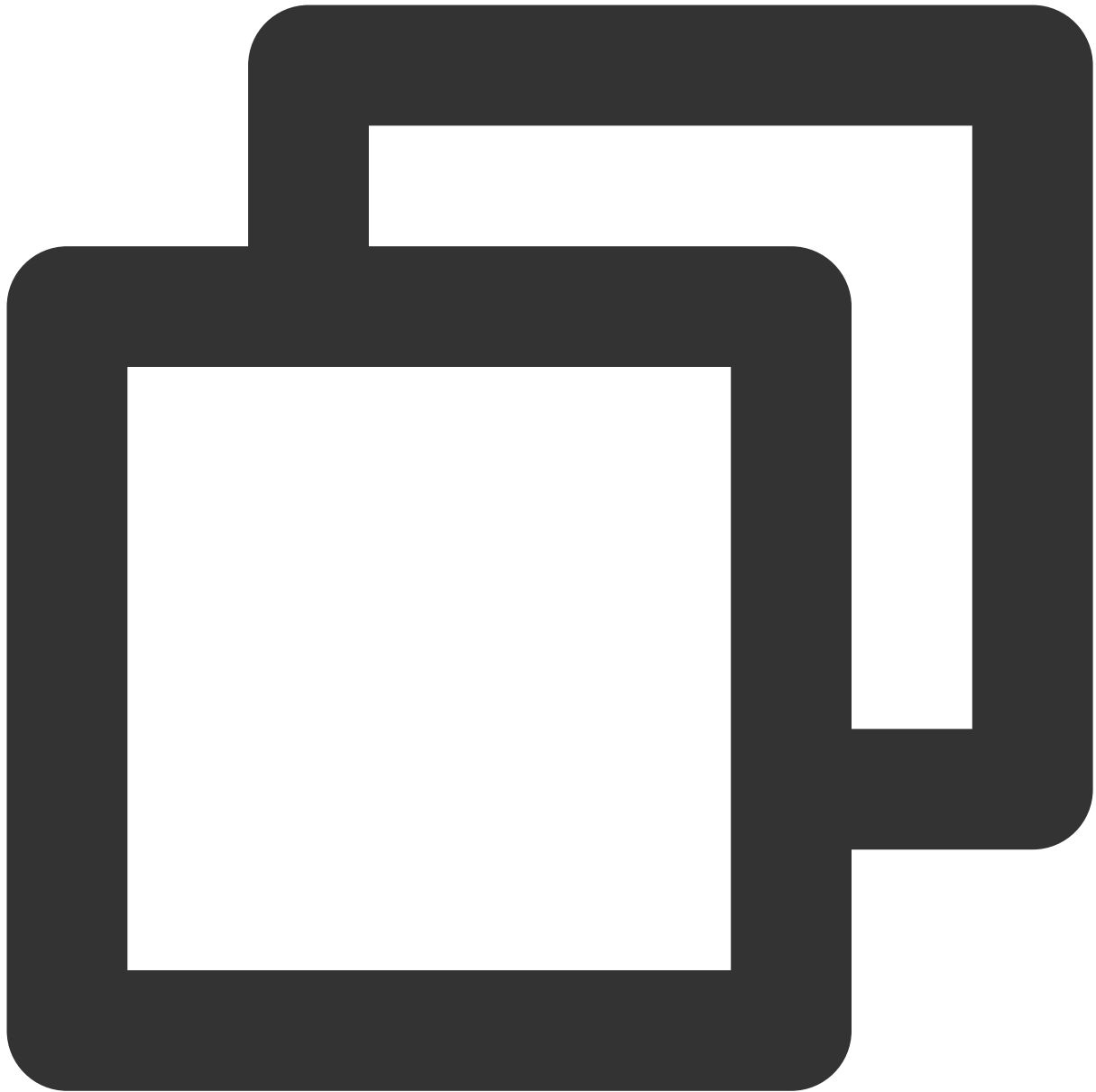
콜백이 1201을 반환하면 방이 존재하지 않는다는 의미입니다. 방 번호가 맞는지 확인하십시오.

콜백이 10001 및 1005를 반환하면 API를 다시 호출하십시오.

반환된 결과에 대한 자세한 설명은 [여러 코드](#)를 참고하십시오.

예시 코드

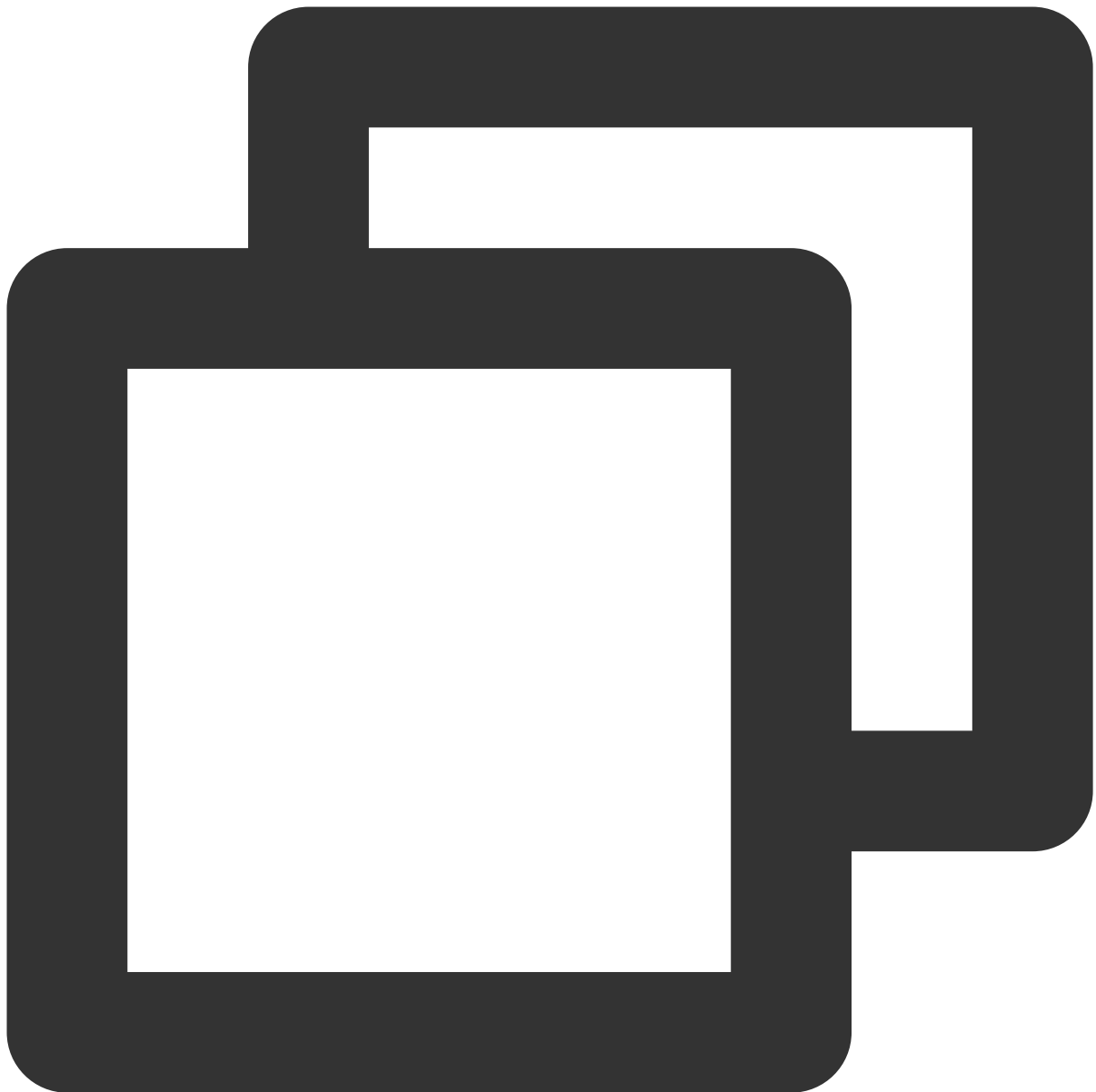
실행 문



```
@synthesize _sendListArray;
@synthesize _recvListArray;

int ret = [[ITMGContext GetInstance] GetRoom] SetServerAudioRouteSendOperateType:
if (ret != QAV_OK) {
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"audioroute 목록 업데이트"
    [alert show];
}
```

콜백



```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSString* log =[NSString stringWithFormat:@"OnEvent:%d,data:%@", (int)eventType
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_SERVER_AUDIO_ROUTE_EVENT:{
            {
                UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"audiorout
                    [alert show];
            }
        }
        default:
            break;
    }
```

```
}  
}
```

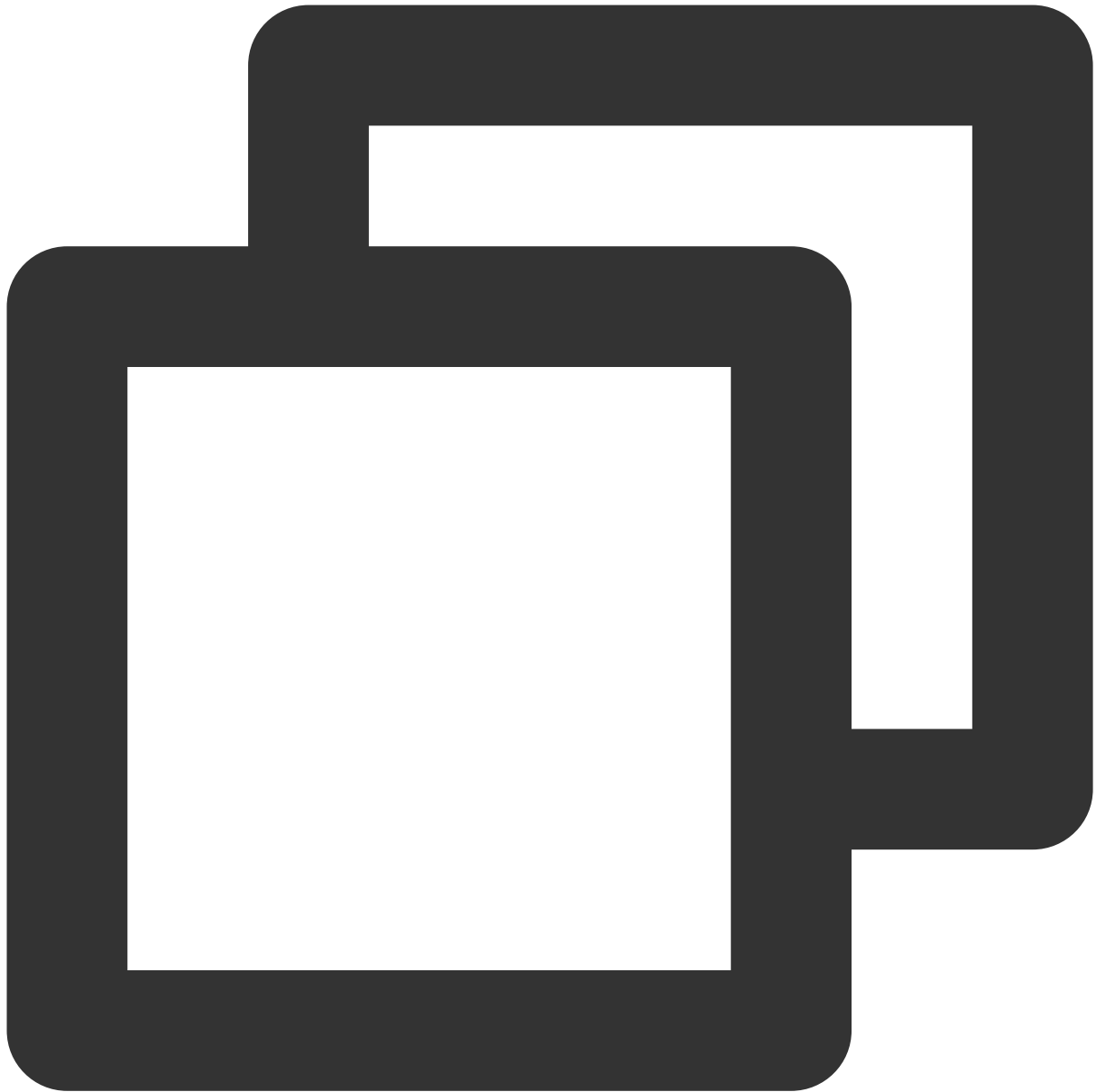
오디오 포워딩 규칙 가져오기

이 API는 오디오 포워딩 규칙을 가져오는 데 사용됩니다. 호출된 후 API는 규칙을 반환하고 전달된 배열 매개변수는 해당 규칙의 openId를 반환합니다.

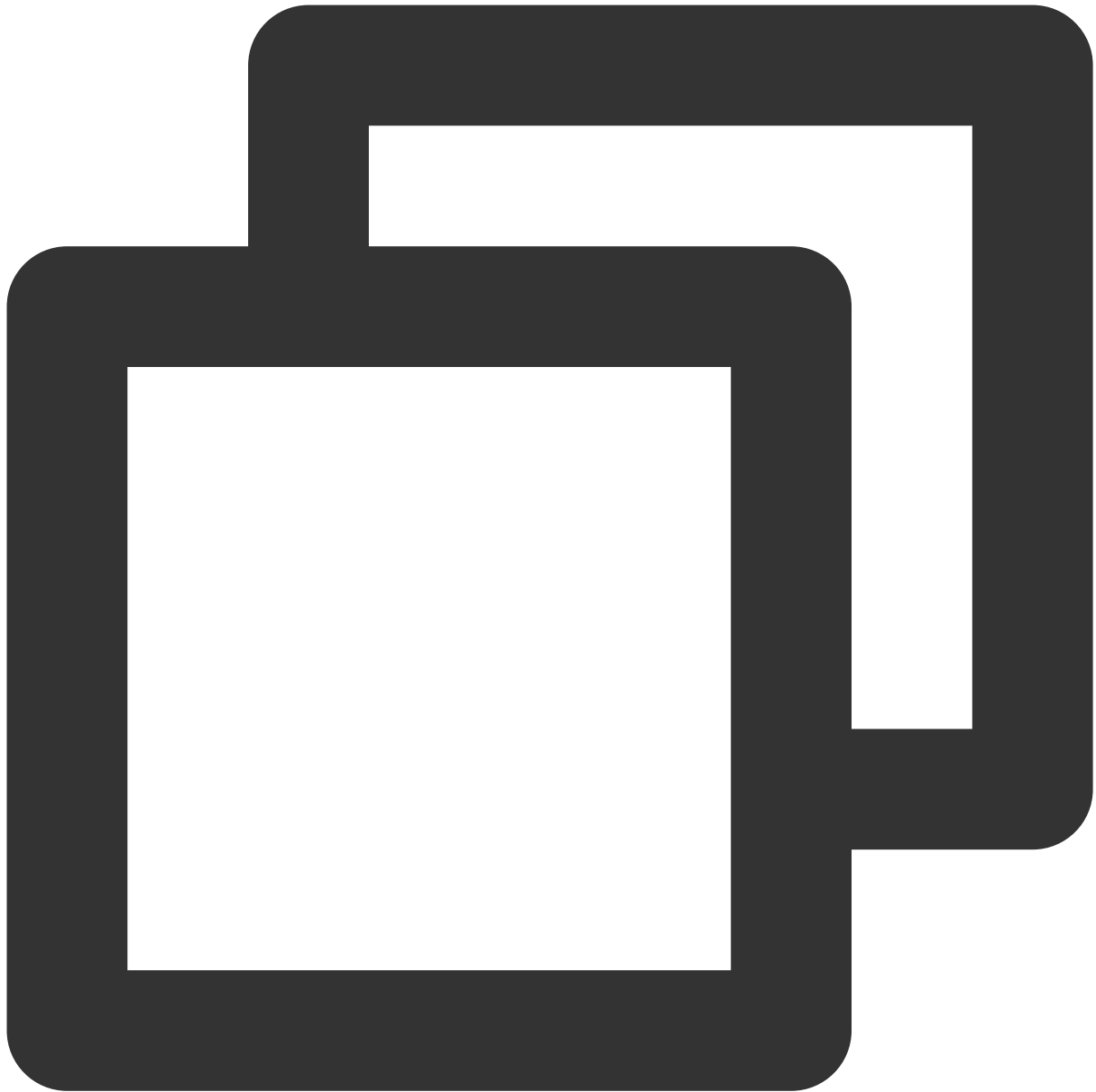
API 프로토타입

Unity

iOS



```
public abstract ITMG_SERVER_AUDIO_ROUTE_SEND_TYPE GetCurrentSendAudioRoute(List<str  
public abstract ITMG_SERVER_AUDIO_ROUTE_RECV_TYPE GetCurrentRecvAudioRoute(List<str
```



```

- (ITMG_SERVER_AUDIO_ROUTE_SEND_TYPE) GetCurrentSendAudioRoute: (NSMutableArray *) Ope
- (ITMG_SERVER_AUDIO_ROUTE_RECV_TYPE) GetCurrentRecvAudioRoute: (NSMutableArray *) Ope

```

반환 규칙

ITMG_SERVER_AUDIO_ROUTE_SEND_TYPE

수신 유형	효과
AUDIO_ROUTE_NOT_SEND_TO_ANYONE	로컬 오디오는 백엔드로 업스트림으로 전송되지만 백엔드는 아무에게도 포워딩하지 않습니다. 이는 로컬을 음소거하는 것

	과 같습니다.
AUDIO_ROUTE_SEND_TO_ALL	로컬 오디오는 모든 사람에게 업스트림으로 전송됩니다.
AUDIO_ROUTE_SEND_BLACK_LIST	로컬 오디오는 업스트림으로 전송되며 블랙리스트에 있는 사람에게 포워딩되지 않습니다.
AUDIO_ROUTE_SEND_WHITE_LIST	로컬 오디오는 업스트림으로 전송되어 얼로우리스트에 있는 사람들에게 포워딩됩니다.
AUDIO_ROUTE_RECV_INQUIRE_ERROR	가져오는 동안 오류가 발생했습니다. 로컬이 방에 들어왔는지, SDK가 초기화되었는지 확인하십시오.

ITMG_SERVER_AUDIO_ROUTE_RECV_TYPE

수신 유형	효과
AUDIO_ROUTE_NOT_RECV_FROM_ANYONE	로컬은 오디오를 수신하지 않으며 이는 방에서 스피커를 비활성화하는 것과 같습니다.
AUDIO_ROUTE_RECV_FROM_ALL	로컬은 모든 사람의 오디오를 수신합니다.
AUDIO_ROUTE_RECV_BLACK_LIST	로컬은 블랙리스트에 있는 사람들의 오디오를 수신하지 않습니다.
AUDIO_ROUTE_RECV_WHITE_LIST	로컬은 얼로우리스트에 있는 사람들의 오디오만 수신합니다.
AUDIO_ROUTE_RECV_INQUIRE_ERROR	가져오는 동안 오류가 발생했습니다. 로컬이 방에 들어왔는지, SDK가 초기화되었는지 확인하십시오.

주의사항 :

`SetServerAudioRouteSendOperateType` API에서 `AUDIO_ROUTE_RECV_INQUIRE_ERROR` 를 사용하지 마십시오.

사용자 지정 메시지 채널

최종 업데이트 날짜: : 2024-01-18 16:00:04

본문은 개발자가 게임 멀티미디어 엔진(GME)용 API를 쉽게 디버깅하고 통합할 수 있도록 하는 메시지와 함께 사용자 지정 오디오 패키지 사용에 대한 자세한 설명을 제공합니다.

사용 사례

메시지가 첨부된 사용자 지정 오디오 패키지 기능을 사용하면 GME 오디오 패키지에 사용자 지정 메시지를 같은 방에 있는 사용자에게 신호 방송으로 첨부할 수 있습니다.

전제 조건

실시간 음성 채팅 서비스 활성화 완료: [서비스 활성화](#)를 참고하십시오.

GME SDK 통합 완료: 핵심 API 및 실시간 음성 채팅 API 통합을 포함합니다. 자세한 내용은 [Native SDK 빠른 통합](#), [Unity SDK 빠른 통합](#), [Unreal SDK 빠른 통합](#)을 참고하십시오.

사용 제한

이 API를 호출하려면 두 가지 요구 사항을 충족해야 합니다. 방 유형은 **표준** 또는

HD(ITMG_ROOM_TYPE_STANDARD 또는 ITMG_ROOM_TYPE_HIGHQUALITY)여야 하며, 발신자의 마이크와 수신자의 스피커가 모두 켜져 있어야 합니다.

사용자 지정 메시지 기능 통합

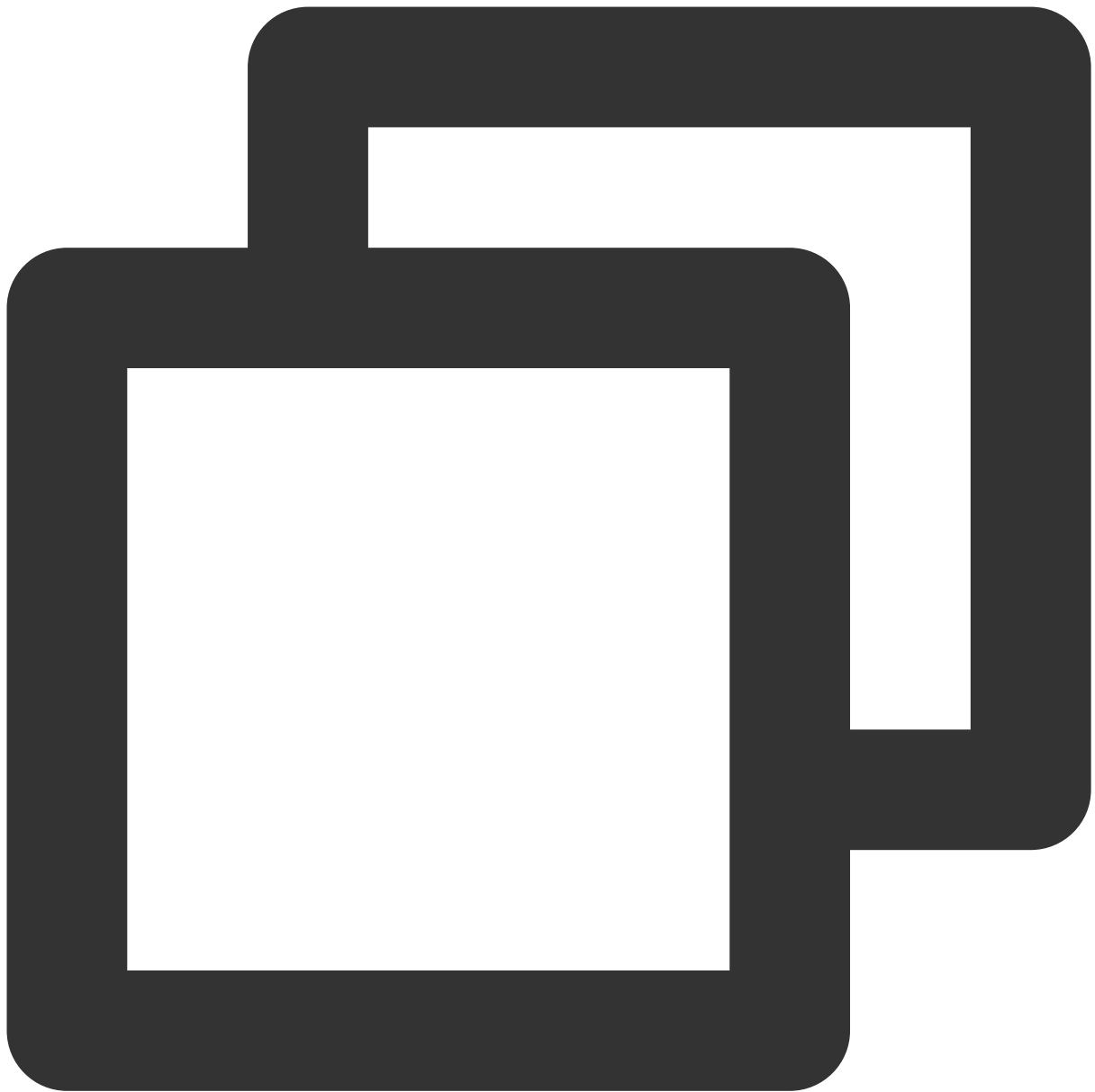
사용자 지정 메시지 전송

API 프로토타입

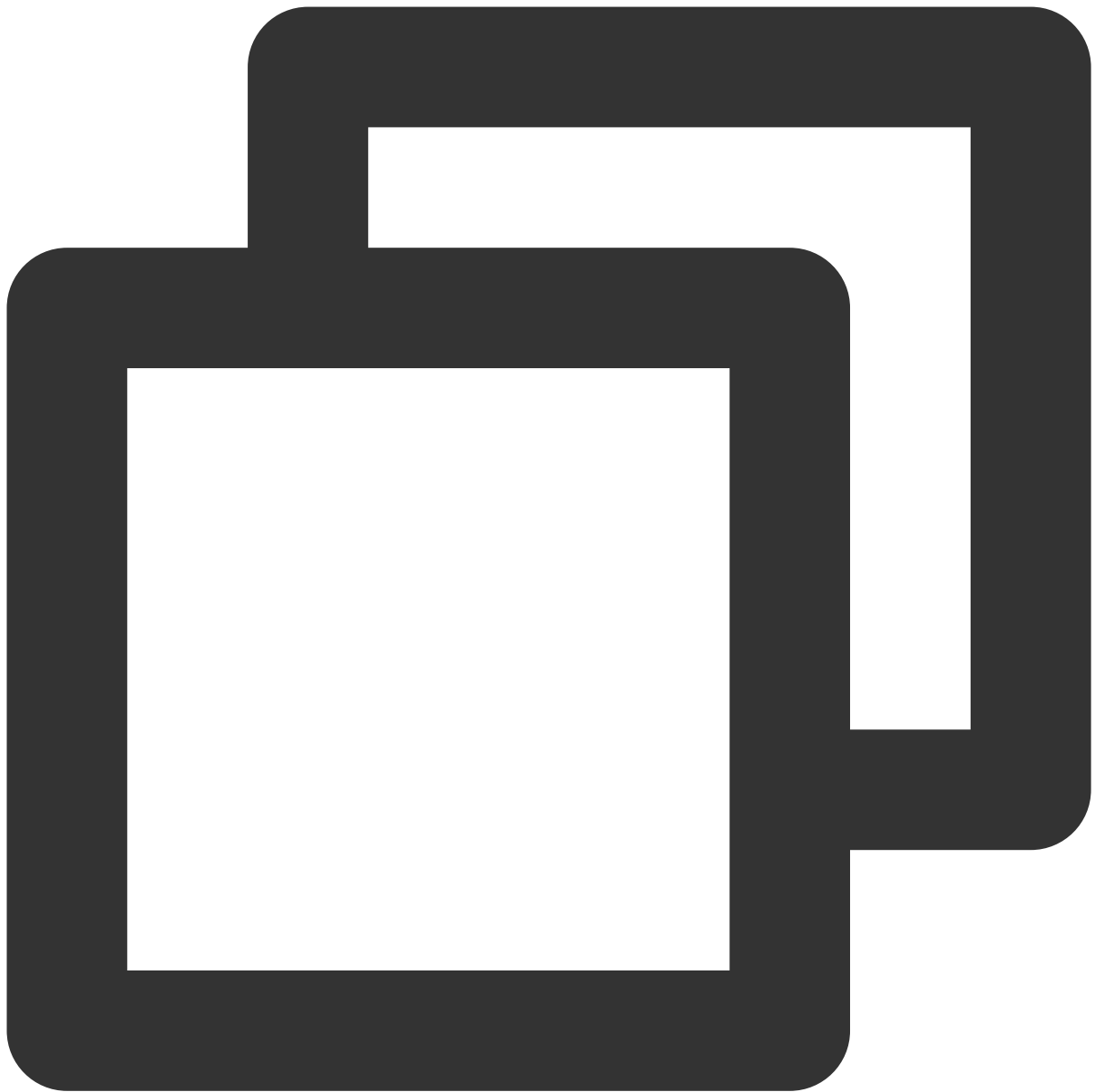
iOS

Android

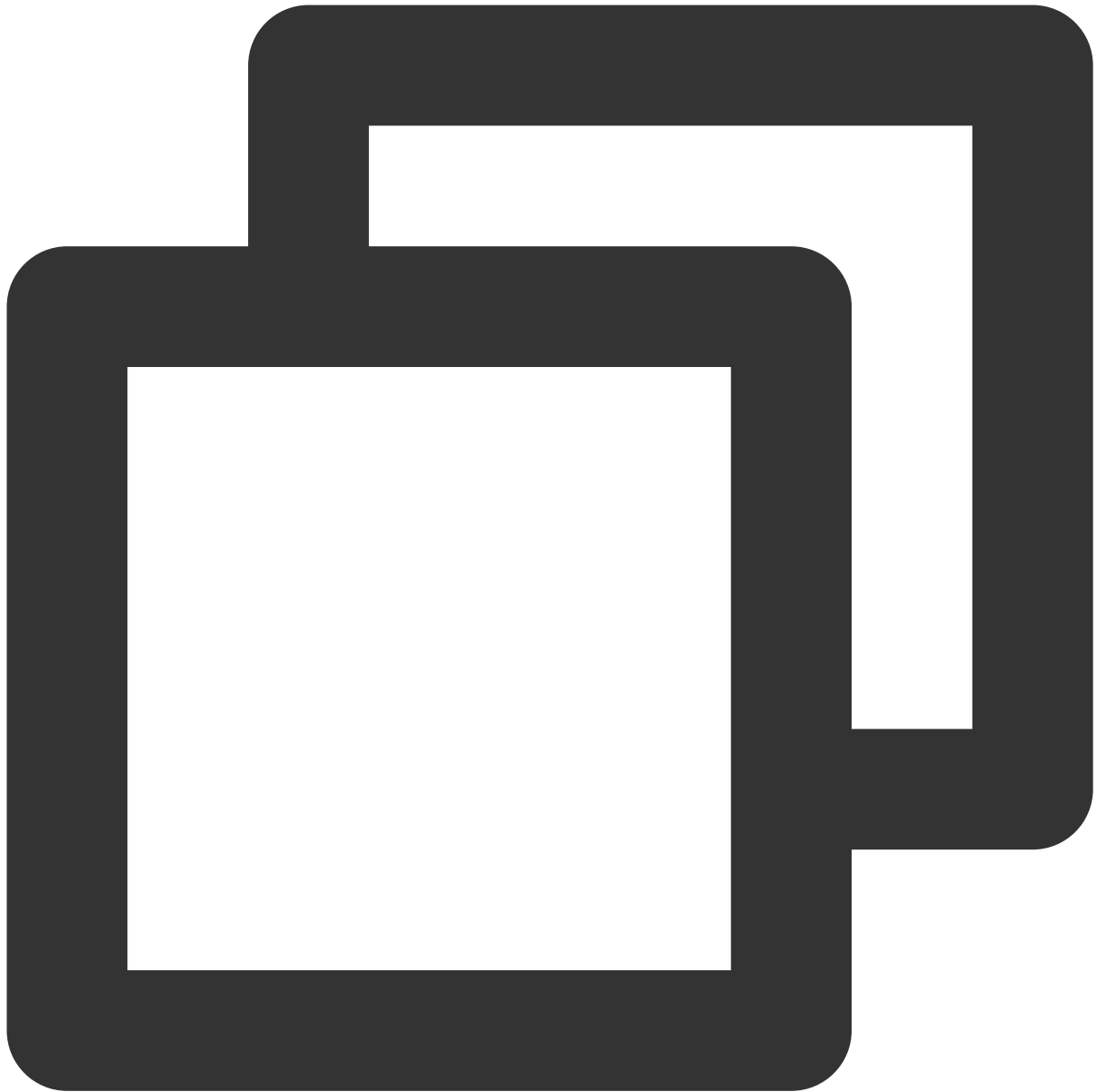
Unity



```
-(int) SendCustomData:(NSData *)data repeatCout:(int)reaptCout;
```



```
public abstract int SendCustomData(byte[] data,int repeatCout);
```



```
public abstract int SendCustomData(byte[] customdata,int repeatCout);
```

매개변수 설명

매개변수	유형	의미
data	NSData *, byte[]	전달할 메시지
reaptCout	int	반복 횟수, -1 입력 시 무제한 반복 전송

반환된 값

반환된 QAV_OK 값은 호출이 성공했음을 나타냅니다.

1004를 반환하는 콜백은 매개변수 오류를 나타냅니다. 매개변수를 확인하십시오. 1201은 존재하지 않는 방임을 나타냅니다. 방 번호를 확인하십시오.

자세한 에러 코드는 [에러 코드](#)를 참고하십시오.

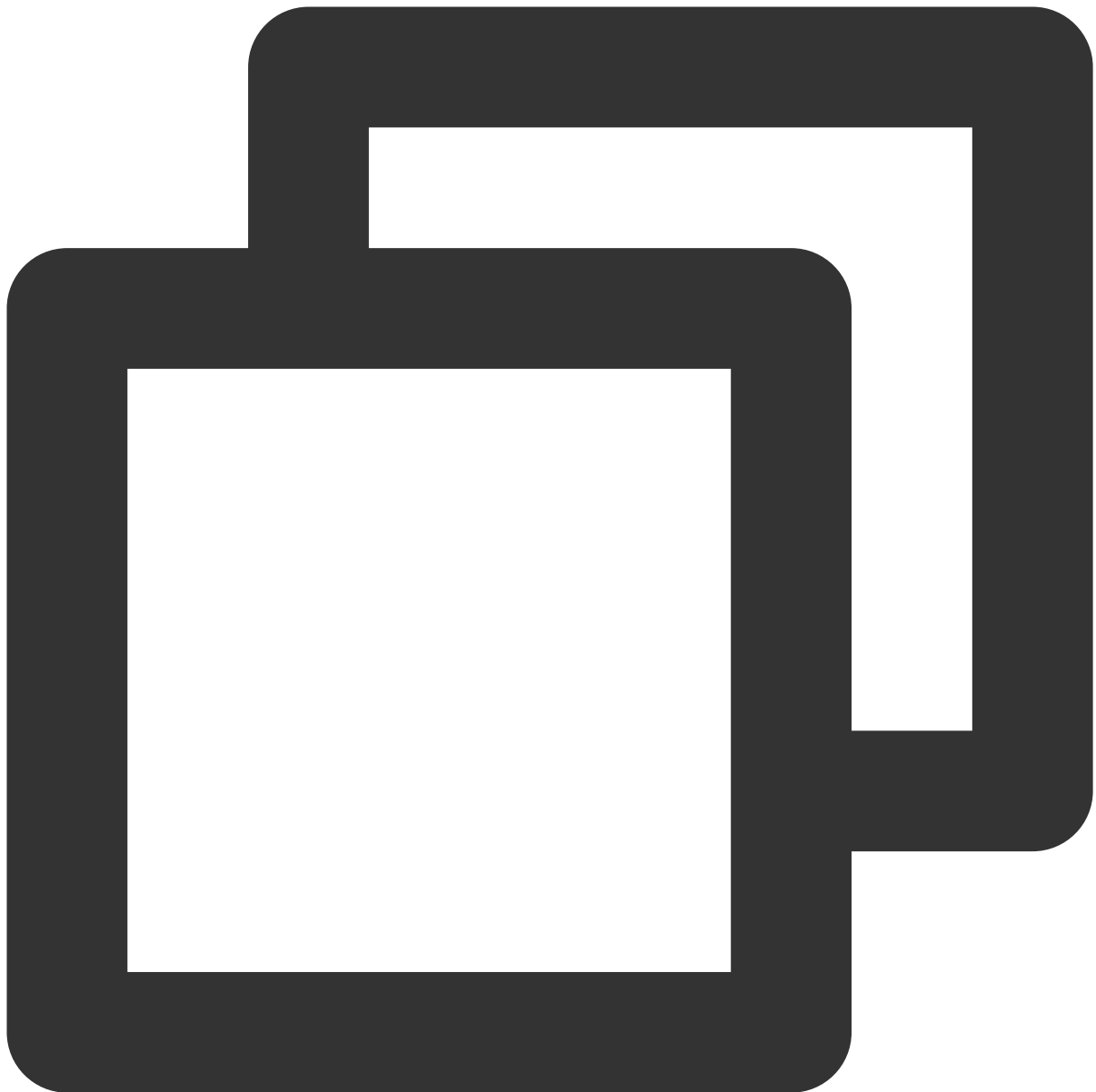
예시 코드

실행 문

iOS

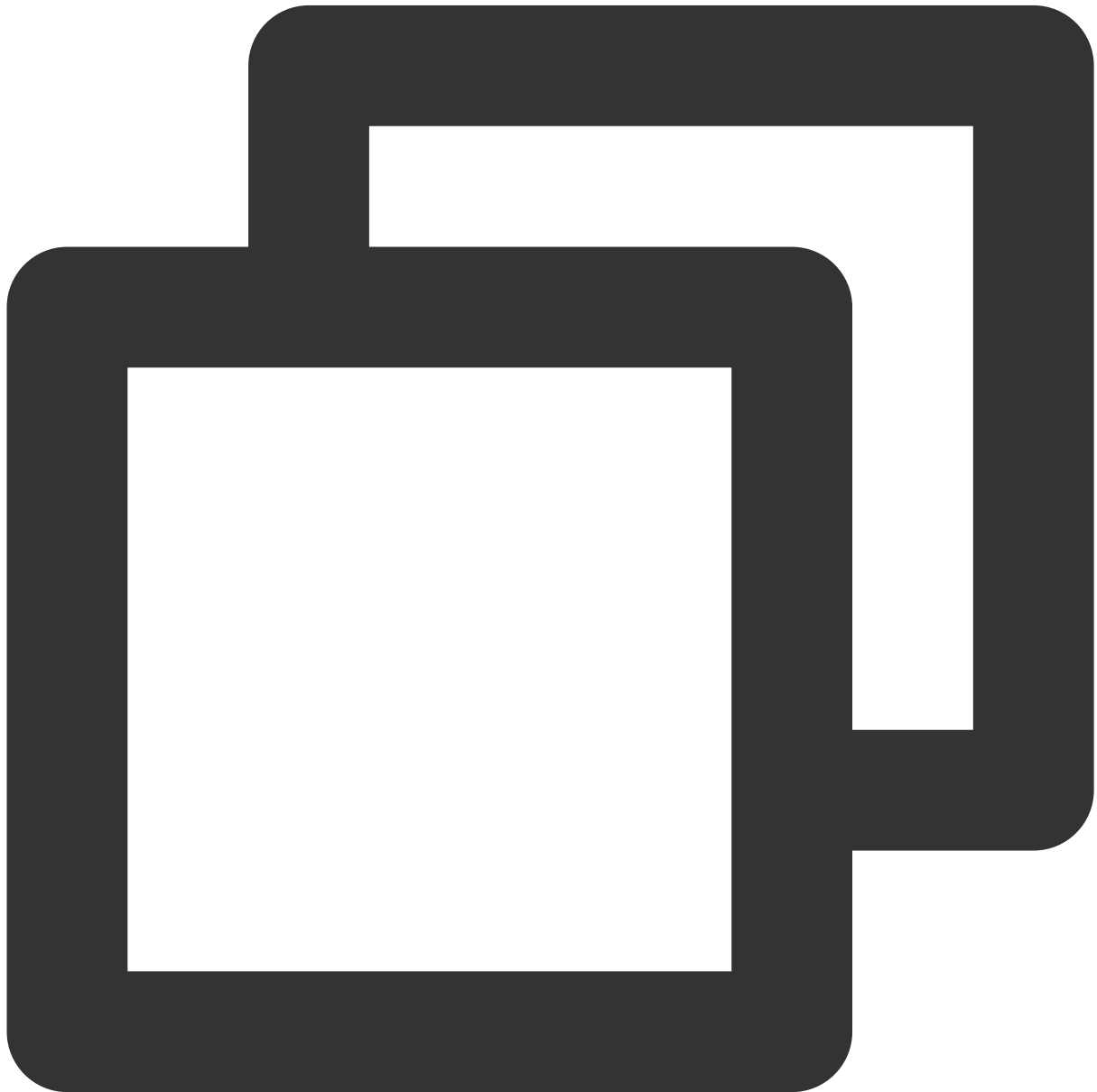
Android

Unity



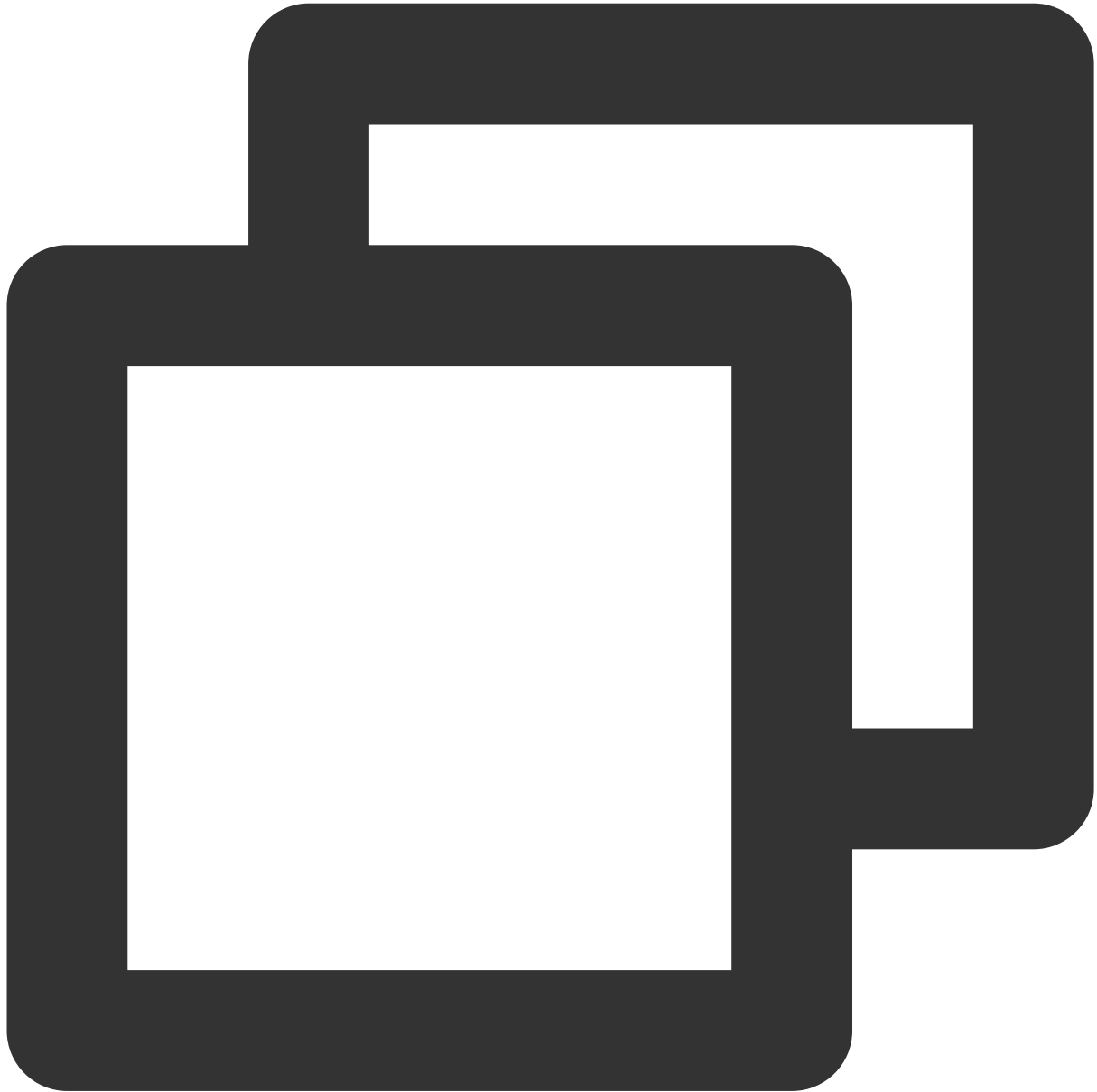
```
-(IBAction)SendCustData:(UIButton*)sender {
    int ret = 0;
    NSString *typeString;
    switch (sender.tag) {
        case 1:
            ret = [[[ITMGContext GetInstance] GetRoom] SendCustomData:[NSData dataW
            typeString = @"sendCustData";
            break;
        case 2:
            ret = [[[ITMGContext GetInstance] GetRoom] StopSendCustomData];
            typeString = @"recvCustData";
```

```
        break;
    default:
        break;
}
if(ret != 0) {
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"set fail" message
    [alert show];
}
}
```



```
String strData = mEditData.getText().toString();
```

```
String repeatCount = mEditRepeatCount.getText().toString();  
int nRet = ITMGContext.GetInstance(getActivity()).GetRoom().SendCustomData(strData.
```



```
InputField SendCustom_Count_InputField = transform.Find("inroomPanel/imPanel/SendCu  
InputField SendCustom_Data_InputField = transform.Find("inroomPanel/imPanel/SendCus  
  
transform.Find("inroomPanel/imPanel/SendCustom_Btn").GetComponent<Button>().onClick  
    {  
        string data = SendCustom_Data_InputField.text;  
        string str_count = SendCustom_Count_InputField.text;
```



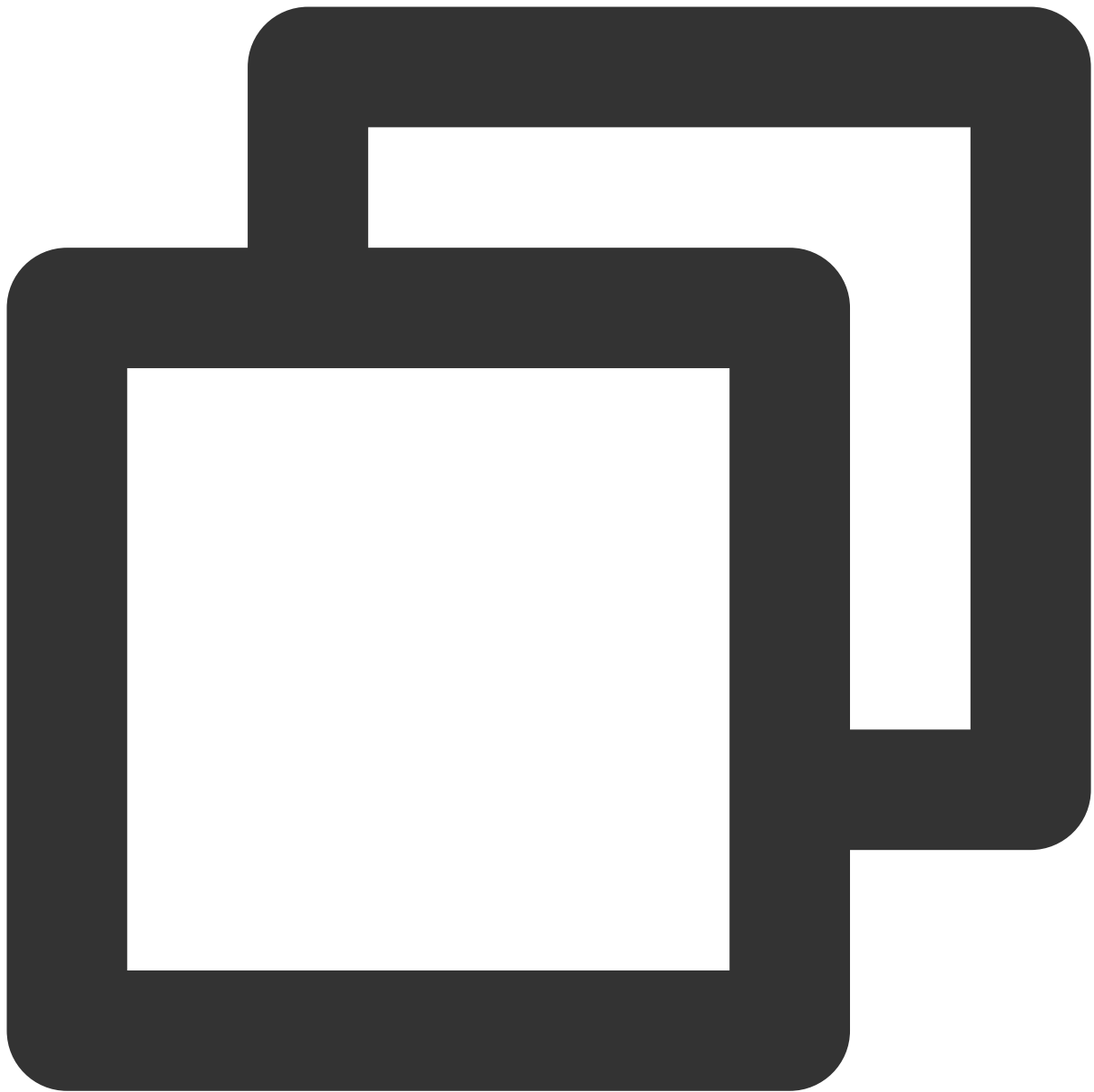
```
int count = 0;
if (int.TryParse(str_count, out count)) {
    Debug.Log(data+ count.ToString());
    byte[] byteData = Encoding.Default.GetBytes(data);
    int ret = ITMGContext.GetInstance().GetRoom().SendCustomData(byteData);
    if(ret != 0 ) {
        ShowWarning(string.Format("send customdata failed err:{0}",ret));
    }
}
});
}
```

콜백

iOS

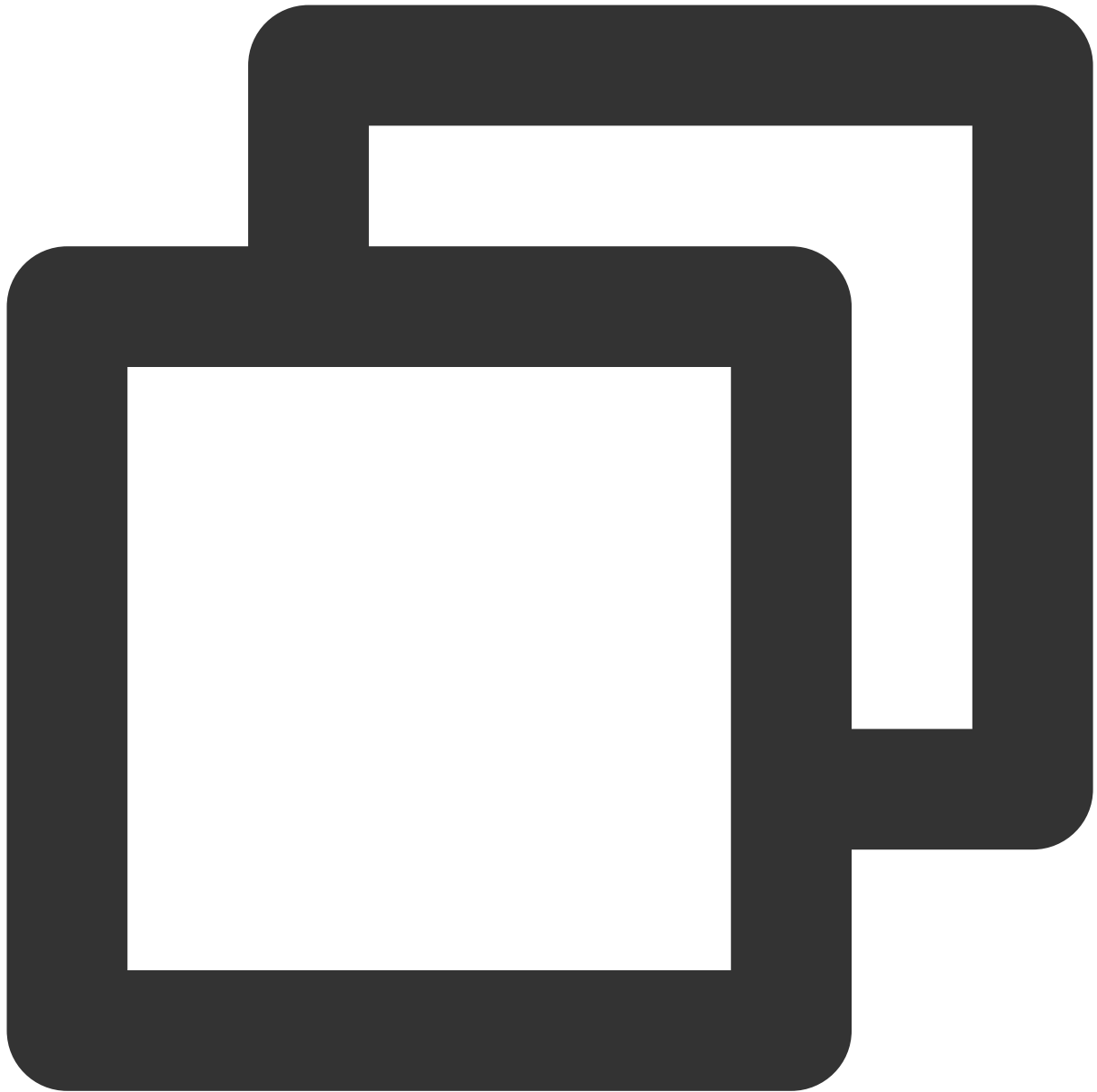
Android

Unity



```
-(IBAction)SendCustData:(UIButton*)sender {
    int ret = ret = [[[ITMGContext GetInstance] GetRoom] SendCustomData:[NSData dat

    if(ret != 0) {
        UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"set fail" message
        [alert show];
    }
}
```



```
if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_CUSTOMDATA_UPDATE == type
    int subtype = data.getIntExtra("sub_event",-1);
    if (subtype == 0) {
        String content = data.getStringExtra("content");
        String sender = data.getStringExtra("senderid");
        Toast.makeText(getActivity(), String.format("recv content =%s, from:%s", co
            sender), Toast.LENGTH_SHORT).show();
    }
}
```



```
void OnEvent(int eventType,int subEventType,string data)
{
    Debug.Log (data);
    switch (eventType) {
        case (int)ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_CUSTOMDATA_UPDATE:
            {
                if(subEventType == (int)ITMG_CUSTOMDATA_SUB_EVENT.ITMG_CUSTOMDATA_AV_SUB_E
                    _customData = JsonUtility.FromJson<CustomDataInfo>(data);
                ShowWarning(string.Format("recve customdata {0}  from {1}",_customData.
            }
        }
    }
```

```
break;  
}
```

사용자 지정 메시지 전송 중지

사용자 지정 메시지 전송을 중지하려면 이 API를 호출하십시오.

API 프로토타입

iOS

Android

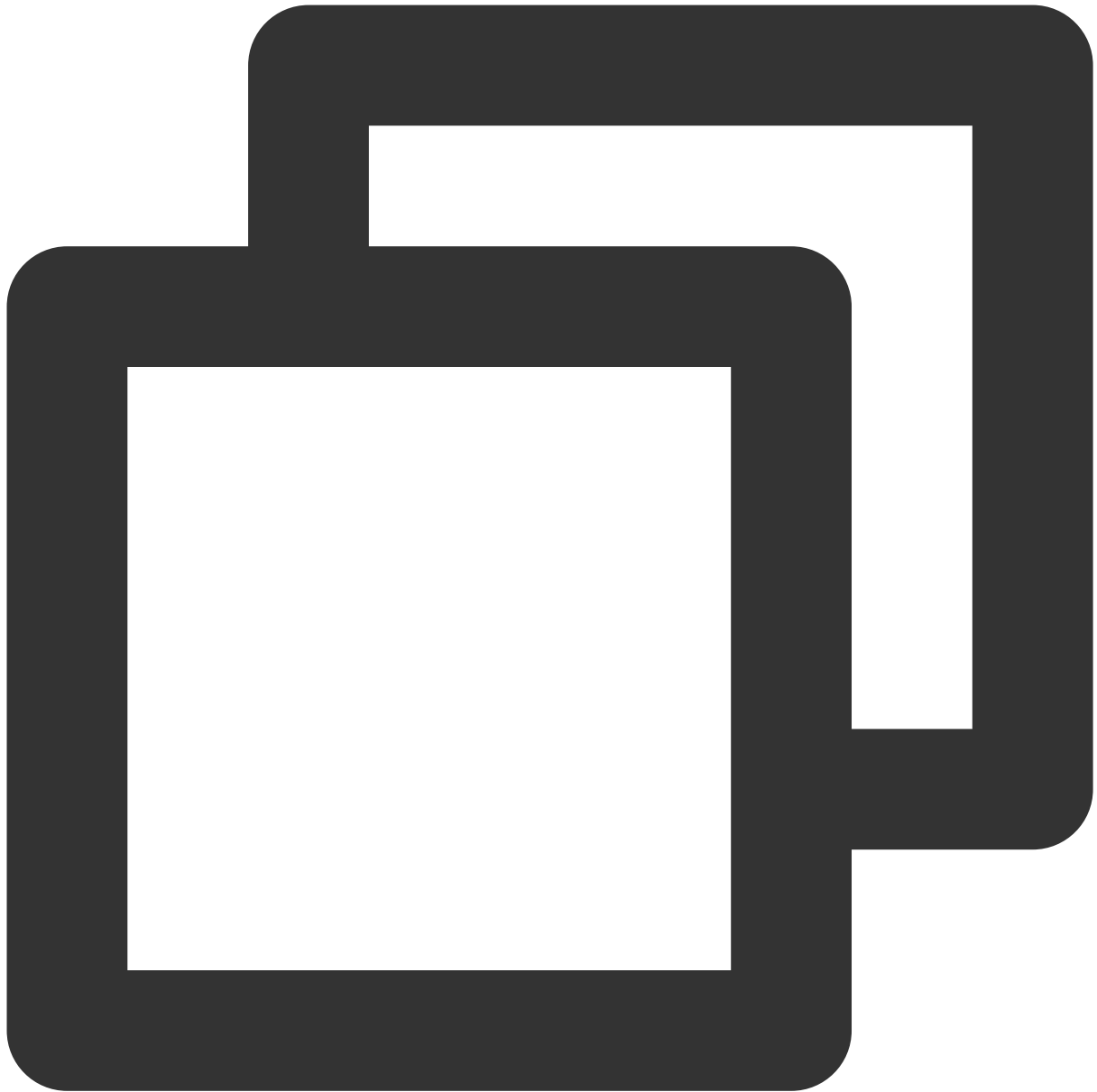
Unity



```
-(int) StopSendCustomData;
```



```
public abstract int StopSendCustomData();
```



```
public abstract int StopSendCustomData();
```

반환된 값

API가 오류 코드 1003을 반환하면 SDK에서 StopSendCustomData를 수행하는 중이므로 다시 호출할 필요가 없습니다.

회사 방화벽 제한을 해결하는 방법

최종 업데이트 날짜: : 2024-01-18 16:00:04

회사에 공중망 액세스에 대한 제한이 있는 경우 방화벽 열로우리스트를 구성해야 액세스할 수 있습니다. 다음은 관련 규칙입니다.

클라이언트 Native SDK(v2.2 이상)

방화벽 포트:

포트 유형	열로우리스트 항목
TCP 포트	443
UDP 포트	8000

도메인 이름 열로우리스트:



```
tcloud.tim.qq.com  
gmeconf.qcloud.com  
yun.tim.qq.com  
gmeosconf.qcloud.com  
sg.global.gme.qcloud.com
```

주의사항 :

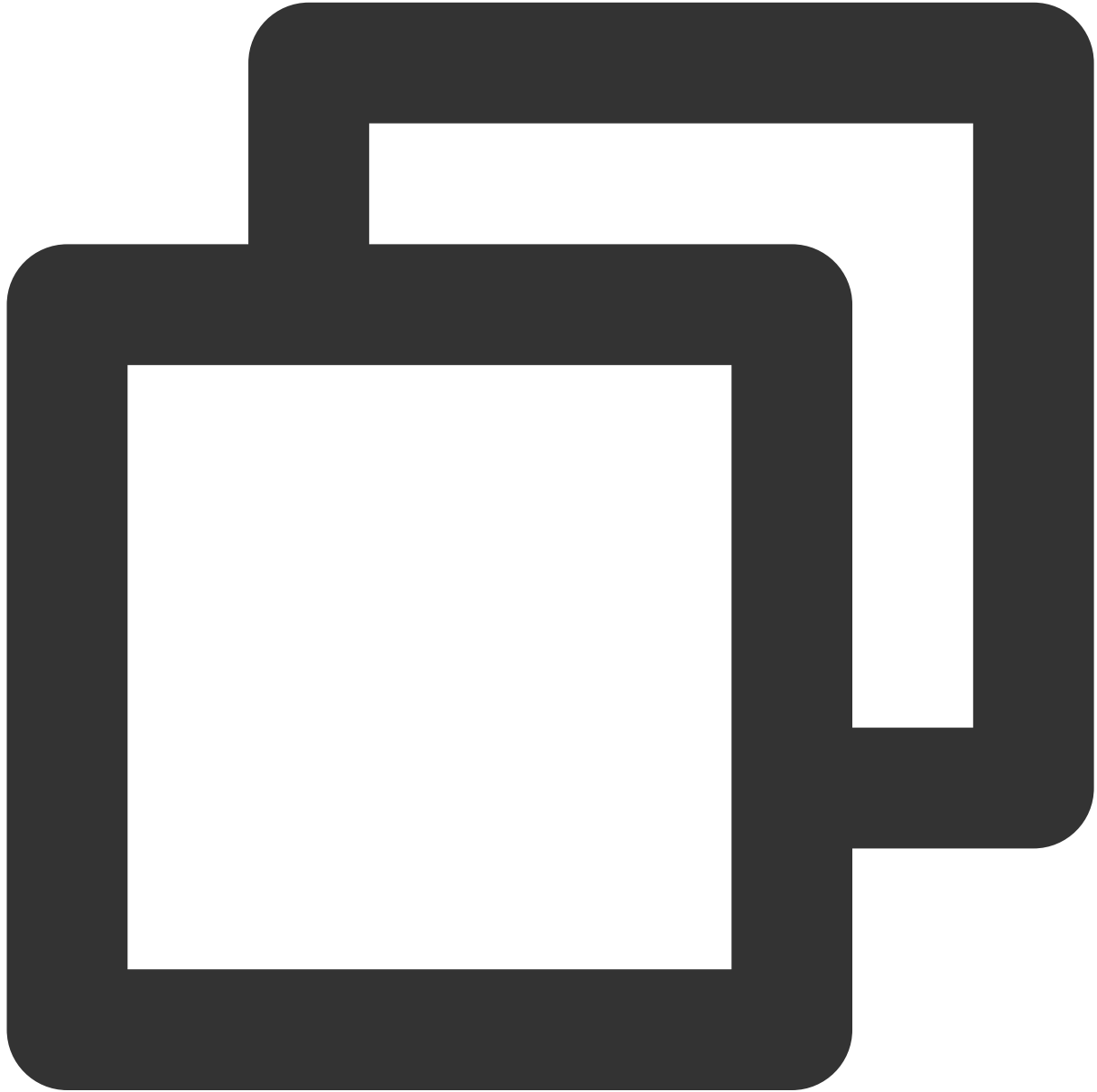
Tencent Cloud 서버 IP 주소는 동적으로 업데이트되므로 고정 IP 목록을 제공할 수 없습니다.

Windows XP에서 GME SDK를 사용하려면 방화벽 알로우리스트에 다음 항목을 추가해야 합니다.

방화벽 포트:

포트 유형	얼로우리스트 항목
TCP 포트	15000

도메인 이름 얼로우리스트:

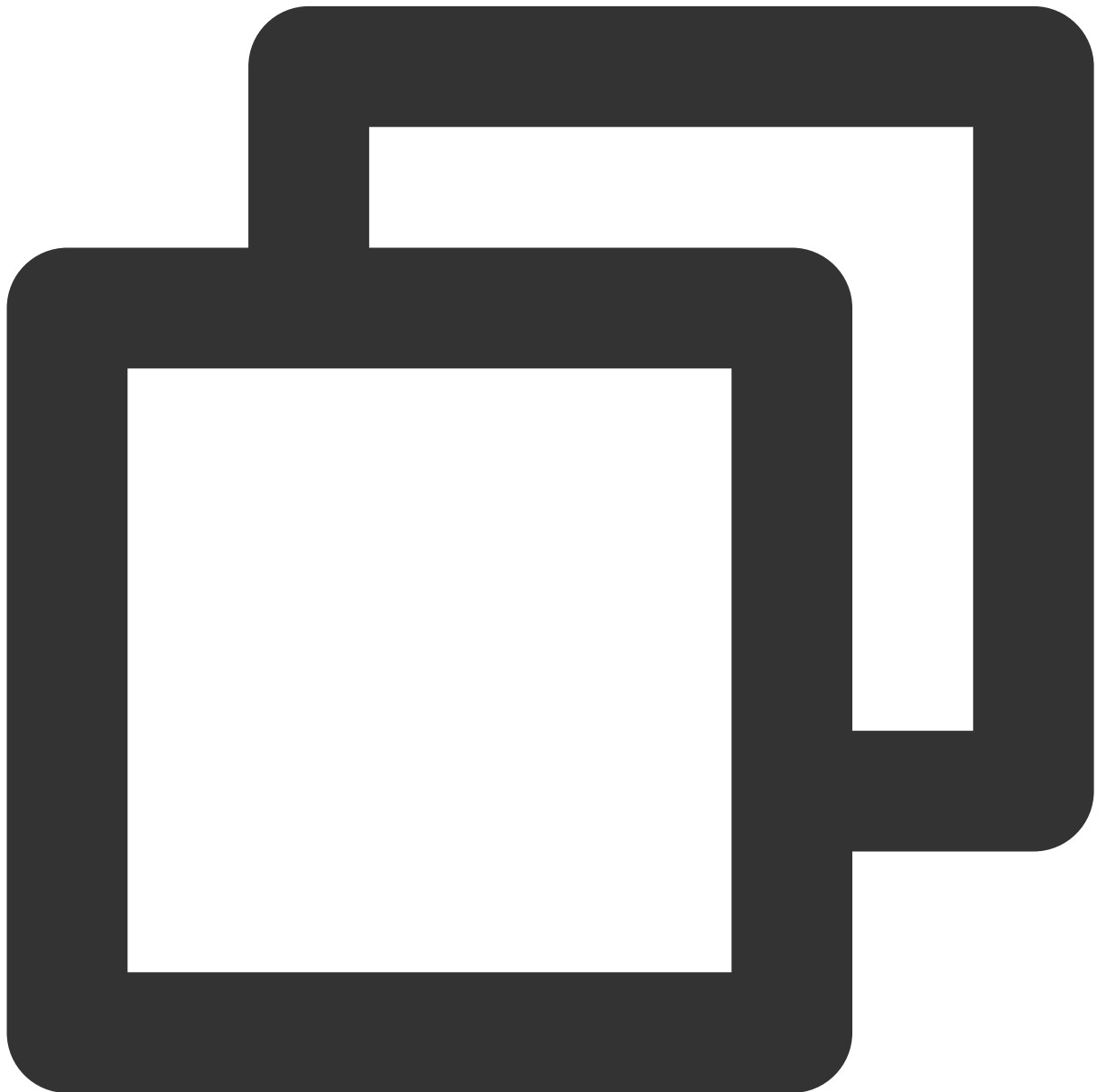


```
cloud.tim.qq.com  
openmsf.3g.qq.com
```

H5 SDK 사용

방화벽 포트:

포트 유형	얼로우리스트 항목
TCP 포트	443, 8687
UDP 포트	8000; 8800; 843; 443

도메인 이름 얼로우리스트:

qcloud.rtc.qq.com

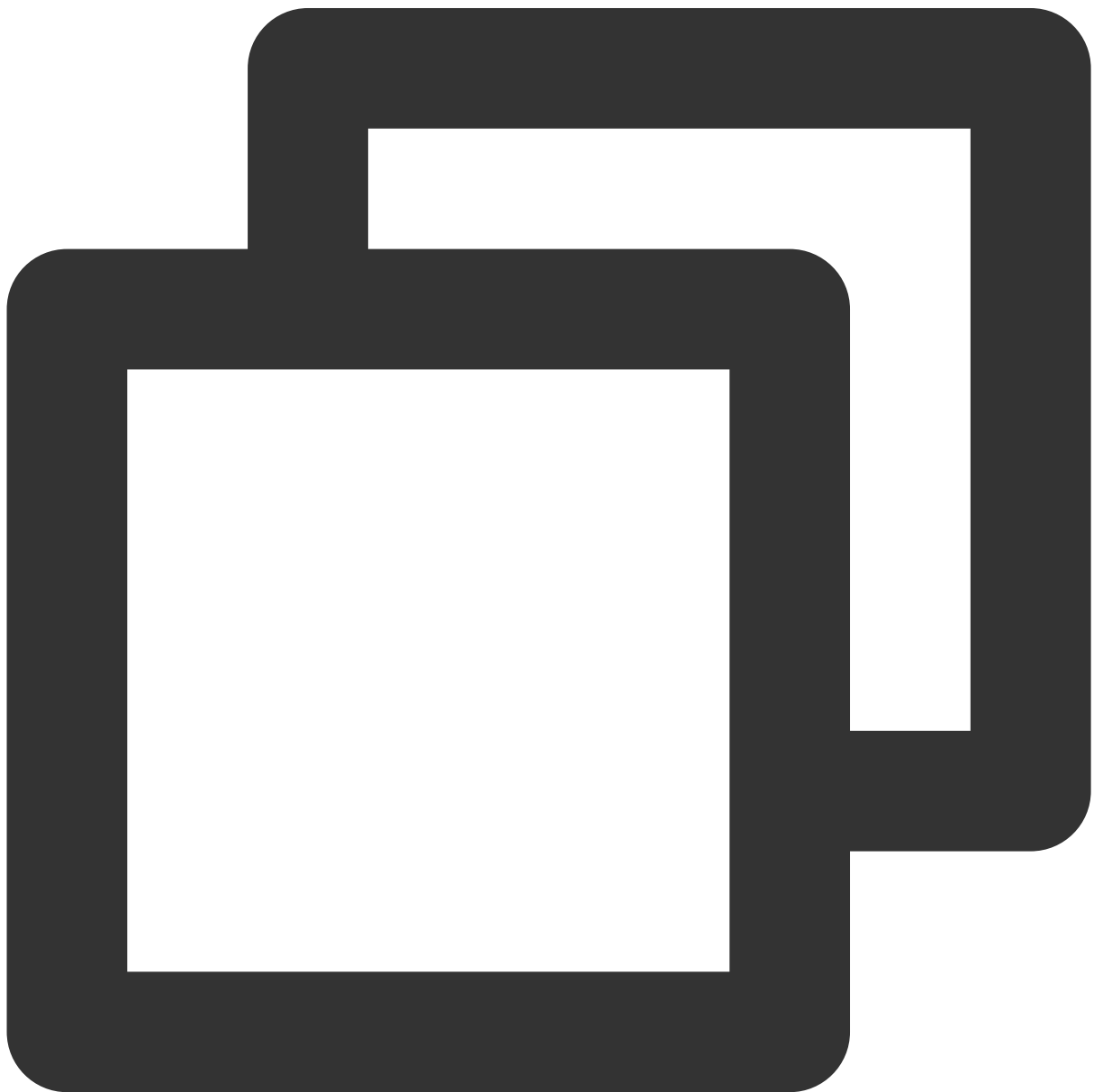
rtc.qqcloud.qq.com

음성 메시지 및 음성-텍스트 변환 서비스 사용

방화벽 포트:

포트 유형	얼로우리스트 항목
TCP 포트	80, 443

도메인 이름 얼로우리스트:



gmespeech.qcloud.com
yun.tim.qq.com
gmeconf.qcloud.com

Language Parameter Reference List

최종 업데이트 날짜: : 2023-04-04 15:33:25

This document describes the language parameters of GME's speech-to-text, text translation, and text-to-speech services.

Speech-to-text, text translation, and text-to-speech services

Currently, the speech-to-text, text translation, and text-to-speech services support mainstream languages. The text translation API is `TranslateText`, and the text-to-speech API is `TextToSpeech`.

Note

If your application needs to use the text translation or text-to-speech feature, [submit a ticket](#) for application.

Language	Parameter	Description
普通话（中国大陆）	cmn-Hans-CN	Chinese, Mandarin (Simplified, Chinese mainland)
國語（中国台灣）	cmn-Hant-TW	Chinese, Mandarin (Traditional, Taiwan (China))
廣東話（中国香港）	yue-Hant-HK	Chinese, Cantonese (Traditional, Hong Kong (China))
普通話（中国香港）	cmn-Hans-HK	Chinese, Mandarin (Simplified, Hong Kong (China))
Afrikaans (Suid-Afrika)	af-ZA	Afrikaans (South Africa)
አማርኛ (ኢትዮጵያ)	am-ET	Amharic (Ethiopia)
Հայ (Հայաստան)	hy-AM	Armenian (Armenia)
Azərbaycan (Azərbaycan)	az-AZ	Azerbaijani (Azerbaijan)
Bahasa Indonesia (Indonesia)	id-ID	Indonesian (Indonesia)
Bahasa Melayu (Malaysia)	ms-MY	Malay (Malaysia)
বাংলা (বাংলাদেশ)	bn-BD	Bengali (Bangladesh)
বাংলা (ভারত)	bn-IN	Bengali (India)
Català (Espanya)	ca-ES	Catalan (Spain)
Čeština (Česká republika)	cs-CZ	Czech (Czech Republic)

Dansk (Danmark)	da-DK	Danish (Denmark)
Deutsch (Deutschland)	de-DE	German (Germany)
English (Australia)	en-AU	English (Australia)
English (Canada)	en-CA	English (Canada)
English (Ghana)	en-GH	English (Ghana)
English (Great Britain)	en-GB	English (United Kingdom)
English (India)	en-IN	English (India)
English (Ireland)	en-IE	English (Ireland)
English (Kenya)	en-KE	English (Kenya)
English (New Zealand)	en-NZ	English (New Zealand)
English (Nigeria)	en-NG	English (Nigeria)
English (Philippines)	en-PH	English (Philippines)
English (South Africa)	en-ZA	English (South Africa)
English (Tanzania)	en-TZ	English (Tanzania)
English (United States)	en-US	English (United States)
Español (Argentina)	es-AR	Spanish (Argentina)
Español (Bolivia)	es-BO	Spanish (Bolivia)
Español (Chile)	es-CL	Spanish (Chile)
Español (Colombia)	es-CO	Spanish (Colombia)
Español (Costa Rica)	es-CR	Spanish (Costa Rica)
Español (Ecuador)	es-EC	Spanish (Ecuador)
Español (El Salvador)	es-SV	Spanish (El Salvador)
Español (España)	es-ES	Spanish (Spain)
Español (Estados Unidos)	es-US	Spanish (United States)
Español (Guatemala)	es-GT	Spanish (Guatemala)

Español (Honduras)	es-HN	Spanish (Honduras)
Español (México)	es-MX	Spanish (Mexico)
Español (Nicaragua)	es-NI	Spanish (Nicaragua)
Español (Panamá)	es-PA	Spanish (Panama)
Español (Paraguay)	es-PY	Spanish (Paraguay)
Español (Perú)	es-PE	Spanish (Peru)
Español (Puerto Rico)	es-PR	Spanish (Puerto Rico)
Español (República Dominicana)	es-DO	Spanish (Dominican Republic)
Español (Uruguay)	es-UY	Spanish (Uruguay)
Español (Venezuela)	es-VE	Spanish (Venezuela)
Euskara (Espainia)	eu-ES	Basque (Spain)
Filipino (Pilipinas)	fil-PH	Filipino (Philippines)
Français (Canada)	fr-CA	French (Canada)
Français (France)	fr-FR	French (France)
Galego (España)	gl-ES	Galician (Spain)
ქართული (საქართველო)	ka-GE	Georgian (Georgia)
ગુજરાતી (ભારત)	gu-IN	Gujarati (India)
Hrvatski (Hrvatska)	hr-HR	Croatian (Croatia)
IsiZulu (Ningizimu Afrika)	zu-ZA	Zulu (South Africa)
Íslenska (Ísland)	is-IS	Icelandic (Iceland)
Italiano (Italia)	it-IT	Italian (Italy)
Jawa (Indonesia)	jv-ID	Javanese (Indonesia)
()	kn-IN	Kannada (India)
()	km-KH	Khmer (Cambodia)
()	lo-LA	Lao (Laos)

Latviešu (latviešu)	lv-LV	Latvian (Latvia)
Lietuvių (Lietuva)	lt-LT	Lithuanian (Lithuania)
Magyar (Magyarország)	hu-HU	Hungarian (Hungary)
മലയാളം (ഇന്ത്യ)	ml-IN	Malayalam (India)
मराठी (भारत)	mr-IN	Marathi (India)
Nederlands (Nederland)	nl-NL	Dutch (Netherlands)
नेपाली (नेपाल)	ne-NP	Nepali (Nepal)
Norsk bokmål (Norge)	nb-NO	Norwegian Bokmål (Norway)
Polski (Polska)	pl-PL	Polish (Poland)
Português (Brasil)	pt-BR	Portuguese (Brazil)
Português (Portugal)	pt-PT	Portuguese (Portugal)
Română (România)	ro-RO	Romanian (Romania)
සිංහල (ශ්‍රී ලංකාව)	si-LK	Sinhalese (Sri Lanka)
Slovenčina (Slovensko)	sk-SK	Slovak (Slovakia)
Slovenščina (Slovenija)	sl-SI	Slovenian (Slovenia)
Urang (Indonesia)	su-ID	Sundanese (Indonesia)
Swahili (Tanzania)	sw-TZ	Swahili (Tanzania)
Swahili (Kenya)	sw-KE	Swahili (Kenya)
Suomi (Suomi)	fi-FI	Finnish (Finland)
Svenska (Sverige)	sv-SE	Swedish (Sweden)
தமிழ் (இந்தியா)	ta-IN	Tamil (India)
தமிழ் (சிங்கப்பூர்)	ta-SG	Tamil (Singapore)
தமிழ் (இலங்கை)	ta-LK	Tamil (Sri Lanka)
தமிழ் (மலேசியா)	ta-MY	Tamil (Malaysia)
()	te-IN	Telugu (India)

Tiếng Việt (Việt Nam)	vi-VN	Vietnamese (Vietnam)
Türkçe (Türkiye)	tr-TR	Turkish (Turkey)
اوردو (پاکستان)	ur-PK	Urdu (Pakistan)
اوردو (بھارت)	ur-IN	Urdu (India)
Ελληνικά (Ελλάδα)	el-GR	Greek (Greece)
Български (България)	bg-BG	Bulgarian (Bulgaria)
Русский (Россия)	ru-RU	Russian (Russia)
Српски (Србија)	sr-RS	Serbian (Serbia)
Українська (Україна)	uk-UA	Ukrainian (Ukraine)
עברית(ישראל)	he-IL	Hebrew (Israel)
العربية (إسرائيل)	ar-IL	Arabic (Israel)
العربية (الأردن)	ar-JO	Arabic (Jordan)
العربية (الإمارات)	ar-AE	Arabic (United Arab Emirates)
العربية (البحرين)	ar-BH	Arabic (Bahrain)
العربية (الجزائر)	ar-DZ	Arabic (Algeria)
العربية (السعودية)	ar-SA	Arabic (Saudi Arabia)
العربية (العراق)	ar-IQ	Arabic (Iraq)
العربية (الكويت)	ar-KW	Arabic (Kuwait)
العربية (المغرب)	ar-MA	Arabic (Morocco)
العربية (تونس)	ar-TN	Arabic (Tunisia)
العربية (عمان)	ar-OM	Arabic (Oman)
العربية (فلسطين)	ar-PS	Arabic (Palestine)
العربية (قطر)	ar-QA	Arabic (Qatar)
العربية (لبنان)	ar-LB	Arabic (Lebanon)
العربية (مصر)	ar-EG	Arabic (Egypt)

فارسی (ایران)	fa-IR	Persian (Iran)
हिन्दी (भारत)	hi-IN	Hindi (India)
ไทย (ประเทศไทย)	th-TH	Thai (Thailand)
한국어 (대한민국)	ko-KR	Korean (South Korea)

GME 채팅방 관리 통합

최종 업데이트 날짜: : 2024-01-18 16:00:04

주의 사항:

GME 3.x 버전은 방 관리 기능을 지원하지 않습니다.

이 문서는 방 관리 기능의 사용 사례와 기능에 빠르게 연결하는 방법을 설명합니다.

기능 소개

클라이언트 방 관리 API를 사용하여 방 구성원 및 구성원 마이크 켜짐/꺼짐 상태를 쉽게 관리할 수 있습니다.

사용 사례

예를 들어 Werewolf의 호스트는 EnableMic를 사용하여 다른 플레이어가 마이크를 사용하도록 제어할 수 있습니다. 플레이어가 '죽었고' 회의실 구성원의 말을 듣거나 마이크와 대화할 필요가 없는 경우 호스트는 ForbidUserOperation API를 사용하여 플레이어가 장치를 조작하지 못하도록 금지합니다.

전제 조건

실시간 음성 채팅 서비스 활성화 완료: [서비스 활성화](#)를 참고하십시오.

GME SDK 통합 완료: 핵심 API 및 실시간 음성 채팅 API 통합을 포함합니다. 자세한 내용은 [Native SDK 빠른 통합](#), [Unity SDK 빠른 통합](#), [Unreal SDK 빠른 통합](#)을 참고하십시오.

설명 :

이 기능은 H5용 SDK에서 지원되지 않습니다.

연결 프로세스

클래스 이름: ITMGRoomManager

GME 채팅방 관리는 방에 1명 이상의 구성원이 있을 때만 사용할 수 있으며, 방 구성원의 상태를 수정하는 경우에만 사용할 수 있습니다.

API 응답에 대한 모든 콜백은 `ITMG_MAIN_EVNET_TYPE_ROOM_MANAGEMENT_OPERATOR` 를 통해 처리됩니다.

콜백에 대한 자세한 내용은 [콜백 처리](#)를 참고하십시오.

API 목록

유형	API
캡처 제어	EnableMic, EnableAudioCaptureDevice, EnableAudioSend
재생 제어	EnableSpeaker, EnableAudioPlayDevice, EnableAudioRecv
디바이스 상태 가져오기	GetMicState, GetSpeakerState
민감한 API	ForbidUserOperation

캡처 관리 API

캡처 관리 API는 **마이크**, **오디오 업스트림** 및 **캡처 장치**를 개별적으로 관리하는 데 사용됩니다. 마이크를 관리하는 것은 오디오 업스트림 및 캡처 장치를 모두 관리하는 것과 같습니다.

캡처 관리

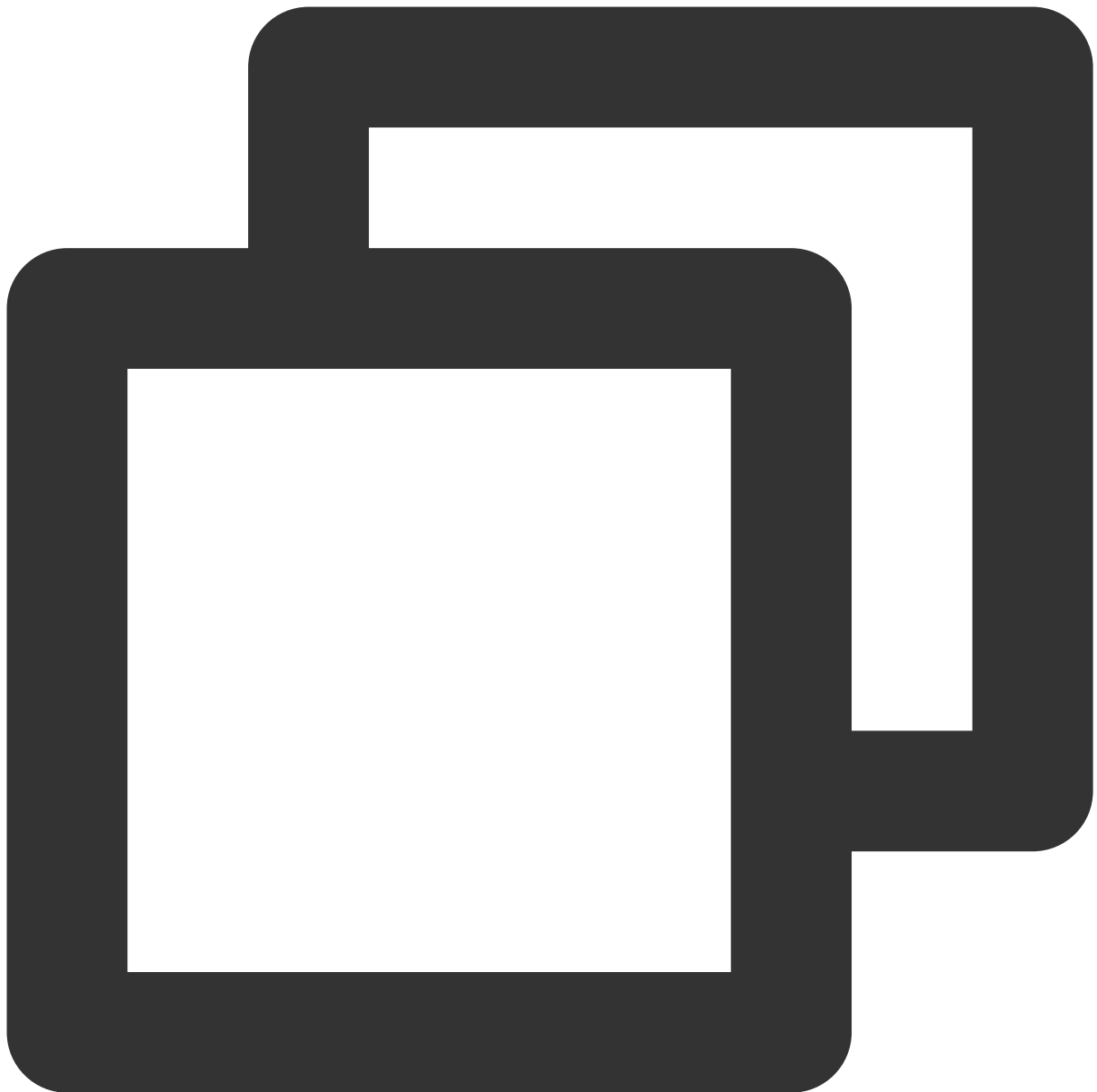
이 API(EnableMic)는 채팅방에서 사용자의 마이크를 활성화/비활성화하는 데 사용됩니다.

EnableMic를 호출하는 것은 EnableAudioSend 및 EnableAudioCaptureDevice를 모두 호출하는 것과 같습니다.

함수 프로토타입

Android

iOS



```
public abstract int EnableMic(boolean isEnabled,String receiverID);
```



```
-(QAVResult)EnableMic:(BOOL)enable Receiver:(NSString *)receiverID;
```

매개변수	유형	설명
enable	BOOL	YES: 사용자 마이크 활성화, NO: 사용자 마이크 비활성화
receiverID	NSString*	사용자의 OpenId 지정

콜백

콜백 매개변수는 ITMG_ROOM_MANAGEMENT_MIC_OP입니다.

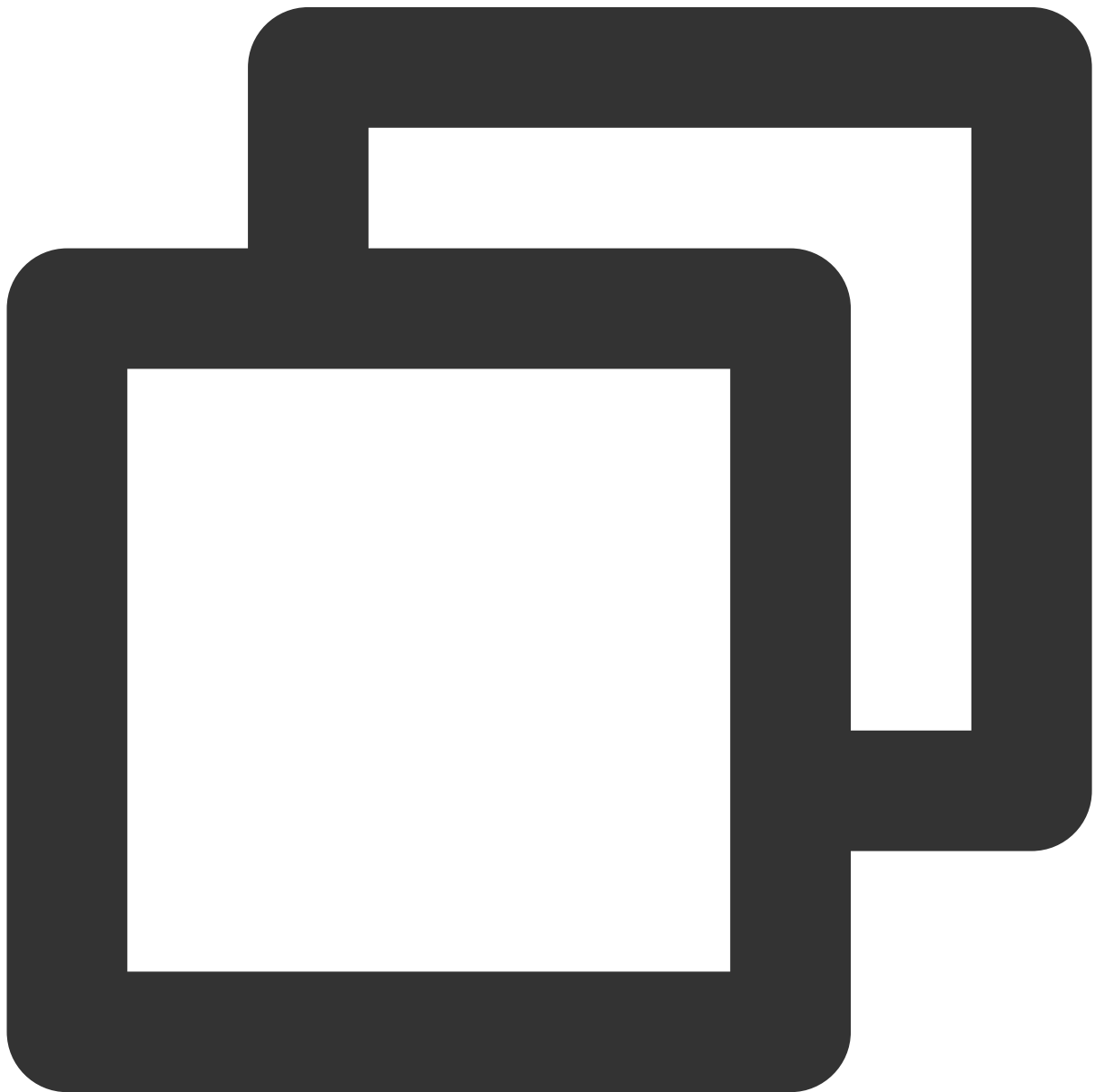
오디오 스트림 전송 관리

이 API(EnableAudioSend)는 채팅방에서 사용자의 오디오 업스트림을 활성화/비활성화하는 데 사용되지만 마이크 캡처에는 영향을 미치지 않습니다.

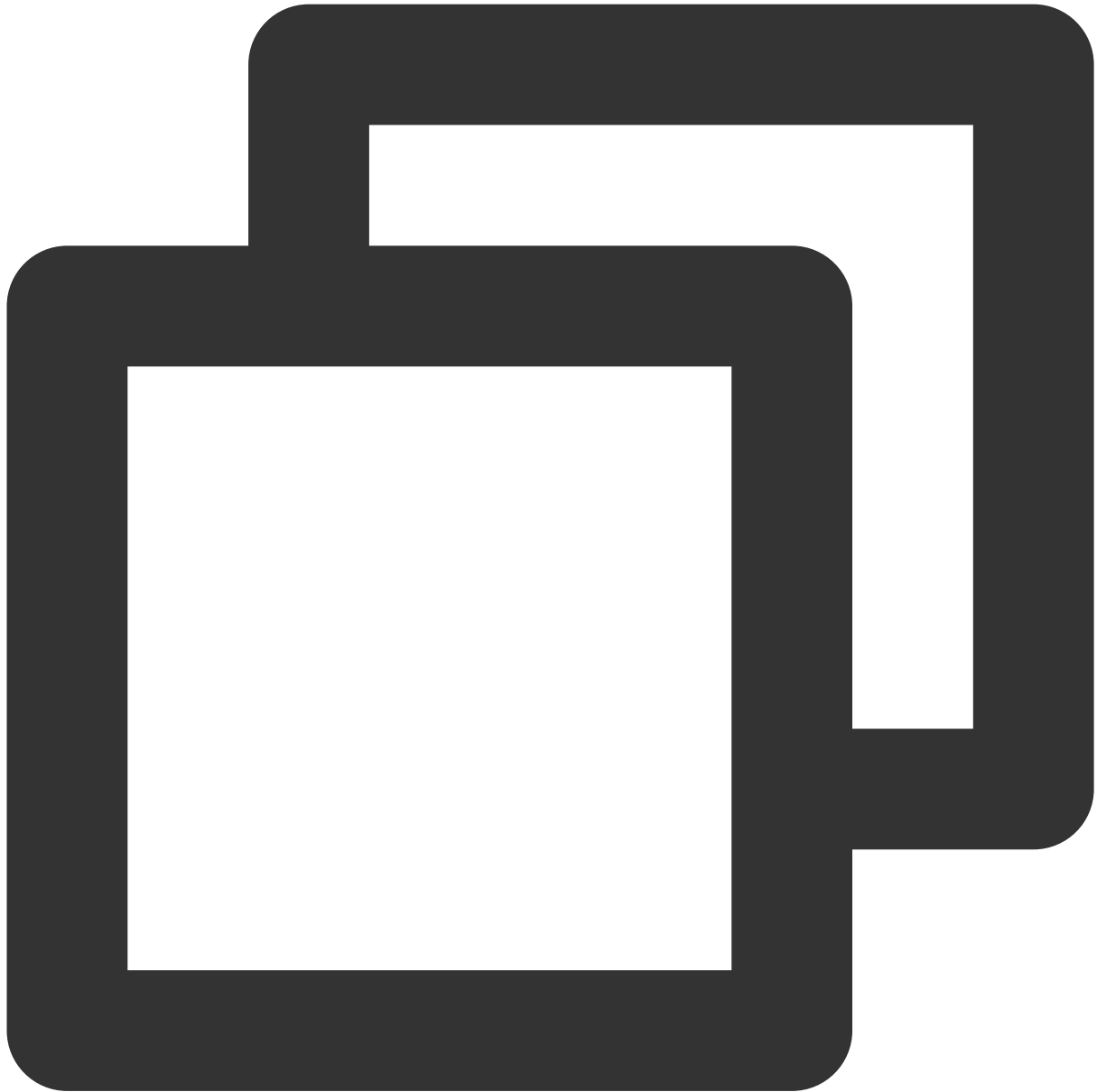
함수 프로토타입

Android

iOS



```
public abstract int EnableAudioSend(boolean isEnabled,String receiverID);
```



```
-(QAVResult)EnableAudioSend:(BOOL)enable Receiver:(NSString *)receiverID;
```

매개변수	유형	설명
enable	BOOL	YES: 사용자 업스트림 활성화, NO: 사용자 업스트림 비활성화
receiverID	NSString*	사용자의 OpenId 지정

콜백

콜백 매개변수는 ITMG_ROOM_MANAGEMENT_AUDIO_SEND_OP입니다.

오디오 캡처 하드웨어 관리

이 API(EnableAudioCaptureDevice)는 채팅방에서 사용자의 오디오 캡처 장치를 활성화/비활성화하는 데 사용되지만 오디오 업스트림에는 영향을 미치지 않습니다.

함수 프로토타입

Android

iOS



```
public abstract int EnableAudioCaptureDevice(boolean isEnabled,String receiverID);
```



```
-(QAVResult)EnableAudioCaptureDevice:(BOOL)enabled Receiver:(NSString *)receiverID;
```

매개변수	유형	설명
enable	BOOL	YES: 사용자 오디오 캡처 장치 활성화, NO: 사용자 오디오 캡처 장치 비활성화
receiverID	NSString*	사용자의 OpenId 지정

콜백

콜백 매개변수는 ITMG_ROOM_MANAGEMENT_CAPTURE_OP입니다.

재생 관리 API

재생 관리 API는 **스피커**, **오디오 다운스트림** 및 **재생 장치**를 개별적으로 관리하는 데 사용됩니다. 스피커를 관리하는 것은 오디오 다운스트림 및 재생 장치를 모두 관리하는 것과 같습니다.

재생 관리

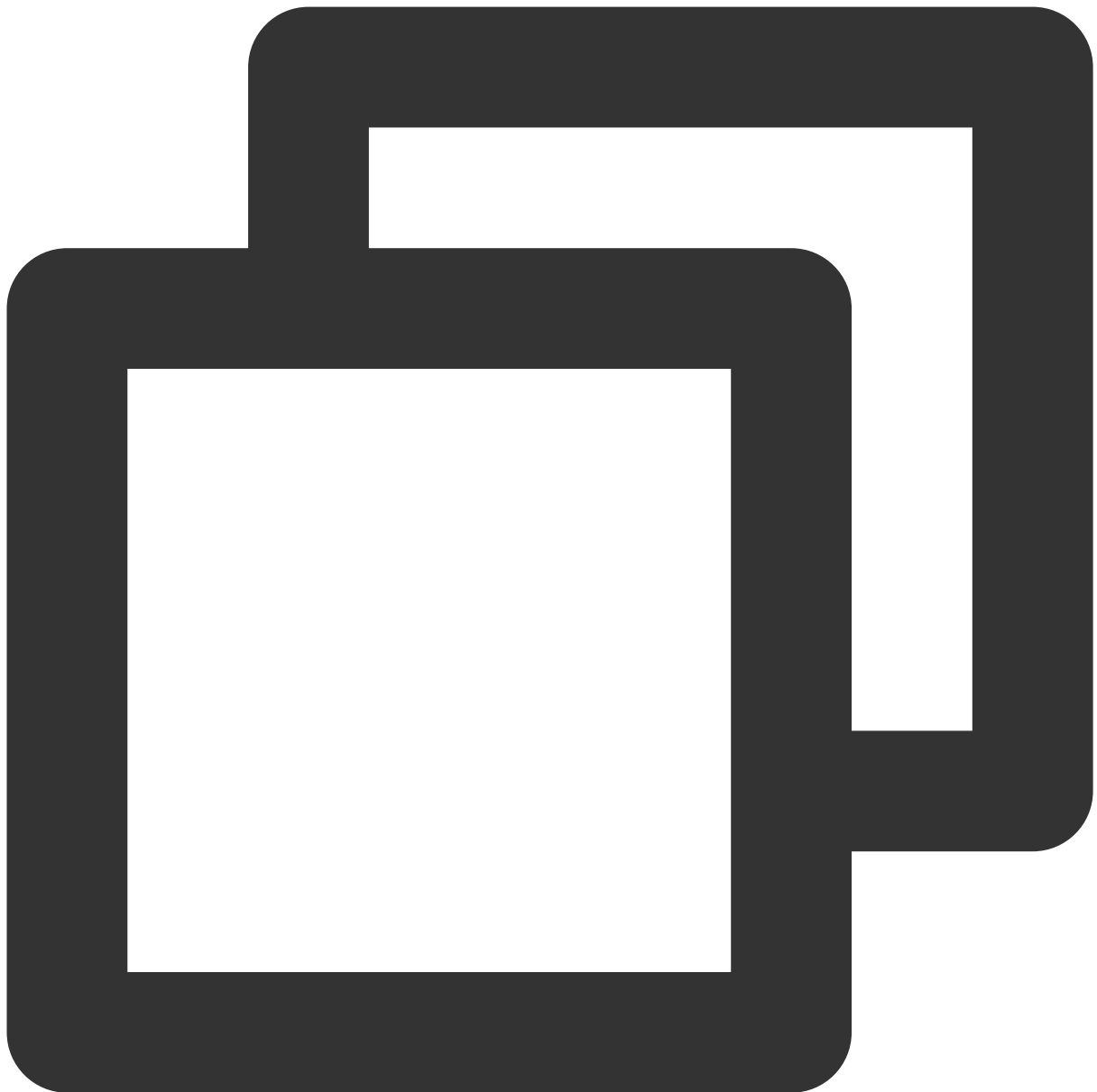
이 API(EnableSpeaker)는 사용자가 채팅방에서 소리를 들을 수 있도록 스피커를 활성화/비활성화하는 데 사용됩니다.

EnableSpeaker를 호출하는 것은 EnableAudioRecv 및 EnableAudioPlayDevice를 모두 호출하는 것과 같습니다.

함수 프로토타입

Android

iOS



```
public abstract int EnableSpeaker(boolean isEnabled,String receiverID);
```



```
-(QAVResult)EnableSpeaker:(BOOL)enable Receiver:(NSString *)receiverID;
```

매개변수	유형	설명
enable	BOOL	YES: 사용자 스피커 활성화, NO: 사용자 스피커 비활성화
receiverID	NSString*	사용자의 OpenId 지정

콜백

콜백 매개변수는 ITMG_ROOM_MANAGEMENT_SPEAKER_OP입니다.

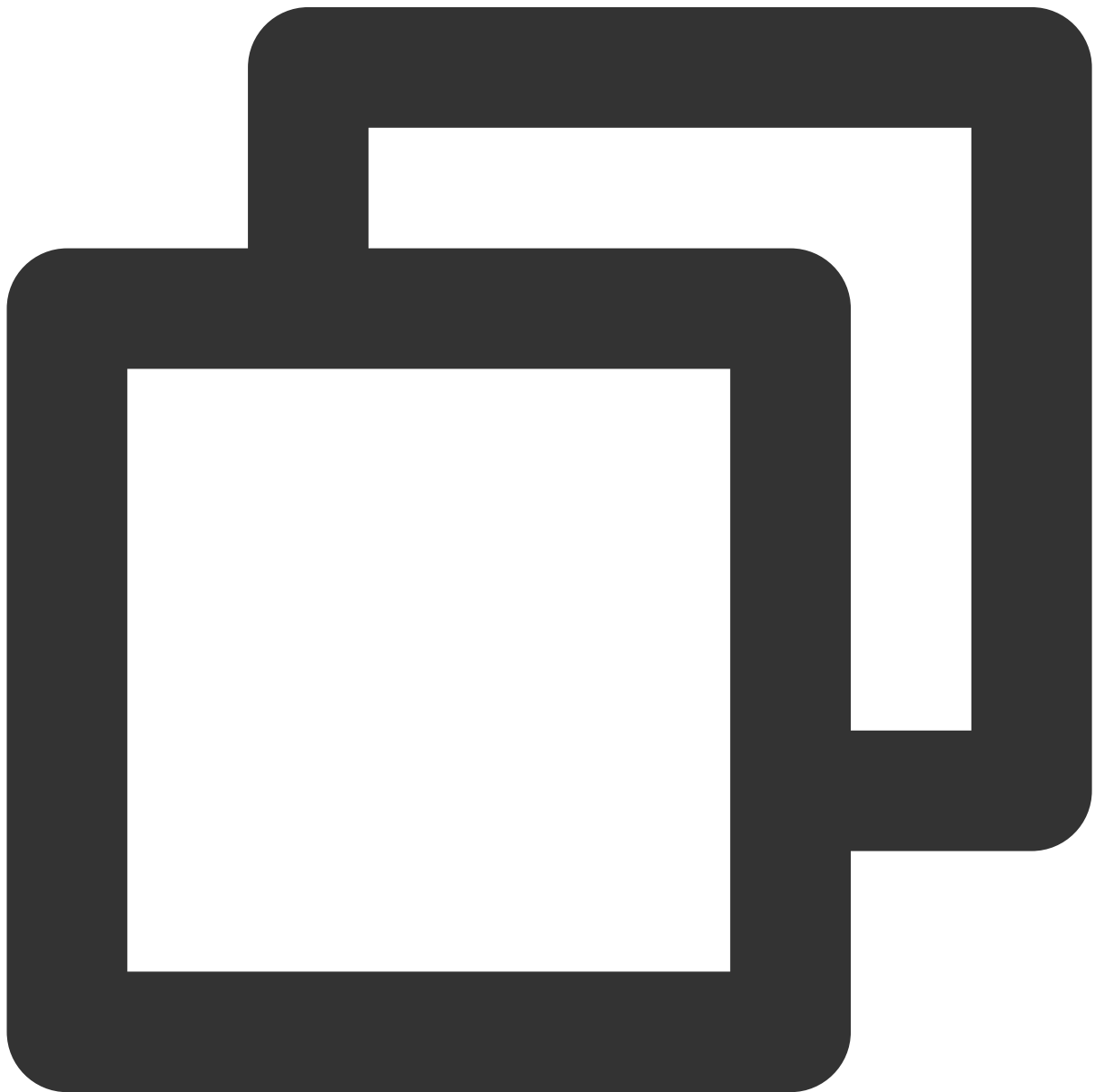
오디오 스트림 수신 관리

이 API(EnableAudioRecv)는 사용자를 위해 오디오 다운스트림을 활성화/비활성화하는 데 사용되지만 재생 장치에는 영향을 미치지 않습니다.

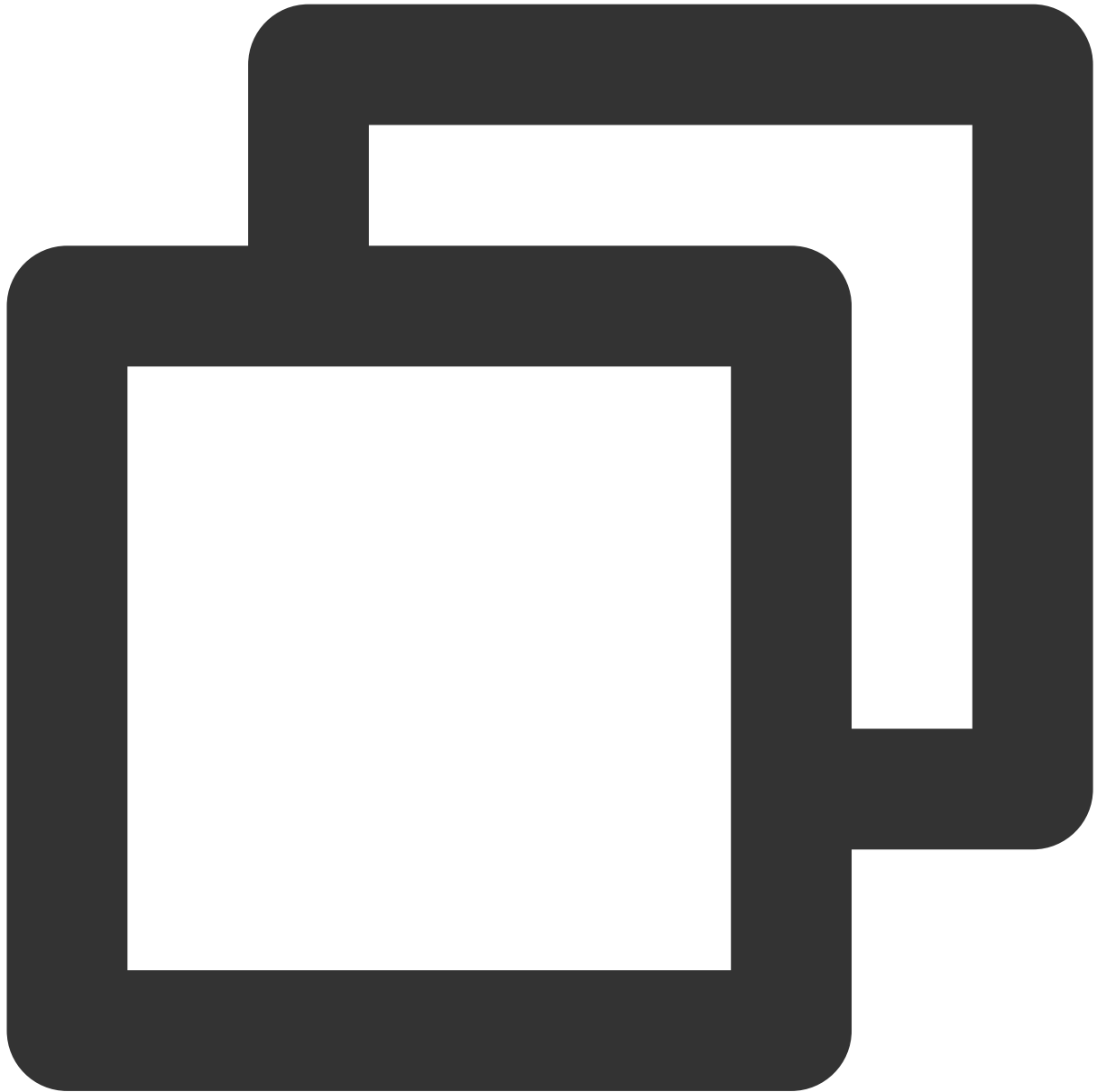
함수 프로토타입

Android

iOS



```
public abstract int EnableAudioRecv(boolean isEnabled, String receiverID);
```



```
-(QAVResult) EnableAudioRecv:(BOOL)enabled Receiver:(NSString *)receiverID;
```

매개변수	유형	설명
enable	BOOL	YES: 사용자 오디오 다운스트림 활성화, NO: 사용자 오디오 다운스트림 비활성화
receiverID	NSString*	사용자의 OpenId 지정

콜백

콜백 매개변수는 ITMG_ROOM_MANAGEMENT_AUDIO_REC_OP입니다.

오디오 재생 하드웨어 관리

이 API(EnableAudioPlayDevice)는 사용자의 오디오 재생 장치를 활성화/비활성화하는 데 사용되지만 오디오 다운스트림에는 영향을 미치지 않습니다.

함수 프로토타입

Android

iOS



```
public abstract int EnableAudioPlayDevice(boolean isEnabled,String receiverID);
```



```
-(QAVResult)EnableAudioPlayDevice:(BOOL)enabled Receiver:(NSString *)receiverID;
```

매개변수	유형	설명
enable	BOOL	YES: 사용자 오디오 재생 장치 활성화, NO: 사용자 오디오 재생 장치 비활성화
receiverID	NSString*	사용자의 OpenId 지정

콜백

콜백 매개변수는 ITMG_ROOM_MANAGEMENT_PLAY_OP입니다.

사용자 상태를 가져오기 위한 API

캡처 상태 가져오기

이 API는 채팅방에서 사용자의 마이크 상태를 가져오는 데 사용됩니다.

함수 프로토타입

Android

iOS



```
public abstract int GetMicState(String receiverID);
```



```
-(QAVResult)GetMicState:(NSString *)receiverID;
```

매개변수	유형	설명
receiverID	NSString*	사용자의 OpenId 지정

콜백

콜백 매개변수는 ITMG_ROOM_MANAGEMENT_GET_MIC_STATE입니다.

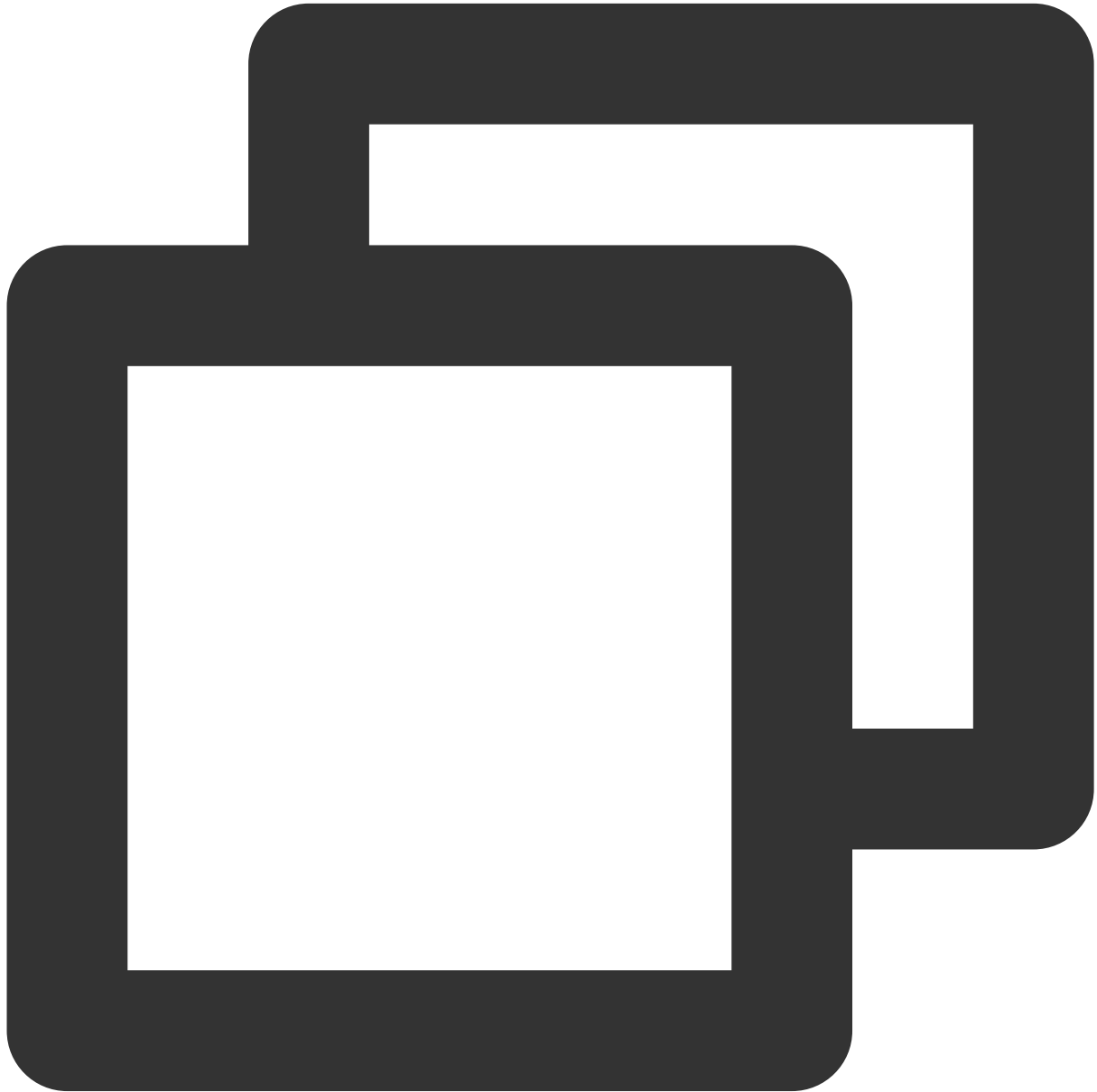
재생 상태 가져오기

이 API는 채팅방에서 사용자의 스피커 상태를 가져오는 데 사용됩니다.

함수 프로토타입

Android

iOS



```
public abstract int GetSpeakerState(String receiverID);
```



```
-(QAVResult) GetSpeakerState: (NSString *) receiverID;
```

콜백

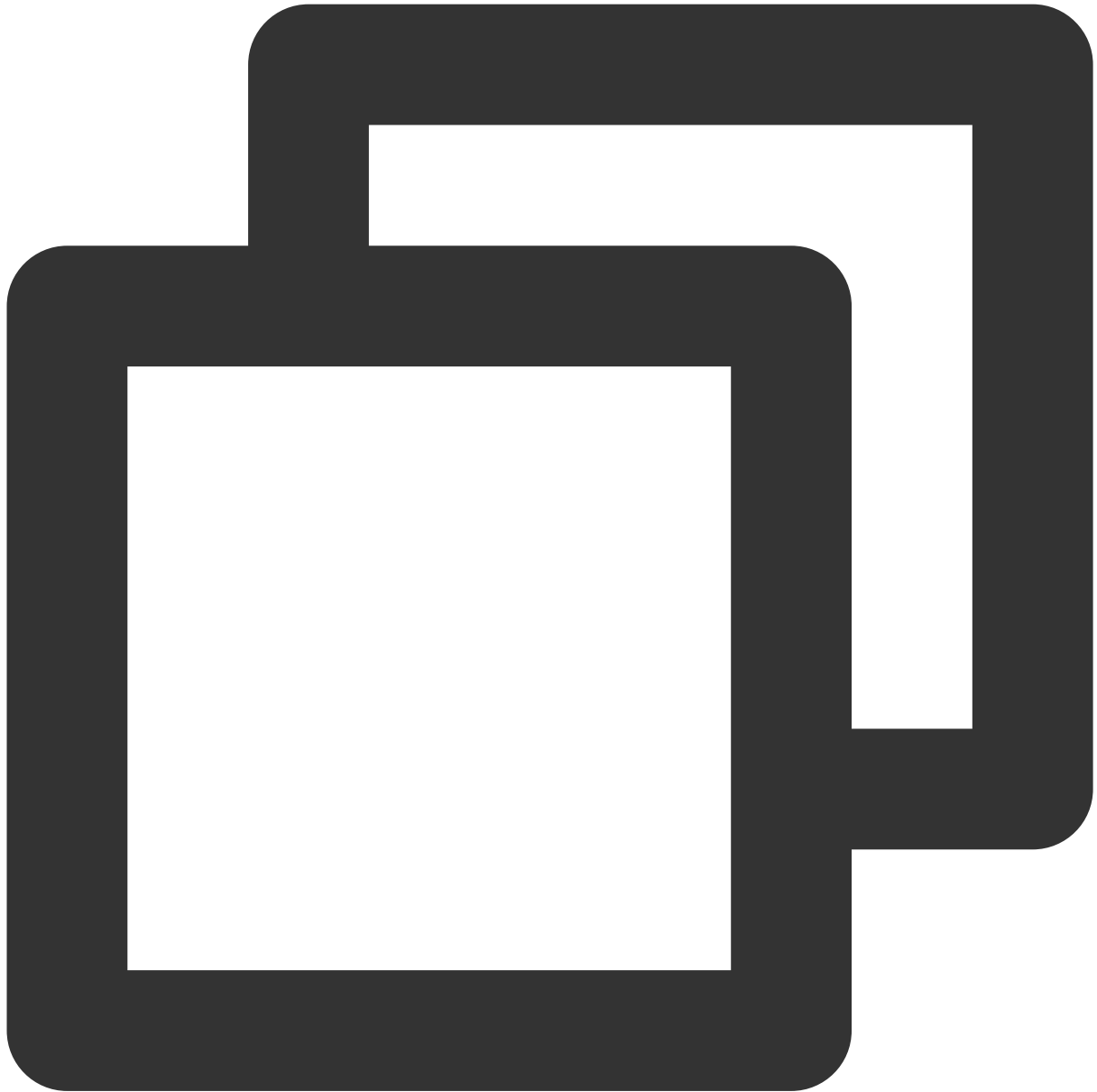
콜백 매개변수는 ITMG_ROOM_MANAGEMENT_GET_SPEAKER_STATE입니다.

사용자의 마이크/스피커를 비활성화하기 위한 API

채팅방에 입장하는 각 사용자는 기본적으로 마이크와 스피커를 사용할 수 있습니다. 이 API를 호출하면 사용자가 방을 나갈 때까지 마이크와 스피커가 비활성화됩니다.

Android

iOS



```
public abstract int ForbidUserOperation(boolean isEnabled,String receiverID);
```



```
-(QAVResult)ForbidUserOperation:(BOOL)enable Receiver:(NSString *)receiverID;
```

매개변수	유형	설명
enable	BOOL	YES: 사용자 장치 비활성화, NO: 사용자 장치 활성화
receiverID	NSString*	사용자의 OpenId 지정

콜백

콜백 매개변수는 ITMG_ROOM_MANAGEMENT_FOBIN_OP입니다.

콜백 처리

다른 모든 GME 콜백과 마찬가지로 채팅방 관리 콜백은

`ITMG_MAIN_EVNET_TYPE_ROOM_MANAGEMENT_OPERATOR` 라는 이벤트와 함께 `OnEvent`를 사용하여 처리됩니다. 이 이벤트는 아래와 같이 매개변수 구조를 반환합니다.

콜백 매개변수

매개변수	유형	설명
SenderID	NSString	이벤트 발신자의 ID입니다. 자신의 OpenId와 같으면 발신자는 클라이언트입니다.
ReceiverID	NSString	이벤트 수신자의 ID입니다. 자신의 OpenId와 같으면 수신자는 클라이언트입니다.
OperateType	NSNumber	이벤트 유형
Result	NSNumber	이벤트 결과, 0은 성공을 의미합니다
OperateValue	NSNumber	콜백 세부 정보

OperateType

값	이벤트 유형	설명
0	ITMG_ROOM_MANAGEMENT_CAPTURE_OP	캡처 장치 제어 콜백
1	ITMG_ROOM_MANAGEMENT_PLAY_OP	재생 장치 제어 콜백
2	ITMG_ROOM_MANAGEMENT_AUDIO_SEND_OP	오디오 업스트림 제어 콜백
3	ITMG_ROOM_MANAGEMENT_AUDIO_REC_OP	오디오 다운스트림 제어 콜백
4	ITMG_ROOM_MANAGEMENT_MIC_OP	마이크 제어 콜백
5	ITMG_ROOM_MANAGEMENT_PLAY_OP	스피커 제어 콜백
6	ITMG_ROOM_MANAGEMENT_GET_MIC_STATE	마이크 상태 가져오기
7	ITMG_ROOM_MANAGEMENT_GET_SPEAKER_STATE	스피커 상태 가져오기
8	ITMG_ROOM_MANAGEMENT_FOBIN_OP	마이크와 스피커 비활성화

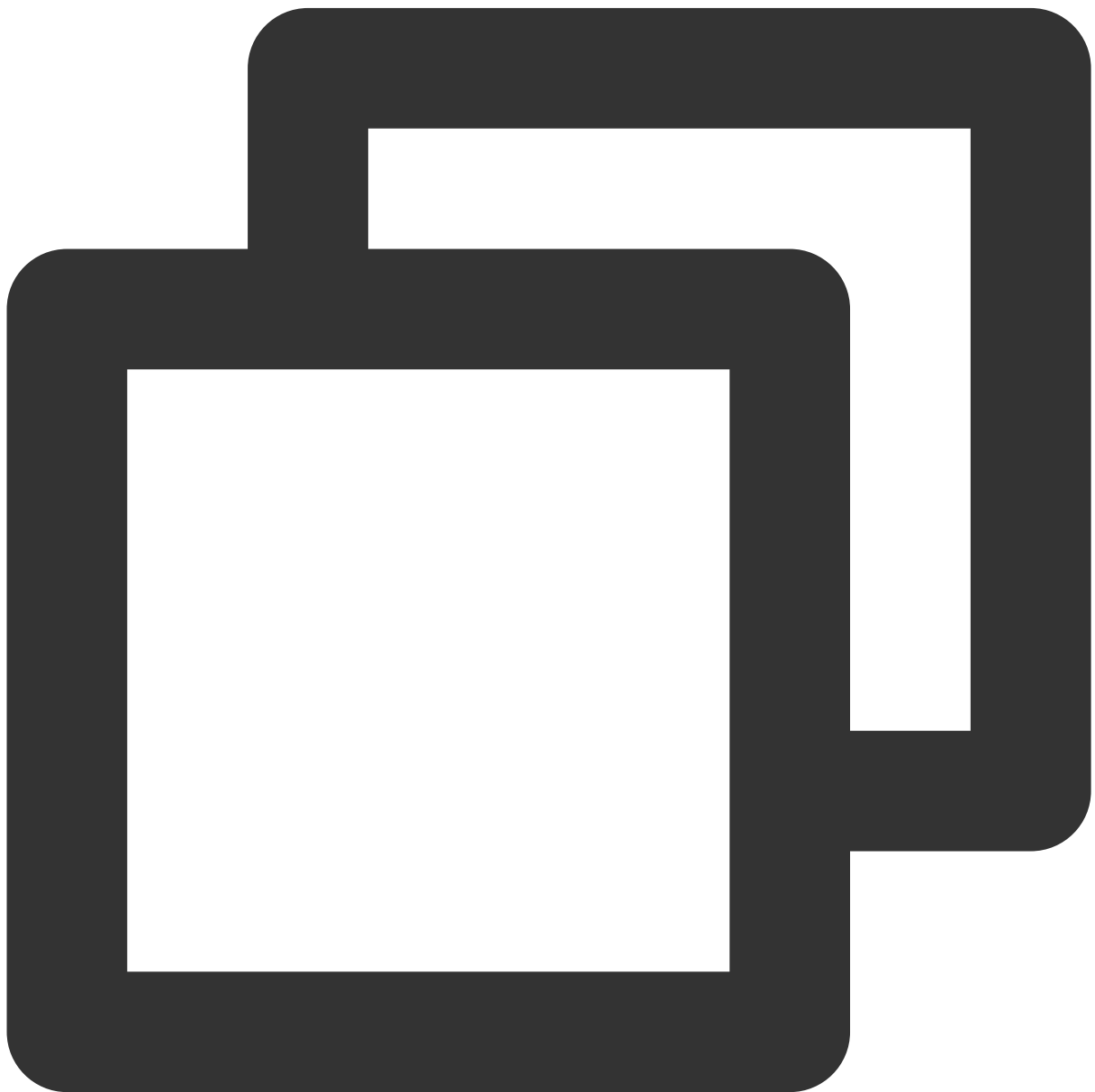
OperateValue

참석자	설명
boolValue	0: 비활성화, 1: 활성화

예시 코드

Android

iOS



```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_ROOM_MANAGEMENT_OPERATOR

        ArrayList<String> operatorArr = new ArrayList<String>();
        operatorArr.add("캡처");
        operatorArr.add("재생");
        operatorArr.add("업스트림");
        operatorArr.add("다운스트림");
        operatorArr.add("캡처 및 업스트림");
        operatorArr.add("재생 및 다운스트림");
        operatorArr.add("mic 상태");
        operatorArr.add("spk 상태");
        operatorArr.add("mic/speak 비활성화");

        String SenderID = data.getStringExtra("SenderID");
        String ReceiverID = data.getStringExtra("ReceiverID");
        int OperateType = data.getIntExtra("OperateType", -1000);

        int Result = data.getIntExtra("Result", -1000);
        boolean OperateValue = data.getBooleanExtra("OperateValue", false);
        if (OperateType == -1000 || Result == -1000) {
            return;
        }
        if (SenderID.equals(identifier)) {
            if (OperateType == ITMGContext.ITMG_ROOM_MANAGEMENT_GET_MIC_STATE |
                Toast.makeText(getActivity(), String.format("id:%s에게 발송한 %s
            } else {
                Toast.makeText(getActivity(), String.format("id:%s에게 발송한 %s%
            }

        } else if (ReceiverID.equals(identifier) || ReceiverID.equals("ALL")) {
            if (Result == 0) {
                switch (OperateType) {
                    case ITMGContext.ITMG_ROOM_MANAGEMENT_CAPTURE_OP:
                        {
                            if (!OperateValue) {
                                mSwitchCapture.setChecked(OperateValue);
                            } else {
                                AlertDialog.Builder dialog = new AlertDialog.Builder
                                dialog.setTitle("장치 캡처 활성화 여부");
                                dialog.setMessage("");
                                dialog.setCancelable(false);
                                dialog.setPositiveButton("On", new DialogInterface.
                                    //확인 버튼의 클릭 이벤트 설정
                                    @Override
                                    public void onClick(DialogInterface dialog, int
```

```
        mSwitchCapture.setChecked(true);
        ITMGContext.GetInstance(getActivity()).GetA
    }
});
dialog.setNegativeButton("Off", new DialogInterface
//취소 버튼의 클릭 이벤트 설정
@Override
public void onClick(DialogInterface dialog, int
}
});
dialog.show();
}

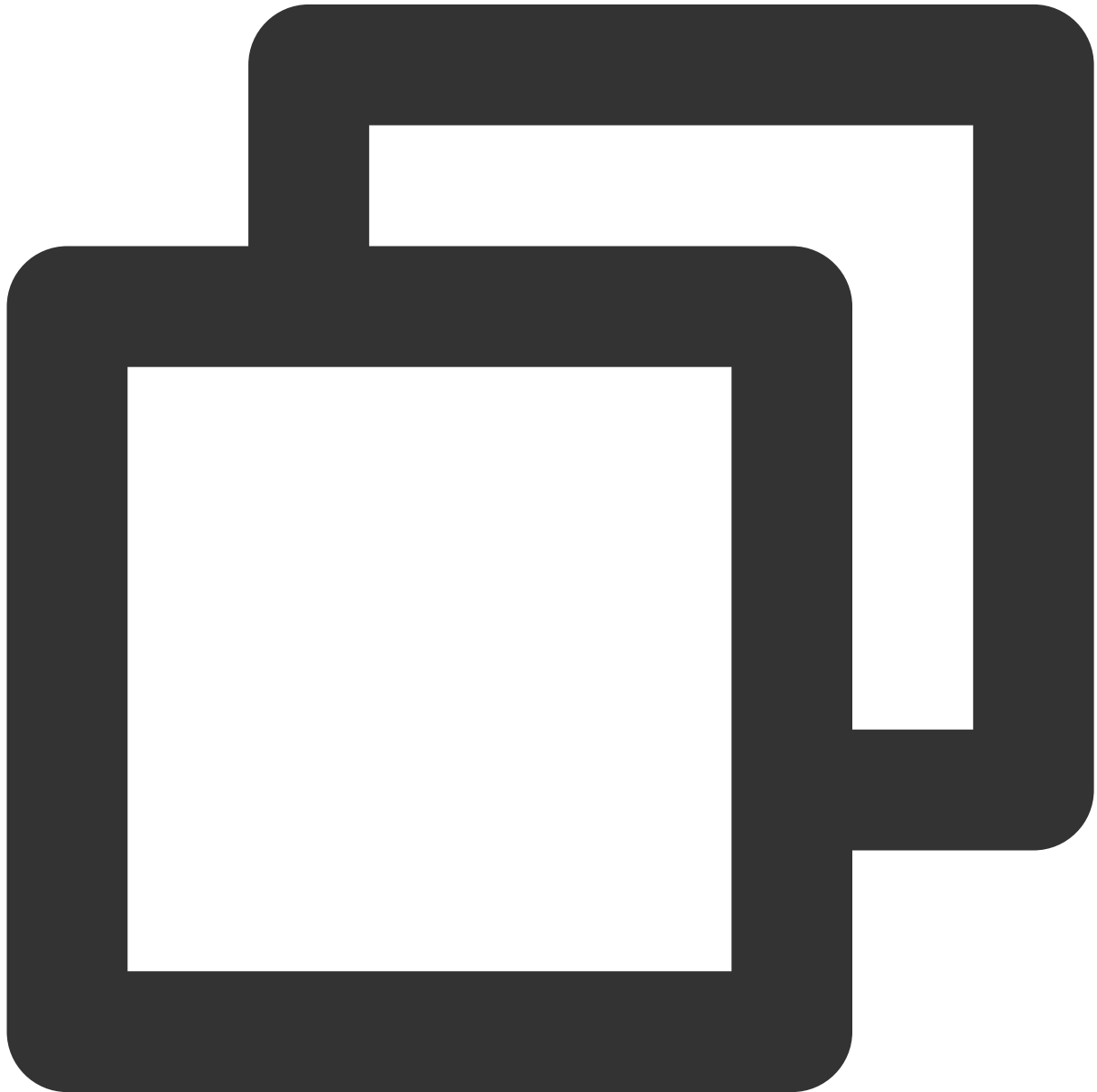
}
break;
case ITMGContext.ITMG_ROOM_MANAGEMENT_PLAY_OP:
{
    mSwitchPlayDevice.setChecked(OperateValue);
}
break;
case ITMGContext.ITMG_ROOM_MANAGEMENT_AUDIO_SEND_OP:
{
    if (!OperateValue) {
        mSwitchSend.setChecked(OperateValue);
    } else {
        AlertDialog.Builder dialog = new AlertDialog.Builder
        dialog.setTitle("업스트림 활성화 여부");
        dialog.setMessage("");
        dialog.setCancelable(false);
        dialog.setPositiveButton("On", new DialogInterface.
            //확인 버튼의 클릭 이벤트 설정
            @Override
            public void onClick(DialogInterface dialog, int
                mSwitchSend.setChecked(true);
                ITMGContext.GetInstance(getActivity()).GetA
            }
        });
        dialog.setNegativeButton("Off", new DialogInterface
            //취소 버튼의 클릭 이벤트 설정
            @Override
            public void onClick(DialogInterface dialog, int
            }
        });
        dialog.show();
    }
}
break;
```



```
case ITMGContext.ITMG_ROOM_MANAGEMENT_AUDIO_REC_OP:
{
    mSwitchRecv.setChecked(OperateValue);
}
break;
case ITMGContext.ITMG_ROOM_MANAGEMENT_MIC_OP:
{
    if (!OperateValue) {
        mSwitchCapture.setChecked(OperateValue);
        mSwitchSend.setChecked(OperateValue);
    } else {
        AlertDialog.Builder dialog = new AlertDialog.Builder
        dialog.setTitle("캡처 및 업스트림 활성화 여부");
        dialog.setMessage("");
        dialog.setCancelable(false);
        dialog.setPositiveButton("On", new DialogInterface
            //확인 버튼의 클릭 이벤트 설정
            @Override
            public void onClick(DialogInterface dialog, in
                mSwitchCapture.setChecked(true);
                mSwitchSend.setChecked(true);
                ITMGContext.GetInstance(getActivity()).Get
            }
        });
        dialog.setNegativeButton("Off", new DialogInterfac
            //취소 버튼의 클릭 이벤트 설정
            @Override
            public void onClick(DialogInterface dialog, in
            }
        });
        dialog.show();
    }
}
break;
case ITMGContext.ITMG_ROOM_MANAGEMENT_SPEAKER_OP:
{
    mSwitchPlayDevice.setChecked(OperateValue);
    mSwitchRecv.setChecked(OperateValue);
}
break;

}
}
if (OperateType == ITMGContext.ITMG_ROOM_MANAGEMENT_GET_MIC_STATE |
{
    Toast.makeText(getActivity(), String.format("id:%s(으)로부터 %s",
}
```

```
        else if (OperateType == ITMGContext.ITMG_ROOM_MANAGEMENT_SPEAKER_OP
                Toast.makeText(getActivity(), String.format("id:%s(으)로부터 %s%:
        } else if (OperateValue == false) {
                Toast.makeText(getActivity(), String.format("id:%s(으)로부터 %s%:
        }
    }
}
```



```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSString* log = [NSString stringWithFormat:@"OnEvent:%d,data:%@", (int)eventType
    [self showLog:log];
}
```

```
NSLog(@"====%@====", log);
switch (eventType) {
    case ITMG_MAIN_EVNET_TYPE_ROOM_MANAGEMENT_OPERATOR:
    {
        NSArray *operatorArr = @[@"캡처",@"재생",@"업스트림",@"다운스트림",@"캡처 및
        // _openId
        NSString *SenderID = [data objectForKey:@"SenderID"];
        NSString *ReceiverID = [data objectForKey:@"ReceiverID"];
        NSNumber *OperateType = [data objectForKey:@"OperateType"];
        NSNumber *Result = [data objectForKey:@"Result"];
        NSNumber *OperateValue = [data objectForKey:@"OperateValue"];

        //자신이 전송한 명령
        if ([SenderID isEqualToString:_openId]) {
            if (OperateType.intValue == ITMG_ROOM_MANAGEMENT_GET_MIC_STATE || O
                NSString *alterString = [NSString stringWithFormat:@"id:%
                    UIAlertView *alert = [[UIAlertView all
                        [alert show];
            }
            else
            {
                NSString *alterString = [NSString stringWithFormat:@"id:%@에게
                    UIAlertView *alert = [[UIAlertView alloc] initWithTi
                        [alert show];
            }
        }

        else if([ReceiverID isEqualToString:_openId] ){ //다른 사람이 전송한 명령
            if (Result.intValue == 0) {
                switch (OperateType.intValue) {
                    case ITMG_ROOM_MANAGEMENT_CAPTURE_OP:{
                        [_micSwitch setOn:OperateValue.boolValue animated:true]
                    }
                    break;
                    case ITMG_ROOM_MANAGEMENT_PLAY_OP:{
                        [_speakerSwitch setOn:OperateValue.boolValue animated:t
                    }
                    break;
                    case ITMG_ROOM_MANAGEMENT_AUDIO_SEND_OP:{
                        [_sendSwitch setOn:OperateValue.boolValue animated:true
                    }
                    break;
                    case ITMG_ROOM_MANAGEMENT_AUDIO_REC_OP:{
                        [_recvSwitch setOn:OperateValue.boolValue animated:true
                    }
                    break;
                }
            }
        }
    }
}
```

```
        case ITMG_ROOM_MANAGEMENT_MIC_OP:{
            [_micSwitch setOn:OperateValue.boolValue animated:true];
            [_sendSwitch setOn:OperateValue.boolValue animated:true];
        }
            break;
        case ITMG_ROOM_MANAGEMENT_SPEAKER_OP:{
            [_speakerSwitch setOn:OperateValue.boolValue animated:true];
            [_recvSwitch setOn:OperateValue.boolValue animated:true];
        }
            break;
        default:
            break;
    }

    if (OperateType.intValue == ITMG_ROOM_MANAGEMENT_GET_MIC_STATE || O
        NSString *alterString = [NSString stringWithFormat:@"id:%@",
            UIAlertView *alert = [[UIAlertView alloc] initWithTit
                [alert show];
        }
    else{
        NSString *alterString = [NSString stringWithFormat:@"id:%@",
            UIAlertView *alert = [[UIAlertView alloc] initWithTit
                [alert show];
        }
    }
}
break;
}
```