

# Game Multimedia Engine

시작하기

제품 문서



Tencent Cloud

## 저작권 고지

©2013-2025 Tencent Cloud. 모든 권리 보유.

본 문서의 저작권은 텐센트 클라우드(Tencent Cloud)에 단독으로 귀속됩니다. 텐센트 클라우드의 사전 서면 승인 없이는 어떠한 주체도 본 문서 내용의 전부 또는 일부를 복제·수정·표절·전송하는 등 어떠한 형태로도 이용할 수 없습니다.

## 상표 고지



텐센트 클라우드(Tencent Cloud) 및 관련 서비스의 모든 상표는 텐센트(Tencent) 그룹 산하 법인들(모회사, 자회사 및 계열사 포함)이 소유합니다. 본 문서에 언급된 제3자 상표는 해당 법적 권리자가 소유합니다.

## 서비스 고지

본 문서는 고객에게 텐센트 클라우드(Tencent Cloud) 제품 및 서비스 전부 또는 일부에 대한 현재적 개괄적 정보를 제공하기 위한 것으로, 특정 제품·서비스의 내용은 수시로 변경될 수 있습니다. 고객이 실제 구매한 제품·서비스의 적용 기준은 고객과 텐센트 클라우드간 체결된 상업계약에 명시된 내용이 우선하며, 별도 서면 합의가 없는 한 텐센트 클라우드는 본 문서 내용에 대해 법적 효력이 있는 명시적·묵시적 진술이나 보증을 일체 하지 않습니다.

## 목록:

시작하기

Quick Integration of SDK

Native SDK 빠른 통합

Unity SDK 빠른 통합

Unreal SDK 빠른 통합

Quick Integration of Sample Project

Unreal Sample Project 빠른 실행

# 시작하기

## Quick Integration of SDK

### Native SDK 빠른 통합

최종 업데이트 날짜: 2024-01-18 15:57:14

본문에서는 Native 프로젝트 개발자가 GME(Game Multimedia Engine)용 API를 쉽게 디버깅하고 통합할 수 있도록 자세한 설명을 제공합니다.

이 문서는 API 디버깅 및 통합을 위해 GME를 시작하는 데 도움이 되는 기본 API만 제공합니다.

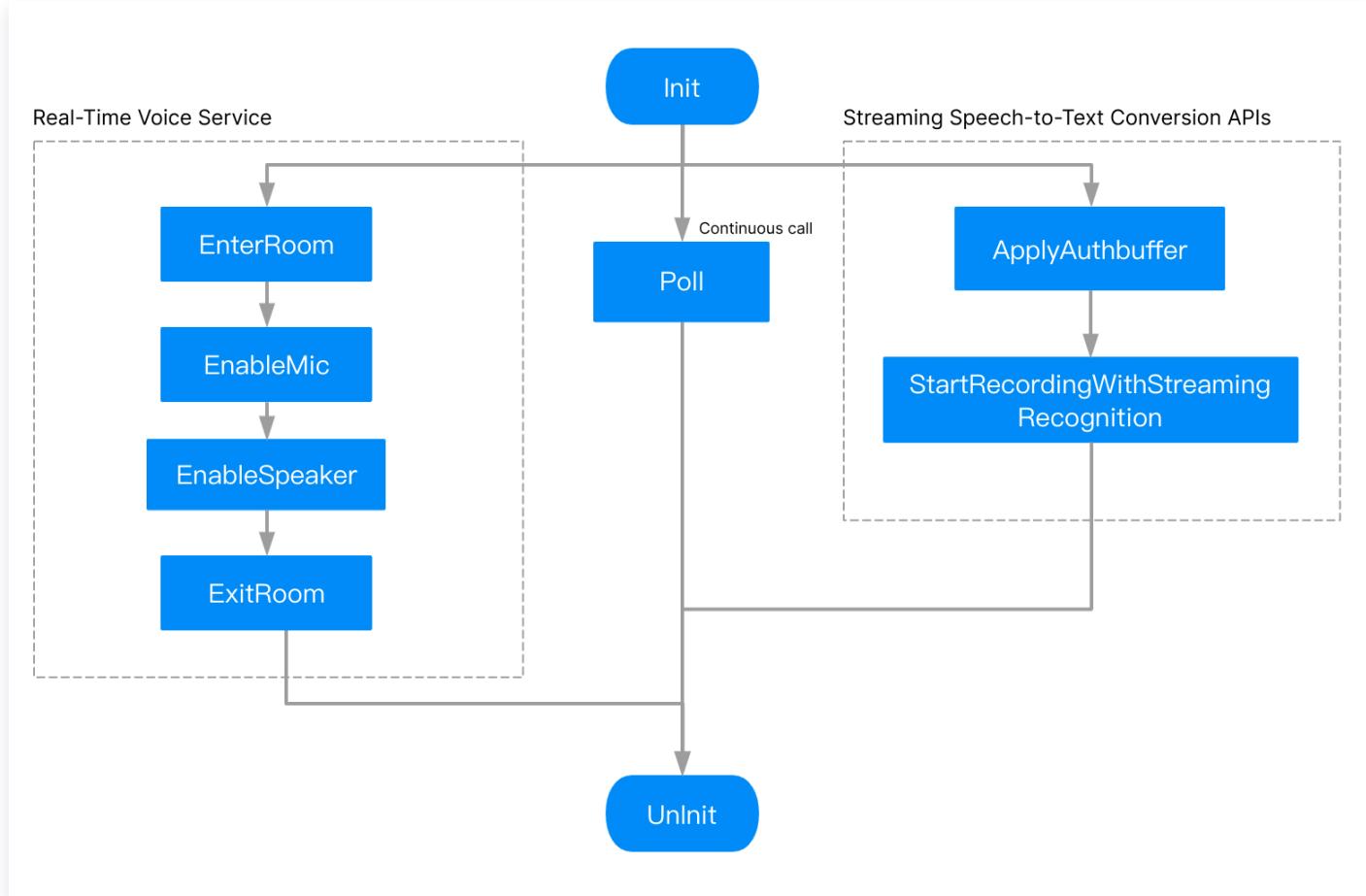
#### GME 사용 시 주의할 점

GME는 실시간 음성 채팅 서비스와 음성 메시지 및 음성-텍스트 변환 서비스의 두 가지 서비스를 제공하며 둘 다 Init 및 Poll과 같은 주요 API에 의존합니다.

##### ⚠️ Init API 정보

실시간 음성 채팅과 음성 메시지 서비스를 동시에 사용해야 하는 경우 Init API를 한 번만 호출하면 됩니다.

#### API 호출 순서도



## 액세스 단계

### 핵심 API

- GME 초기화 API: Init
- 이벤트 콜백을 트리거하기 위해 주기적으로 Poll 호출 API: Poll

### 실시간 음성 채팅

1. 음성 채팅방 입장, API: EnterRoom
2. 마이크 활성화 또는 비활성화, API: EnableMic
3. 스피커 활성화 또는 비활성화, API: EnableSpeaker
4. 음성 채팅방 퇴장, API: ExitRoom

### 음성 메시지

1. 인증 초기화, API: ApplyPTTAuthbuffer
  2. 스트리밍 음성 인식 시작, API: StartRecordingWithStreamingRecognition
  3. 녹화 중지, API: StopRecording
- GME 초기화 해제, API: UnInit

## 핵심 API 액세스

### 1. SDK 다운로드

SDK 다운로드 가이드 페이지에서 적절한 [클라이언트 SDK](#)를 다운로드합니다.

### 2. 헤더 파일 가져오기

Java

```
import com.tencent.TMG.IMGContext;
import com.tencent.av.sig.AuthBuffer;
import com.tencent.bugly.crashreport.CrashReport;
```

Object-C

```
#import "GMESDK/TMGEEngine.h"
#import "GMESDK/QAVAuthBuffer.h"
```

C++

```
#include "auth_buffer.h"
#include "tmg_sdk.h"
#include "AdvanceHeaders/tmg_sdk_adv.h"
#include <vector>
```

### 3. 싱글톤 가져오기

음성 기능을 사용하려면 ITMGContext 객체를 먼저 가져와야 합니다.

#### 함수 프로토 타입

## Java

```
public static ITMGContext GetInstance(Context context)
```

## Object-C

```
+ (ITMGContext*) GetInstance;
```

## C++

```
__UNUSED static ITMGContext* ITMGContextGetInstance () {
    return ITMGContextGetInstanceInner (TMG_SDK_VERSION);
}
```

예시 코드

## Java

```
//MainActivity.java
import com.tencent.TMG.ITMGContext;
ITMGContext tmgContext = ITMGContext.GetInstance(this);
```

## Object-C

```
//TMGSampleViewController.m
ITMGContext* _context = [ITMGContext GetInstance];
```

## C++

```
ITMGContext* context = ITMGContextGetInstance();
```

## 4. 콜백 설정

API 클래스는 Delegate 메소드를 사용하여 애플리케이션에 콜백 알림을 보냅니다. 콜백 메시지 수신을 위해 입장 전 SDK에 콜백 함수를 등록합니다.

### 함수 프로토타입 및 예시 코드

콜백 메시지 수신을 위해 입장 전에 SDK에 콜백 함수를 등록합니다.

#### Java

```
//ITMGContext
public abstract int SetTMGDelegate(ITMGDelegate delegate);

//MainActivity.java
tmgContext.SetTMGDelegate(TMGCallbackDispatcher.getInstance());
```

#### Object-C

```
ITMGDelegate < NSObject >

//TMGSampleViewController.m
ITMGContext* _context = [ITMGContext GetInstance];
_context.TMGDelegate = [DispatchCenter getInstance];
```

#### C++

```
//SDK 초기화 시
m_pTmgContext = ITMGContextGetInstance();
m_pTmgContext->SetTMGDelegate(this);
//소멸자 중
CTMGS�K_For_AudioDlg::~CTMGS�K_For_AudioDlg()
{
```

```
    ITMGContextGetInstance() -> SetTMGDelegate(NULL);  
}
```

## 콜백 예시

콜백의 매개변수를 처리하려면 생성자에서 이 콜백 함수를 재정의하십시오.

### Java

```
//MainActivity.java  
tmgContext.SetTMGDelegate(TMGCallbackDispatcher.getInstance());  
  
//RealTimeVoiceActivity.java  
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data)  
{  
    if (type == ITMG_MAIN_EVENT_TYPE_ENTER_ROOM)  
    {  
        //콜백 프로세스  
    }  
}  
  
//TMGCallbackDispatcher.java, TMGCallbackHelper.java 및  
TMGDispatcherBase.java 참고
```

### Object-C

```
// TMGRealTimeViewController.m  
TMGRealTimeViewController () < ITMGDelegate >  
  
- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary  
*)data {  
    NSString *log = [NSString stringWithFormat:@"OnEvent:%d,data:%@",  
    (int)eventType, data];  
    [self showLog:log];  
    NSLog(@"%@", log);  
    switch (eventType) {
```

```
// Step 6/11 : Perform the enter room event
case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM: {
    int result = ((NSNumber *) [data
objectForKey:@"result"]).intValue;
    NSString *error_info = [data objectForKey:@"error_info"];

    [self showLog:[NSString
stringWithFormat:@"OnEnterRoomComplete:%d msg:(%@)", result,
error_info]];

    if (result == 0) {
        [self updateStatusEnterRoom:YES];
    }
    break;
}

}

//DispatchCenter.h 및 DispatchCenter.m 참고
```

## C++

```
//헤더 파일 중 선언
virtual void OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char* data);
//예시 코드
void CTMGS�K_For_AudioDlg::OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const
char* data)
{
    switch(eventType)
    {
        case ITMG_MAIN_EVENT_TYPE_XXXX_XXXX:
        {
            //콜백 처리 진행
        }
        break;
    }
}
```

매개변수	유형	설명
type	ITMGContext.ITMG_MAIN_EVENT_TYPE	콜백 이벤트 유형
data	Intent 메시지 유형	콜백 메시지, 즉, 이벤트 데이터

## 5. SDK 초기화

실시간 음성, 음성 메시지, 음성 텍스트 변환 서비스를 사용하려면 먼저 **Init API**를 통해 SDK를 초기화해야 합니다. Init API는 다른 API와 동일한 스레드에서 호출해야 합니다. 기본 스레드에서 모든 API를 호출하는 것이 좋습니다.

### API 프로토 타입

#### Java

```
public abstract int Init(String sdkAppId, String openId);
```

#### Object-C

```
- (int)InitEngine: (NSString*) sdkAppID openID: (NSString*) openID;
```

#### C++

```
ITMGContext virtual int Init(const char* sdkAppId, const char* openId)
```

매개변수	유형	설명
sdkAppId	string	GME 콘솔에서 제공되는 AppID는 <a href="#">서비스 활성화</a> 에서 안내된 대로 얻을 수 있습니다.
openID	string	openID는 Int64 타입만 가능하며 문자열로 변환되어 전달됩니다. 해당 규칙을 사용자 정의할 수 있으며 App에서 고유해야 합니다. Openid를 문자열로 전달하기 위해서는 <a href="#">Submit Ticket</a> 하여 신청하십시오.

## 예시 코드

### Java

```
//MainActivity.java
int nRet = tmgContext.Init(appId, openId);
if (nRet == AV_OK)
{
    GMEAAuthBufferHelper.getInstance().setGEMParams(appId, key, openId);
    // Step 4/11: Poll to trigger callback
    //https://intl.cloud.tencent.com/document/product/607/40860
    EnginePollHelper.createEnginePollHelper();
    showToast("Init success");
} else if (nRet == AV_ERR_HAS_IN_THE_STATE) // SDK가 초기화되었습니다. 이 작업은 성공적입니다.
{
    showToast("Init success");
} else
{
    showToast("Init error errorCode:" + nRet);
}
```

### Object-C

```
//TMGSampleViewController.m
QAVResult result = [_context InitEngine:self.appIDTF.text
openID:self.openIDTF.text];
if (result == QAV_OK) {
    self.isSDKInit = YES;
}
```

### C++

```
#define SDKAPPID3RD "14000xxxxx"
const char* openId="10001";
```

```
ITMGContext* context = ITMGContextGetInstance();
context->Init(SDKAPPID3RD, openId);
```

## 6. 이벤트 콜백 트리거

이벤트 콜백은 update에서 Poll API를 주기적으로 호출하여 트리거할 수 있습니다. Poll API는 GME의 메시지 펌프이며 GME가 이벤트 콜백을 트리거하도록 주기적으로 호출해야 합니다. 그렇지 않으면 전체 SDK 서비스가 비정상적으로 실행됩니다. 자세한 내용은 [SDK 다운로드 가이드](#)의 EnginePollHelper 파일을 참고하십시오.

### 예시 코드

#### Java

```
//MainActivity.java
[EnginePollHelper createEnginePollHelper];

//EnginePollHelper.java
private Handler mHandler = new Handler();
private Runnable mRunnable = new Runnable() {
    @Override
    public void run() {
        if (s_pollEnabled) {
            if (ITMGContext.GetInstance(null) != null)
                ITMGContext.GetInstance(null).Poll();
        }
        mHandler.postDelayed(mRunnable, 33);
    }
};

//Poll을 주기적으로 호출하는 코드는 EnginePollHelper.java를 참고하십시오.
```

#### Object-C

```
//TMGSampleViewController.m
[EnginePollHelper createEnginePollHelper];
//EnginePollHelper.m 및 EnginePollHelper.h 참고
```

C++

```
void TMGTestScene::update(float delta)
{
    ITMGContextGetInstance() -> Poll();
}
```

## 7. 로컬 인증 키 계산

관련 기능의 암호화 및 인증을 위해 AuthBuffer를 생성합니다. 프로덕션 환경에서 릴리스하려면 [Authentication Key](#)에 설명된 대로 백엔드 배포 키를 사용하십시오.

### API 프로토 타입

Java

```
AuthBuffer public native byte[] genAuthBuffer(int sdkAppId, String
roomId, String openId, String key)
```

Object-C

```
// TMGSampleViewController.m
[EnginePollHelper createEnginePollHelper];
// EnginePollHelper.m 및 EnginePollHelper.h 참고
```

C++

```
void TMGTestScene::update(float delta)
{
    ITMGContextGetInstance() -> Poll();
}
```

매개변수	유형	설명
------	----	----

appId	int	Tencent Cloud 콘솔의 AppId.
roomId	string	방 ID, 최대 127자 지원(음성 메시지의 경우 null 입력).
openId	string	사용자 ID. Init 시 openid와 동일.
key	string	Tencent Cloud 콘솔의 권한 키.

## 예시 코드

### Java

```
//GMEAAuthBufferHelper.java
import com.tencent.av.sig.AuthBuffer; //헤더파일
public byte[] createAuthBuffer(String roomId)
{
    byte[] authBuffer;
    // Generate AuthBuffer for encryption and authentication of
    relevant features. For release in the production environment,
    // please use the backend deployment key as detailed in
    https://intl.cloud.tencent.com/document/product/607/12218
    if (TextUtils.isEmpty(roomId))
    {
        authBuffer =
AuthBuffer.getInstance().genAuthBuffer(Integer.parseInt(mAppId), "0",
mOpenId, mKey);
    } else
    {
        authBuffer =
AuthBuffer.getInstance().genAuthBuffer(Integer.parseInt(mAppId), roomId,
mOpenId, mKey);
    }
    return authBuffer;
}
```

### Object-C

```
//실시간 음성 채팅 인증
```

```
NSData * authBuffer = [QAVAuthBuffer GenAuthBuffer:SDKAPPID3RD.intValue
roomID:self.roomIdTF.text openID:_openId key:_key];
//음성 메시지 인증
NSData * authBuffer = [QAVAuthBuffer GenAuthBuffer:(unsigned
int)SDKAPPID3RD.integerValue roomID:nil openID:self.openId key:AUTHKEY];
```

C++

```
unsigned int bufferLen = 512;
unsigned char retAuthBuff[512] = {0};
QAVSDK_AuthBuffer_GenAuthBuffer(atoi(SDKAPPID3RD), roomId, "10001",
AUTHKEY, retAuthBuff, bufferLen);
```

## 실시간 음성 채팅 액세스

### 1. 방 입장

생성된 인증 정보로 방에 입장하기 위해 사용하는 API입니다. 마이크와 스피커는 방 입장 후 기본적으로 켜지지 않습니다. 반환된 AV\_OK 값은 API 호출은 성공했지만 방 입장은 성공하지 못했다는 의미입니다.

#### API 프로토타입

Java

```
public abstract int EnterRoom(String roomId, int roomType, byte[]
authBuffer);
```

Object-C

```
- (int)EnterRoom:(NSString *) roomId roomType:(int) roomType authBuffer:
(NSData *) authBuffer;
```

C++

```
ITMGContext virtual int EnterRoom(const char* roomID, ITMG_ROOM_TYPE  
roomType, const char* authBuff, int buffLen);
```

매개변수	유형	설명
roomId	String	방 ID, 최대 127자 지원
roomType	int	FLUENCY 음질을 사용하여 방에 입장하십시오
authBuffer	byte[]	인증 코드

## 예시 코드

### Java

```
//RealTimeVoiceActivity.java  
byte[] authBuffer =  
GMEAuthBufferHelper.getInstance().createAuthBuffer(roomId);  
ITMGContext.GetInstance(this).EnterRoom(roomId, roomType, authBuffer);
```

### Object-C

```
//TMGRealTimeViewController.m  
[[ITMGContext GetInstance] EnterRoom:self.roomIdTF.text roomType:  
(int)self.roomTypeControl.selectedSegmentIndex + 1  
authBuffer:authBuffer];
```

### C++

```
ITMGContext* context = ITMGContextGetInstance();  
context->EnterRoom(roomID, ITMG_ROOM_TYPE_FLUENCY,  
(char*)retAuthBuff,bufferLen);
```

## 방 입장 콜백

사용자가 방에 입장한 후 ITMG\_MAIN\_EVENT\_TYPE\_ENTER\_ROOM 메시지가 전송되고 콜백 및 처리를 위해 OnEvent 함수에서 식별됩니다. 성공적인 콜백은 방 입장 성공 및 **과금** 시작을 의미합니다.

### 과금 참고

- [Purchase Guide](#)
- [Billing](#)
- [음성 채팅 사용 시 클라이언트가 서버와 연결이 끊긴 경우에도 계속 과금됩니까?](#)

- **예시 코드**방 입장 및 네트워크 연결 해제 이벤트를 포함한 콜백 처리 예시 코드입니다.

#### Java

```
//RealTimeVoiceActivity.java
public void OnEvent (ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data)
{
    if (type == ITMG_MAIN_EVENT_TYPE_ENTER_ROOM)
    {
        // Step 6/11 : Perform the enter room event
        int nErrCode =
TMGCallbackHelper.ParseIntentParams2 (data) .nErrCode;
        String strMsg =
TMGCallbackHelper.ParseIntentParams2 (data) .strErrMsg;
        if (nErrCode == AV_OK)
        {
            appendLog2MonitorView ("EnterRomm success");
        } else
        {
            appendLog2MonitorView (String.format (Locale.getDefault (),
"EnterRomm errCode:%d errMsg:%s", nErrCode, strMsg));
        }
    }
}
```

#### Object-C

```
// TMGRealTimeViewController.m
```

```
- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary
*)data {
    NSString *log = [NSString stringWithFormat:@"OnEvent:%d,data:%@",
(int)eventType, data];
    [self showLog:log];
    NSLog(@"%@", log);
    switch (eventType) {
        // Step 6/11 : Perform the enter room event
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM: {
            int result = ((NSNumber *) [data
objectForKey:@"result"])).intValue;
            NSString *error_info = [data objectForKey:@"error_info"];

            [self showLog:[NSString
stringWithFormat:@"OnEnterRoomComplete:%d msg:(%@)", result,
error_info]];

            if (result == 0) {
                [self updateStatusEnterRoom:YES];
            }
            break;
        }
    }
}
```

## C++

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char*
data) {
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
            ListMicDevices();
            ListSpeakerDevices();
            std::string strText = "EnterRoom complete: ret=";
            strText += data;
            m_EditMonitor.SetWindowText(MByteToWChar(strText).c_str());
        }
    }
}
```

### • 에러 코드

에러 코드 값	원인 및 권장 방안
7006	<p>인증 실패. 가능한 원인들:</p> <ul style="list-style-type: none"><li>• AppID 가 존재하지 않거나 올바르지 않음</li><li>• authbuff를 인증하는 동안 오류 발생</li><li>• 인증 만료</li><li>• openid 사양 불충족</li></ul>
7007	이미 다른 방에 있음
1001	사용자는 이미 방에 들어가는 과정에 있었지만 이 작업을 반복했습니다. 방 입장 콜백이 반환될 때까지 방 입장 API를 호출하지 않는 것이 좋습니다.
1003	사용자가 이미 방에 있었고 API를 입력하는 방을 다시 호출함
1101	SDK가 초기화되었는지, openid가 규칙을 준수하는지, API가 동일한 스레드에서 호출되는지, Poll API가 정상적으로 호출되는지 확인

## 2. 마이크 활성화 또는 비활성화

이 API는 마이크를 활성화/비활성화하는 데 사용됩니다. 방 입장 후 마이크와 스피커는 기본적으로 활성화되어 있지 않습니다.

### 예시 코드

Java

## Object-C

C++

```
ITMGContextGetInstance() -> GetAudioCtrl() -> EnableMic(true);
```

### 3. 스피커 활성화 또는 비활성화

이 API는 스피커를 활성화/비활성화하는 데 사용됩니다.

#### 예시 코드

Java

```
//RealTimeVoiceActivity.java
ITMGContext.GetInstance(this).GetAudioCtrl().EnableSpeaker(true);
```

Object-C

```
//TMGRealTimeViewController.m
[[[ITMGContext GetInstance] GetAudioCtrl] EnableSpeaker:YES];
```

C++

```
ITMGContextGetInstance() -> GetAudioCtrl() -> EnableSpeaker(true);
```

### 4. 방 퇴장

이 API는 현재 방을 나가기 위해 호출됩니다. 종료를 위해 콜백을 기다리고 처리해야 합니다.

#### 예시 코드

Java

```
//RealTimeVoiceActivity.java
ITMGContext.GetInstance(this).ExitRoom();
```

### Object-C

```
//TMGRealTimeViewController.m
[ [ITMGContext GetInstance] ExitRoom];
```

### C++

```
ITMGContext* context = ITMGContextGetInstance();
context->ExitRoom();
```

## 방 퇴장 콜백

사용자가 방을 나가면 ITMG\_MAIN\_EVENT\_TYPE\_EXIT\_ROOM이라는 메시지와 함께 콜백이 반환됩니다. 예시 코드는 아래와 같습니다.

### Java

```
//RealTimeVoiceActivity.java
public void OnEvent (ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data)
{
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_EXIT_ROOM
== type)
    {
        //방 퇴장 성공 이벤트 수신
    }
}
```

### Object-C

```
//TMGRealTimeViewController.m
```

```
- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@", (unsigned long)eventType, data);
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_EXIT_ROOM:
        {
            //방 퇴장 성공 이벤트 수신
        }
        break;
    }
}
```

C++

```
void TMGTestScene::OnEvent (ITMG_MAIN_EVENT_TYPE eventType, const char* data) {
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_EXIT_ROOM:
        {
            //프로세스
            break;
        }
    }
}
```

## 음성 메시지 액세스

### 1. 인증 초기화

SDK 초기화 후 인증 초기화를 호출합니다. authBuffer를 얻는 방법에 대한 자세한 내용은 genAuthBuffer(실시간 음성 채팅 인증 정보 API)를 참고하십시오.

#### API 프로토타입

Java

```
public abstract int ApplyPTTAuthbuffer(byte[] authBuffer);
```

## Object-C

```
- (QAVResult)ApplyPTTAuthbuffer:(NSData *)authBuffer;
```

## C++

```
ITMGPTT virtual int ApplyPTTAuthbuffer(const char* authBuffer, int authBufferLen)
```

매개변수	유형	의미
authBuffer	String	인증

예시 코드

## Java

```
//VoiceMessageRecognitionActivity.java
byte[] authBuffer =
GMEAAuthBufferHelper.getInstance().createAuthBuffer("");
ITMGContext.GetInstance(this).GetPTT().ApplyPTTAuthbuffer(authBuffer);
```

## Object-C

```
//TMGPTTViewController.m
NSData* authBuffer = [QAVAuthBuffer GenAuthBuffer:(unsigned
int)SDKAPPID3RD.integerValue roomID:nil openID:self.openId key:AUTHKEY];
[[[ITMGContext GetInstance] GetPTT] ApplyPTTAuthbuffer:authBuffer];
```

## C++

```
ITMGContextGetInstance() -> GetPTT() ->  
> ApplyPTTAuthbuffer(authBuffer, authBufferLen);
```

## 2. 스트리밍 음성 인식 시작

이 API는 스트리밍 음성 인식을 시작하는 데 사용됩니다. 음성-텍스트 변환에서 얻은 텍스트는 콜백에서 실시간으로 반환됩니다. 녹음을 중지하려면 `StopRecording`을 호출합니다. 녹음이 중지된 후 콜백이 반환됩니다.

### API 프로토 타입

Java

```
public abstract int StartRecordingWithStreamingRecognition (String  
filePath);  
  
public abstract int StopRecording();
```

Object-C

```
- (int)StartRecordingWithStreamingRecognition: (NSString *)filePath;  
  
- (QAVResult) StopRecording;
```

C++

```
ITMGPTT virtual int StartRecordingWithStreamingRecognition (const char*  
filePath)  
  
ITMGPTT virtual int StopRecording()
```

매개변수	유형	설명
filePath	String	저장된 오디오 파일의 경로

## 예시 코드

### Java

```
//VoiceMessageRecognitionActivity.java
ITMGContext.GetInstance(this).GetPTT().StartRecordingWithStreamingRecognition(recordfilePath);
```

### Object-C

```
//TMGPTTViewController.m
QAVResult ret = [[[ITMGContext GetInstance] GetPTT]
StartRecordingWithStreamingRecognition:[self pttTestPath]];
if (ret == 0) {
    self.currentStatus = @"스트리밍 녹음 시작";
} else {
    self.currentStatus = @"스트리밍 녹음 시작 실패";
}
```

### C++

```
ITMGContextGetInstance() ->GetPTT() ->
StartRecordingWithStreamingRecognition(filePath);
```

## 스트리밍 음성 인식 콜백

스트리밍 음성 인식이 시작된 후 콜백 함수 OnEvent에서 콜백 메시지를 수신 대기해야 합니다. 이벤트 메시지는 ITMG\_MAIN\_EVNET\_TYPE\_PTT\_STREAMINGRECOGNITION\_COMPLETE 입니다. 즉, 녹화가 중지되고 인식이 완료된 후 텍스트를 반환합니다. 이는 한 단락을 말한 이후에 인식된 텍스트를 반환하는 것과 같습니다. 이벤트 메시지는 실제 필요에 따라 OnEvent 함수에서 식별됩니다. 전달된 매개변수에는 다음 네 가지 메시지가 포함됩니다.

메시지 이름	설명
result	스트리밍 음성 인식 성공 여부를 판단하기 위한 반환 코드

text	음성에서 변환된 텍스트
file_path	저장된 녹음 파일의 로컬 경로
file_id	90일 동안 보관되는 녹음 파일의 백엔드 url 주소

### ● 예시 코드

#### Java

```
//VoiceMessageRecognitionActivity.java
import static
com.tencent.TMG.ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE;
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data)
{
    if (type == ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE)
    {
        // Step 1.3/3 handle
        ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE event
        mIsRecording = false;
        if (nErrCode == 0)
        {
            String recordfilePath =
data.getStringExtra("file_path");
            mRecFilePathView.setText(recordfilePath);

            String recordFileUrl = data.getStringExtra("file_id");
            mRecFileUrlView.setText(recordFileUrl);
        }
        else
        {
            appendLog2MonitorView("Record and recognition fail
errCode:" + nErrCode);
        }
    }
}
```

## Object-C

```
//TMGPTTViewController.m

- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary*)data
{
    NSNumber *number = [data objectForKey:@"result"];
    switch (eventType)
    {
        case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE:
        {
            if (data != NULL && [[data objectForKey:@"result"]
intValue]== 0)
            {
                self.translateTF.text = [data objectForKey:@"text"];
                self.currentStatus = @"스트리밍 변환 완료";
            }
        }
        break;
    }
}
```

## C++

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char*
data) {
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE:
        {
            HandleSTREAM2TEXTComplete(data, true);
            break;
        }
        ...
        case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING:
        {
            HandleSTREAM2TEXTComplete(data, false);
            break;
        }
    }
}
```

```
    }

    void CTMGSDK_For_AudioDlg::HandleSTREAM2TEXTComplete(const char* data,
bool isComplete)
{
    std::string strText = "STREAM2TEXT: ret=";
    strText += data;
    m>EditMonitor.SetWindowText(MByteToWChar(strText).c_str());
    Json::Reader reader;
    Json::Value root;
    bool parseRet = reader.parse(data, root);
    if (!parseRet) {

        ::SetWindowText(m>EditInfo.GetSafeHwnd(), MByteToWChar(std::string("parse
result Json error")).c_str());
    }
    else
    {
        if (isComplete) {

            ::SetWindowText(m>EditUpload.GetSafeHwnd(),
MByteToWChar(root["file_id"].asString()).c_str());
        }
        else{
            std::string isruning =
"STREAMINGRECOGNITION_IS_RUNNING";

            ::SetWindowText(m>EditUpload.GetSafeHwnd(),
MByteToWChar(isruning).c_str());
        }
    }
}
```

## ● 에러 코드

에러 코드	설명	제안된 솔루션
32775	스트리밍 음성을 텍스트로 변환하는데 실패했지만 녹음은 성공함	UploadRecordedFile API를 호출하여 녹음 파일을 업로드한 다음 SpeechToText API를 호출하

		여 음성을 텍스트로 변환
32777	스트리밍 음성을 텍스트로 변환하는 데 실패했지만 녹음 및 업로드는 성공함	반환된 메시지에는 업로드 성공 후 백엔드 url이 포함되어 있으며, SpeechToText API를 호출하여 음성을 텍스트로 변환
32786	스트리밍 음성-텍스트 변환 실패	스트리밍 녹화 중 스트리밍 녹화 API의 실행 결과가 반환될 때까지 기다리십시오

### 3. 녹음 중지

이 API는 녹음을 중지하는 데 사용됩니다. 비동기식이며 녹음이 중지된 후 녹음 완료에 대한 콜백이 반환됩니다. 녹음 파일은 녹음이 성공한 후에만 사용할 수 있습니다.

#### API 프로토타입

##### Java

```
public abstract int StopRecording();
```

##### Object-C

```
- (QAVResult) StopRecording;
```

##### C++

```
ITMGPTT virtual int StopRecording();
```

### 예시 코드

##### Java

```
//VoiceMessageRecognitionActivity.java
ITMGContext.GetInstance(this).GetPTT().StopRecording();
```

## Object-C

```
//TMGPTTViewController.m

- (void)stopRecClick {
    // Step 3/12 stop recording,  need handle
    ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE event
    // https://intl.cloud.tencent.com/document/product/607/15221
    QAVResult ret = [[[ITMGContext GetInstance] GetPTT] StopRecording];
    if (ret == 0) {
        self.currentStatus = @"녹음 중지";
    } else {
        self.currentStatus = @"녹음 중지 실패";
    }
}
```

## C++

```
ITMGContextGetInstance()->GetPTT()->StopRecording();
```

# Unity SDK 빠른 통합

최종 업데이트 날짜:: 2025-10-16 19:32:56

본문에서는 Unity 프로젝트 개발자가 GME(Game Multimedia Engine)용 API를 쉽게 디버깅하고 통합할 수 있도록 자세한 설명을 제공합니다.

이 문서는 API 디버깅 및 통합을 위해 GME를 시작하는 데 도움이 되는 기본 API만 제공합니다.

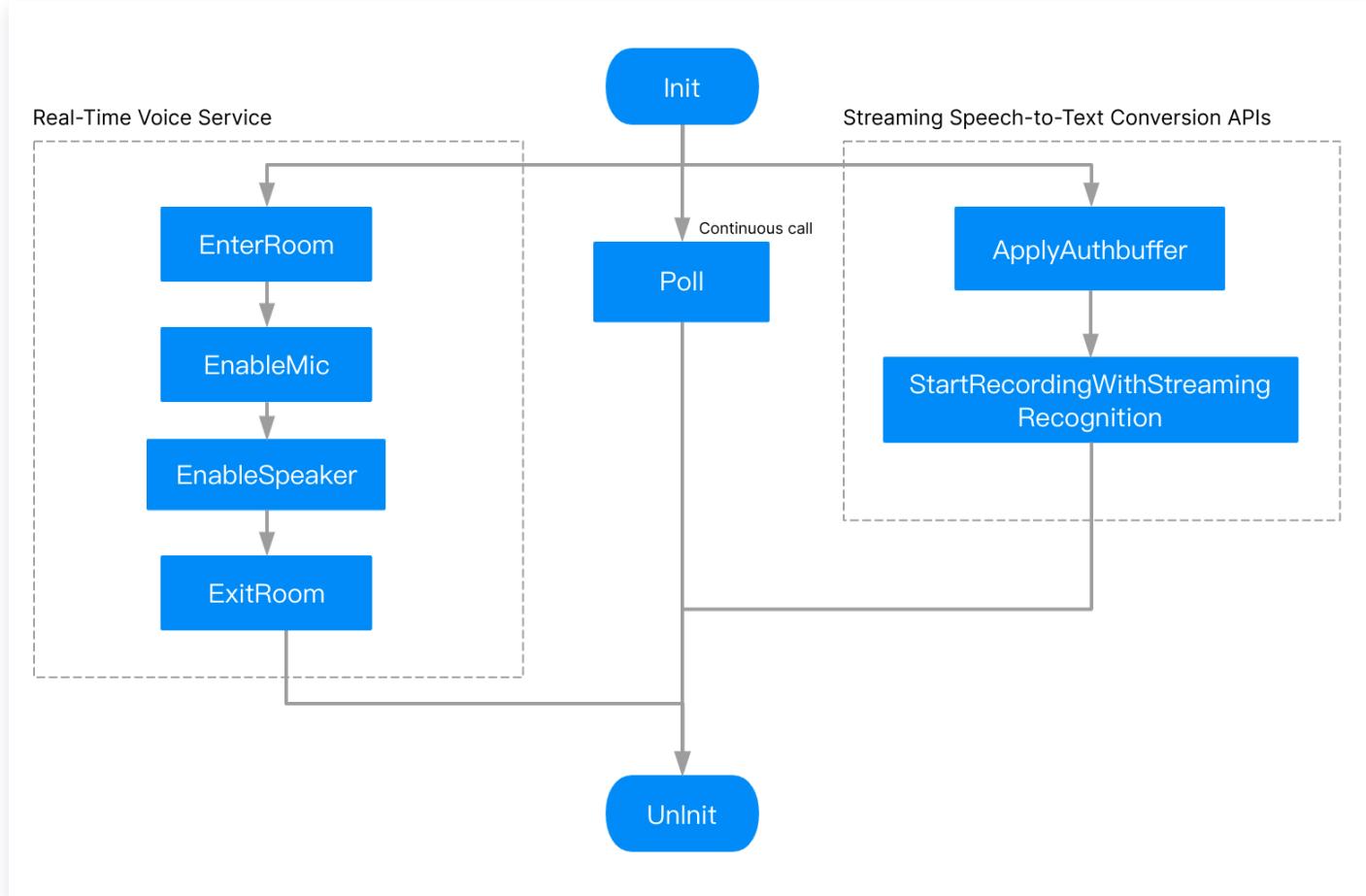
## GME 사용 시 주의할 점

GME는 실시간 음성 채팅 서비스와 음성 메시지 및 음성-텍스트 변환 서비스의 두 가지 서비스를 제공하며 둘 다 `Init` 및 `Poll`과 같은 주요 API에 의존합니다.

### **Init API 정보**

실시간 음성 채팅과 음성 메시지 서비스를 동시에 사용해야 하는 경우 `Init API`를 한 번만 호출하면 됩니다.

## API 호출 순서도



## 액세스 단계

### SDK 통합

SDK를 프로젝트에 통합하려면 [Integrating SDK](#)를 참고하십시오.

### 주요 API

- GME 초기화, API: Init
- 이벤트 콜백을 트리거하기 위해 주기적으로 Poll 호출, API: Poll
- 방 입장/퇴장 알림 수신, 수신: QAVEnterRoomComplete

### 실시간 음성 채팅

1. 음성 채팅방 입장, API: EnterRoom
2. 마이크 활성화 또는 비활성화, API: EnableMic

3. 스피커 활성화 또는 비활성화, API: EnableSpeaker
4. 음성 채팅방 퇴장, API: ExitRoom

## 음성 메시지

1. 인증 초기화, API: ApplyPTTAuthbuffer
2. 스트리밍 음성 인식 시작, API: StartRecordingWithStreamingRecognition
3. 녹화 중지, API: StopRecording
4. GME 초기화 해제, API: UnInit

## 주요 API 액세스

### 1. SDK 다운로드

SDK 다운로드 가이드 페이지에서 적절한 [클라이언트 SDK](#)를 다운로드합니다.

### 2. 헤더 파일 가져오기

```
using GME;
```

### 3. Context 인스턴스 가져오기

QAVContext.GetInstance() 대신 ITMGContext 메소드를 사용하여 Context 인스턴스를 가져옵니다.

### 예시 코드

```
int ret = ITMGContext.GetInstance().Init(sdkAppId, openID);
```

### 4. SDK 초기화

실시간 음성, 음성 메시지, 음성 텍스트 변환 서비스를 사용하려면 먼저 **Init API**를 통해 SDK를 초기화해야 합니다. Init API는 다른 API와 동일한 스레드에서 호출해야 합니다. 기본 스레드에서 모든 API를 호출하는 것이 좋습니다.

### API 프로토 타입

```
//class ITMGContext
public abstract int Init(string sdkAppID, string openID);
```

매개변수	유형	설명

sdkAppId	string	GME 콘솔에서 제공되는 AppID는 서비스 활성화에 안내된 대로 얻을 수 있습니다.
openID	string	openID는 Int64 유형만 가능하며 문자열로 변환되어 전달됩니다. 해당 규칙을 사용자 정의할 수 있으며 App에서 고유해야 합니다. Openid를 문자열로 전달하려면 <a href="#">Submit Ticket</a> 하여 신청하십시오.

## 예시 코드

```
int ret = ITMGContext.GetInstance().Init(sdkAppId, openID);
// 반환된 값으로 초기화 성공 여부 판단
if (ret != QAVError.OK)
{
    Debug.Log("SDK 초기화 실패: " + ret);
    return;
}
```

## 5. 이벤트 콜백 트리거

이벤트 콜백은 update에서 Poll API를 주기적으로 호출하여 트리거할 수 있습니다. Poll API는 GME의 메시지 펌프이며 GME가 이벤트 콜백을 트리거하도록 주기적으로 호출해야 합니다. 그렇지 않으면 전체 SDK 서비스가 비정상적으로 실행됩니다. 자세한 내용은 [SDK 다운로드 가이드](#)의 EnginePollHelper 파일을 참고하십시오.

## 예시 코드

```
public void Update()
{
    ITMGContext.GetInstance().Poll();
}
```

## 6. 방 입장/퇴장 알림 수신

### 방 입장 알림

```
//위임 함수:
public delegate void QAVEnterRoomComplete(int result, string
error_info);
//이벤트 트리거 함수:
public abstract event QAVEnterRoomComplete OnEnterRoomCompleteEvent;
```

## 방 퇴장 알림

위임 함수:

```
ublic delegate void QAVExitRoomComplete();
```

이벤트 트리거 함수:

```
public abstract event QAVExitRoomComplete OnExitRoomCompleteEvent;
```

## 7. 로컬 인증 키 계산

관련 기능의 암호화 및 인증을 위해 AuthBuffer를 생성합니다. 프로덕션 환경에서 릴리스하려면 [Authentication Key](#)에 설명된 대로 백엔드 배포 키를 사용하십시오.

### API 프로토타입

```
QAVAuthBuffer GenAuthBuffer(int appId, string roomId, string openId,
string key)
```

매개변수	유형	설명
appId	int	Tencent Cloud 콘솔의 AppId.
roomId	string	방 ID, 최대 127자 지원(음성 메시지의 경우 null 입력).
openId	string	사용자 ID. Init 시 openId와 동일.
key	string	Tencent Cloud 콘솔의 권한 키.

### 예시 코드

```
public static byte[] GetAuthBuffer(string AppID, string RoomID, string
OpenId, string AuthKey) {
    return QAVAuthBuffer.GenAuthBuffer(int.Parse(AppID), RoomID, OpenId,
AuthKey);
}
```

## 실시간 음성 채팅 액세스

### 1. 방 입장

생성된 인증 정보로 방에 입장하기 위해 사용하는 API입니다. 방 입장 후 마이크와 스피커는 기본적으로 활성화되지 않습니다. 반환된 AV\_OK 값은 API 호출은 성공했지만 방 입장은 성공하지 못했다는 의미입니다.

## API 프로토타입

```
ITMGContext EnterRoom(string roomId, int roomType, byte[] authBuffer)
```

매개변수	유형	설명
roomId	String	방 ID, 최대 127자 지원
roomType	ITMGRoomType	ITMGRoomType.ITMG_ROOM_TYPE_FLUENCY 입력
authBuffer	byte[]	인증 코드

## 예시 코드

```
ITMGContext.GetInstance().EnterRoom(strRoomId,
ITMGRoomType.ITMG_ROOM_TYPE_FLUENCY, byteAuthbuffer);
```

## 방 입장 콜백

사용자가 방에 들어가면 방 입장 결과가 다시 호출되어 처리를 위해 들을 수 있습니다. 성공적인 콜백은 방 입장이 성공적이며 과금이 시작됨을 의미합니다.

### 과금 참고

- [Purchase Guide](#)
- [Billing](#)
- [음성 채팅 사용 시 클라이언트가 서버와 연결이 끊긴 경우에도 계속 과금됩니까?](#)

- 예시 코드 콜백 처리를 위한 예시 코드입니다.

```
//이벤트 수신:
ITMGContext.GetInstance().OnEnterRoomCompleteEvent += new
QAVEnterRoomComplete(OnEnterRoomComplete);

//수신된 이벤트 처리:
void OnEnterRoomComplete(int err, string errInfo)
{
    if (err != 0) {
        ShowLoginPanel("에러 코드:" + err + " 에러 메시지:" + errInfo);
```

```

        return;
    }
    else{
        //방 입장 성공
    }
}

```

### • 예러 코드

에러 코드 값	원인 및 권장 방안
7006	인증 실패. 가능한 원인들: <ul style="list-style-type: none"> <li>• AppID 가 존재하지 않거나 올바르지 않음</li> <li>• authbuff를 인증하는 동안 오류 발생</li> <li>• 인증 만료</li> <li>• openid 사양 불충족</li> </ul>
7007	이미 다른 방에 있음
1001	사용자는 이미 방에 들어가는 과정에 있었지만 이 작업을 반복했습니다. 방 입장 콜백이 반환될 때까지 방 입장 API를 호출하지 않는 것이 좋습니다.
1003	사용자가 이미 방에 있었고 API를 입력하는 방을 다시 호출함
1101	SDK가 초기화되었는지, openid가 규칙을 준수하는지, API가 동일한 스레드에서 호출되는지, Poll API가 정상적으로 호출되는지 확인

## 2. 마이크 활성화 또는 비활성화

이 API는 마이크를 활성화/비활성화하는 데 사용됩니다. 방 입장 후 마이크와 스피커는 기본적으로 활성화되어 있지 않습니다.

### 예시 코드

```

//이벤트 수신:
ITMGContext.GetInstance().OnEnterRoomCompleteEvent += new
QAVEnterRoomComplete(OnEnterRoomComplete);

//수신된 이벤트 처리:
void OnEnterRoomComplete(int err, string errInfo)
{
    if (err != 0) {
        ShowLoginPanel("에러 코드:" + err + " 에러 메시지:" + errInfo);
        return;
    }
}

```

```
    }  
    else{  
        //방 입장 성공  
        //마이크 활성화  
        ITMGContext.GetInstance().GetAudioCtrl().EnableMic(true);  
    }  
}
```

### 3. 스피커 활성화 또는 비활성화

이 API는 스피커를 활성화/비활성화하는 데 사용됩니다.

#### 예시 코드

```
//이벤트 수신:  
ITMGContext.GetInstance().OnEnterRoomCompleteEvent += new  
QAVEnterRoomComplete(OnEnterRoomComplete);  
//수신된 이벤트 처리:  
void OnEnterRoomComplete(int err, string errInfo)  
{  
    if (err != 0) {  
        ShowLoginPanel("에러 코드:" + err + " 에러 메시지:" + errInfo);  
        return;  
    }  
    else{  
        //방 입장 성공  
        //스피커 활성화  
        ITMGContext.GetInstance().GetAudioCtrl().EnableSpeaker(true);  
    }  
}
```

### 4. 방 퇴장

이 API는 현재 방을 나가기 위해 호출됩니다. 종료를 위해 콜백을 기다리고 처리해야 합니다.

#### 예시 코드

```
ITMGContext.GetInstance().ExitRoom();
```

#### 방 퇴장 콜백

사용자가 방을 나가면 콜백이 반환됩니다. 예시 코드는 아래와 같습니다.

이벤트 수신:

```
ITMGContext.GetInstance().OnExitRoomCompleteEvent += new  
QAVExitRoomComplete(OnExitRoomComplete);
```

수신된 이벤트 처리:

```
void OnExitRoomComplete()  
{//방 퇴장 후 콜백 전송  
}
```

## 음성 메시지 액세스

### 1. 인증 초기화

SDK 초기화 후 인증 초기화를 호출합니다. authBuffer를 얻는 방법에 대한 자세한 내용은 genAuthBuffer(실시간 음성 채팅 인증 정보 API)를 참고하십시오.

#### API 프로토타입

```
ITMGPTT int ApplyPTTAuthbuffer (byte[] authBuffer)
```

매개변수	유형	설명
authBuffer	String	인증

#### 예시 코드

```
UserConfig.SetAppID(transform.Find ("appId").GetComponent<InputField>  
().text);  
UserConfig.SetUserID(transform.Find ("userId").GetComponent<InputField>  
().text);  
UserConfig.SetAuthKey(transform.Find("authKey").GetComponent<InputField>  
().text);  
byte[] authBuffer = UserConfig.GetAuthBuffer(UserConfig.GetAppID(),  
UserConfig.GetUserID(), null, UserConfig.GetAuthKey());  
ITMGContext.GetInstance ().GetPttCtrl ().ApplyPTTAuthbuffer(authBuffer);
```

### 2. 스트리밍 음성 인식 시작

이 API는 스트리밍 음성 인식을 시작하는 데 사용됩니다. 음성-텍스트로 변환에서 얻은 텍스트는 콜백에서 실시간으로 반환됩니다. 녹음을 중지하려면 **StopRecording**을 호출합니다. 녹음이 중지된 후 콜백이 반환됩니다.

## API 프로토타입

```
ITMGPTT int StartRecordingWithStreamingRecognition(string filePath)
```

매개변수	유형	설명
filePath	String	저장된 오디오 파일의 경로

## 예시 코드

```
string recordPath = Application.persistentDataPath +
string.Format("/{0}.silk", sUid++);
int ret =
ITMGContext.GetInstance().GetPttCtrl().StartRecordingWithStreamingRecognition(recordPath);
```

## 스트리밍 음성 인식 콜백

스트리밍 음성 인식이 시작된 후 `OnStreamingSpeechComplete` 또는 `OnStreamingSpeechisRunning` 알림에서 콜백 메시지를 수신해야 합니다. 자세한 내용은 아래와 같습니다.

- `OnStreamingSpeechComplete` 은 녹화가 중지되고 인식이 완료된 후 텍스트를 반환합니다. 이는 음성 단락 이후에 인식된 텍스트를 반환하는 것과 같습니다.
- `OnStreamingSpeechisRunning` 은 녹음 중 실시간으로 인식된 텍스트를 반환합니다. 이는 말하는 동안 인식된 텍스트를 반환하는 것과 같습니다.

이벤트 메시지는 실제 필요에 따라 `OnEvent` 함수에서 식별됩니다. 전달된 매개변수에는 다음 네 가지 메시지가 포함됩니다.

메시지 이름	설명
result	스트리밍 음성 인식 성공 여부를 판단하기 위한 반환 코드
text	음성에서 변환된 텍스트
file_path	저장된 녹음 파일의 로컬 경로
file_id	90일 동안 보관되는 녹음 파일의 백엔드 url 주소

### 예시 코드

```

//이벤트 수신:
ITMGContext.GetInstance().GetPttCtrl().OnStreamingSpeechComplete
+=new QAVStreamingRecognitionCallback (OnStreamingSpeechComplete);
ITMGContext.GetInstance().GetPttCtrl().OnStreamingSpeechisRunning +=
new QAVStreamingRecognitionCallback (OnStreamingRecisRunning);
//수신된 이벤트 처리:
void OnStreamingSpeechComplete(int code, string fileid, string
filepath, string result){
    //스트리밍 음성 인식을 위한 콜백
}
void OnStreamingRecisRunning(int code, string fileid, string filePath,
string result){
    if (code == 0)
    {
        setBtnText (mStreamBtn, "스트리밍");
        InputField field =
transform.Find("recordFilePath").GetComponent<InputField>();
        field.text = filePath;
        field =
transform.Find("downloadUrl").GetComponent<InputField>();
        field.text = "Stream is Running";
        field =
transform.Find("convertTextResult").GetComponent<InputField>();
        field.text = result;
        showWarningText ("녹화 중");
    }
}

```

## • 예러 코드

에러 코드	설명	제안된 솔루션
32775	스트리밍 음성을 텍스트로 변환하는데 실패했지만 녹음은 성공함	UploadRecordedFile API를 호출하여 녹음 파일을 업로드한 다음 SpeechToText API를 호출하여 음성을 텍스트로 변환
32777	스트리밍 음성을 텍스트로 변환하는데 실패했지만 녹음 및 업로드는 성공함	반환된 메시지에는 업로드 성공 후 백엔드 url이 포함되어 있으며, SpeechToText API를 호출하여 음성을 텍스트로 변환
32786	스트리밍 음성-텍스트 변환 실패	스트리밍 녹화 중 스트리밍 녹화 API의 실행 결

과가 반환될 때까지 기다리십시오

### 3. 녹음 중지

이 API는 녹음을 중지하는 데 사용됩니다. 비동기식이며 녹음이 중지된 후 녹음 완료에 대한 콜백이 반환됩니다. 녹음 파일은 녹음이 성공한 후에만 사용할 수 있습니다.

#### API 프로토타입

```
ITMGPTT int StopRecording()
```

#### 예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().StopRecording();
```

# Unreal SDK 빠른 통합

최종 업데이트 날짜: 2024-01-18 15:57:14

본문에서는 Unreal Engine 프로젝트 개발자가 GME(Game Multimedia Engine)용 API를 쉽게 디버깅하고 통합할 수 있도록 자세한 설명을 제공합니다.

이 문서는 API 디버깅 및 통합을 위해 GME를 시작하는 데 도움이 되는 기본 API만 제공합니다.

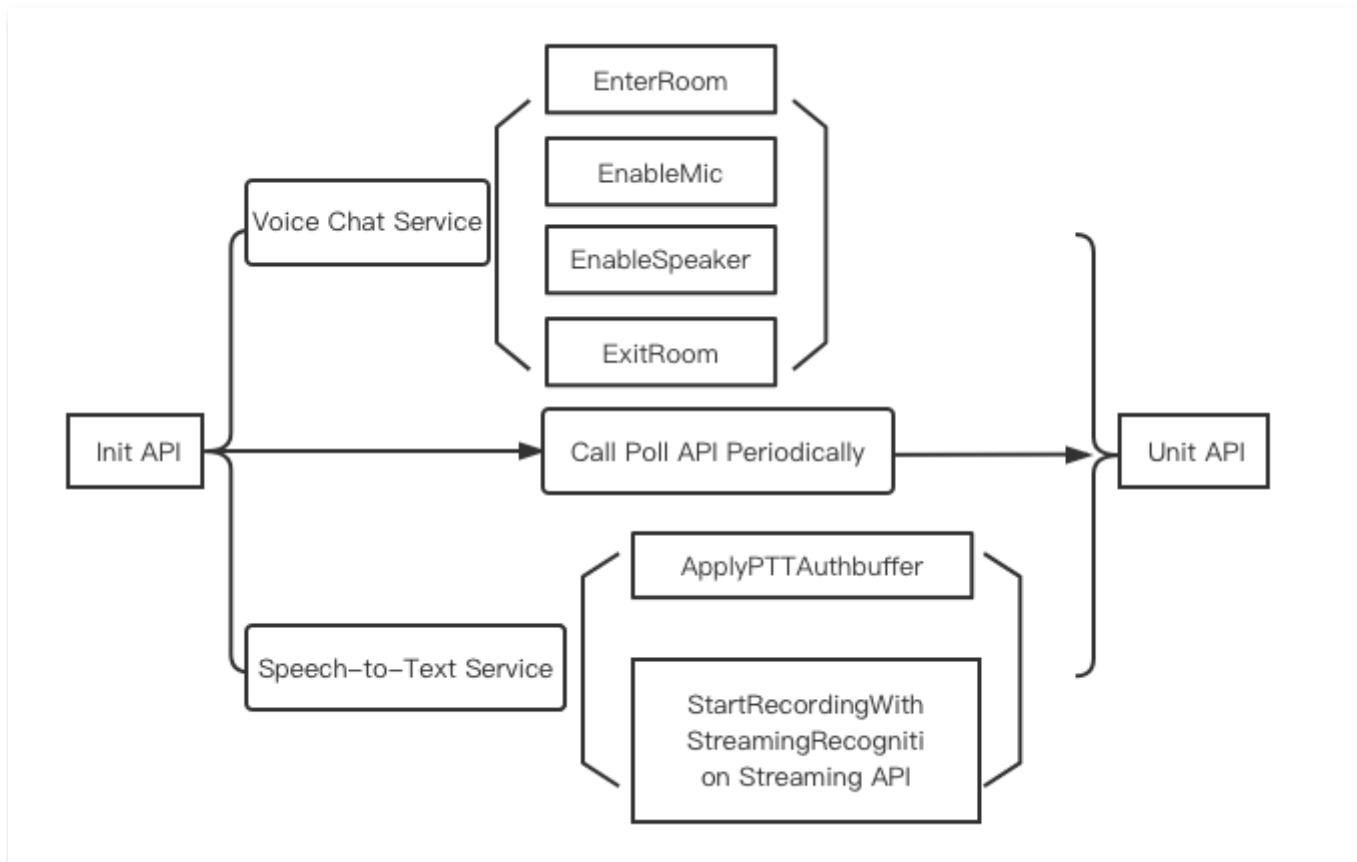
## GME 사용 시 주의할 점

GME는 실시간 음성 채팅 서비스와 음성 메시지 및 음성-텍스트 변환 서비스의 두 가지 서비스를 제공하며 둘 다 Init 및 Poll과 같은 주요 API에 의존합니다.

### ⚠️ Init API 정보

실시간 음성 채팅과 음성 메시지 서비스를 동시에 사용해야 하는 경우 **Init API**를 한 번만 호출하면 됩니다.

## API 호출 순서도



## 액세스 단계

### SDK 통합

SDK를 프로젝트에 통합하려면 [SDK for Unreal Engine](#)을 참고하십시오.

## 주요 API

- GME 초기화API: `Init`
- 이벤트 콜백을 트리거하기 위해 주기적으로 `Poll` 호출API: `Poll`
- 방 입장/퇴장 알림 수신, 수신: `QAVEnterRoomComplete`

## 실시간 음성 채팅

1. 음성 채팅방 입장, API: `EnterRoom`
2. 마이크 활성화 또는 비활성화, API: `EnableMic`
3. 스피커 활성화 또는 비활성화, API: `EnableSpeaker`
4. 음성 채팅방 퇴장, API: `ExitRoom`

## 음성 메시지

- 인증 초기화, API: `ApplyPTTAuthbuffer`
- 스트리밍 음성 인식 시작, API: `StartRecordingWithStreamingRecognition`
- 녹화 중지, API: `StopRecording`
- GME 초기화 해제, API: `UnInit`

## 주요 API 액세스

### 1. SDK 다운로드

SDK 다운로드 가이드 페이지에서 적절한 [클라이언트 SDK](#)를 다운로드합니다.

### 2. 헤더 파일 가져오기

```
#include "tmg_sdk.h"

class UEDEMO1_API AUEDemoLevelScriptActor : public ALevelScriptActor,
public ITMGDelegate
{
public:
...
private:
...
}
```

### 3. 싱글톤 설정

모든 호출이 ITMGContext로 시작하고 콜백이 ITMGDelegate를 통해 애플리케이션에 전달되기 때문에 EnterRoom 함수를 호출하기 전에 먼저 ITMGContext를 가져와야 합니다.

## 예시 코드

```
ITMGContext* context = ITMGContextGetInstance();
context->SetTIMGDelegate(this);
```

## 4. SDK 초기화

- 이 API는 GME 서비스를 초기화하는 데 사용됩니다. 애플리케이션을 초기화할 때 호출하는 것이 좋습니다.
- OpenId는 App 개발자가 규정한 규칙으로 사용자를 고유하게 식별하며, App 내에서 중복될 수 없습니다(현재 INT64만 지원됨).
- 사용자가 로그인 계정을 전환하면 Uninit를 호출한 다음 새 OpenId로 Init GME를 다시 호출해야 합니다.

### 함수 프로토타입

```
//class ITMGContext
ITMGContext virtual int Init(const char* sdkAppId, const char* openId)
```

매개변수	유형	의미
sdkAppId	const char*	-
OpenId	const char*	OpenId는 Int64 유형만 가능(string으로 변환되어 전달됩니다).

## 예시 코드

```
std::string appid = TCHAR_TO_UTF8(CurrentWidget->editAppID-
>GetText().ToString().operator*());
std::string userId = TCHAR_TO_UTF8(CurrentWidget->editUserID-
>GetText().ToString().operator*());
ITMGContextGetInstance()->Init(appid.c_str(), userId.c_str());
```

## 5. 이벤트 콜백 트리거

update에서 Poll API를 주기적으로 호출하여 이벤트 콜백을 트리거할 수 있습니다. GME가 이벤트 콜백을 트리거 하려면 Poll API를 주기적으로 호출해야 합니다. 그렇지 않으면 전체 SDK 서비스가 예외적으로 실행됩니다. Demo에서 UEDemoLevelScriptActor.cpp 파일을 참고하십시오.

## 예시 코드

```
//헤더 파일의 선언
virtual void Tick(float DeltaSeconds);

void AUEDemoLevelScriptActor::Tick(float DeltaSeconds) {
    Super::Tick(DeltaSeconds);
    ITMGContextGetInstance() -> Poll();
}
```

## 6. 콜백 설정

API 유형은 Delegate 메소드를 사용하여 애플리케이션에 콜백 알림을 보냅니다. ITMG\_MAIN\_EVENT\_TYPE은 메시지 유형을 나타냅니다. Windows의 data는 json 문자열 형식입니다. key-value pairs에 대해서는 관련 문서를 참고하십시오.

### 예시 코드

```
//함수 구현:
//UEDemoLevelScriptActor.h:
class UEDEMO1_API AUEDemoLevelScriptActor : public ALevelScriptActor,
public SetTMGDelegate
{
public:
    void OnEvent (ITMG_MAIN_EVENT_TYPE eventType, const char* data);
}

//UEDemoLevelScriptActor.cpp:
void AUEDemoLevelScriptActor::OnEvent (ITMG_MAIN_EVENT_TYPE eventType,
const char* data){
    //eventType 판단 및 작업
}
```

## 7. 인증 정보

관련 기능의 암호화 및 인증을 위해 AuthBuffer를 생성합니다. 음성 메시지 및 음성 텍스트 변환에 대한 인증을 얻으려면 방 ID 매개변수를 null로 설정해야 합니다.

### 함수 프로토 타입

```
int QAVSDK_AuthBuffer_GenAuthBuffer (unsigned int dwSdkAppID, const
char* strRoomID, const char* strOpenID,
```

```
const char* strKey, unsigned char* strAuthBuffer, unsigned int
bufferLength);
```

매개변수	유형	의미
dwSdkAppID	int	Tencent Cloud 콘솔의 AppId
strRoomID	char*	방 ID, 최대 127자 지원
strOpenID	char*	사용자 ID. 초기화 중 openid와 동일.
strKey	char*	Tencent Cloud 콘솔의 권한 키
strAuthBuffer	char*	반환된 authbuff
bufferLength	int	전달된 authbuff 길이, 500 권장

## 예시 코드

```
unsigned int bufferLen = 512;
unsigned char retAuthBuff[512] = {0};
QAVSDK_AuthBuffer_GenAuthBuffer(atoi(SDKAPPID3RD), roomId, "10001",
AUTHKEY, retAuthBuff, bufferLen);
```

## 실시간 음성 채팅 액세스

### 1. 방 입장

생성된 인증 정보로 방에 입장할 때 사용하는 API입니다. 방 입장 후 마이크와 스피커는 기본적으로 활성화되어 있지 않습니다. 반환된 값 0은 API 호출이 성공했음을 의미하지만 방 입장이 성공했다는 의미는 아닙니다. 방 오디오 유형을 선택하는 방법에 대한 자세한 내용은 [Sound Quality](#)를 참고하십시오.

#### 함수 프로토 타입

```
ITMGContext virtual int EnterRoom(const char* roomID, ITMG_ROOM_TYPE
roomType, const char* authBuff, int buffLen)
```

매개변수	유형	의미
roomID	char*	방 ID, 최대 127자 지원
roomType	ITMG_ROOM_TYPE	방 오디오 유형

authBuffer	char*	인증 코드
buffLen	int	인증 코드 길이

## 예시 코드

```
ITMGContext* context = ITMGContextGetInstance();
context->EnterRoom(roomID, ITMG_ROOM_TYPE_FLUENCY,
(char*)retAuthBuff,bufferLen);
```

## 방 입장 콜백

사용자가 방에 입장하면 방 입장 알림이 수신되고 처리를 위해 수신 기능에서 식별됩니다. 성공적인 콜백(err = 0)은 방 입장 성공 및 과금 시작을 의미합니다. 당일 총 통화 시간이 700분 미만인 경우 요금이 부과되지 않습니다.

### 과금 참고

- [Purchase Guide](#)
- [Billing](#)
- [Will the billing continue if the client is disconnected from the server when using the voice chat?](#)

- 예시 코드 콜백 처리를 위한 예시 코드입니다.

```
void UBaseViewController::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,
const char *data) {

    FString jsonData = FString(UTF8_TO_TCHAR(data));
    TSharedPtr<FJsonObject> JsonObject;
    TSharedRef<TJsonReader<>> Reader =
        TJsonReaderFactory<>::Create(FString(UTF8_TO_TCHAR(data)));
    FJsonSerializer::Deserialize(Reader, JsonObject);

    if (eventType == ITMG_MAIN_EVENT_TYPE_ENTER_ROOM) {
        int32 result = JsonObject->GetIntegerField(TEXT("result"));
        FString error_info = JsonObject-
>GetStringField(TEXT("error_info"));
        if (result == 0) {
            GEngine->AddOnScreenDebugMessage(INDEX_NONE, 20.0f,
FColor::Yellow, TEXT("Enter room success."));
```

```

    }
    else{
        FString msg = FString::Printf(TEXT("Enter room failed. result=%d,
info = %ls"), result, *error_info);
        GEngine->AddOnScreenDebugMessage(INDEX_NONE, 20.0f,
FColor::Yellow, *msg);
    }
    onEnterRoomCompleted(result, error_info);
}
}

```

### • 에러 코드

에러 코드 값	원인 및 권장 방안
7006	인증 실패. 가능한 원인들: <ul style="list-style-type: none"> <li>AppID 가 존재하지 않거나 올바르지 않음</li> <li>authbuff를 인증하는 동안 오류 발생</li> <li>인증 만료</li> <li>openid 사양 불충족</li> </ul>
7007	이미 다른 방에 있음
1001	사용자는 이미 방에 들어가는 과정에 있었지만 이 작업을 반복했습니다. 방 입장 콜백이 반환될 때까지 방 입장 API를 호출하지 않는 것이 좋습니다.
1003	사용자가 이미 방에 있었고 API를 입력하는 방을 다시 호출함
1101	SDK가 초기화되었는지, openid가 규칙을 준수하는지, API가 동일한 스레드에서 호출되는지, Poll API가 정상적으로 호출되는지 확인

## 2. 마이크 활성화 또는 비활성화

이 API는 마이크를 활성화/비활성화하는 데 사용됩니다. 방 입장 후 마이크와 스피커는 기본적으로 활성화되어 있지 않습니다.

### 예시 코드

```

void UBaseViewController::OnEvent (ITMG_MAIN_EVENT_TYPE eventType, const
char *data) {

    FString jsonData = FString(UTF8_TO_TCHAR (data));
    TSharedPtr<FJsonObject> JsonObject;

```

```
TSharedRef<TJsonReader<>> Reader =
TJsonReaderFactory<>::Create(FString(UTF8_TO_TCHAR(data)));
FJsonSerializer::Deserialize(Reader, JsonObject);

if (eventType == ITMG_MAIN_EVENT_TYPE_ENTER_ROOM) {
    int32 result = JsonObject->GetIntegerField(TEXT("result"));
    FString error_info = JsonObject->GetStringField(TEXT("error_info"));
    if (result == 0) {
        GEngine->AddOnScreenDebugMessage(INDEX_NONE, 20.0f,
FColor::Yellow, TEXT("Enter room success."));
        //마이크 활성화
        ITMGContextGetInstance()->GetAudioCtrl()->EnableMic(true);
    }
    else{
        FString msg = FString::Printf(TEXT("Enter room failed. result=%d,
info = %ls"), result, *error_info);
        GEngine->AddOnScreenDebugMessage(INDEX_NONE, 20.0f,
FColor::Yellow, *msg);
    }
    onEnterRoomCompleted(result, error_info);
}
}
```

### 3. 스피커 활성화 또는 비활성화

이 API는 스피커를 활성화/비활성화하는 데 사용됩니다.

#### 예시 코드

```
void UBaseViewController::OnEvent (ITMG_MAIN_EVENT_TYPE eventType, const
char *data) {

    FString jsonData = FString(UTF8_TO_TCHAR(data));
    TSharedPtr<FJsonObject> JsonObject;
    TSharedRef<TJsonReader<>> Reader =
TJsonReaderFactory<>::Create(FString(UTF8_TO_TCHAR(data)));
    FJsonSerializer::Deserialize(Reader, JsonObject);
```

```
if (eventType == ITMG_MAIN_EVENT_TYPE_ENTER_ROOM) {  
    int32 result = JsonObject->GetIntegerField(TEXT("result"));  
    FString error_info = JsonObject->GetStringField(TEXT("error_info"));  
    if (result == 0) {  
        GEngine->AddOnScreenDebugMessage(INDEX_NONE, 20.0f,  
        FColor::Yellow, TEXT("Enter room success."));  
        //스피커 활성화  
        ITMGContextGetInstance()->GetAudioCtrl()->EnableSpeaker(true);  
    }  
    else {  
        FString msg = FString::Printf(TEXT("Enter room failed. result=%d,  
        info = %ls"), result, *error_info);  
        GEngine->AddOnScreenDebugMessage(INDEX_NONE, 20.0f,  
        FColor::Yellow, *msg);  
    }  
    onEnterRoomCompleted(result, error_info);  
}  
}  
}
```

## 4. 방 퇴장

이 API는 현재 방을 나가기 위해 호출됩니다. 종료를 위해 콜백을 기다리고 처리해야 합니다.

### 예시 코드

```
ITMGContext* context = ITMGContextGetInstance();  
context->ExitRoom();
```

### 방 퇴장 콜백

사용자가 방을 나가면 콜백이 반환됩니다. 예시 코드는 아래와 같습니다.

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char*  
data) {  
    switch (eventType) {  
        case ITMG_MAIN_EVENT_TYPE_EXIT_ROOM:  
        {  
            //프로세스
```

## 음성 메시지 액세스

## 1. 인증 초기화

SDK 초기화 후 인증 초기화를 호출합니다. authBuffer를 얻는 방법에 대한 자세한 내용은 `genAuthBuffer`(실시간 음성 채팅 인증 정보 API)를 참고하십시오.

## 함수 프로토타입

매개변수	유형	의미
authBuffer	char*	인증
authBufferLen	int	인증 길이

## 예시 코드

```
ITMGContextGetInstance() -> GetPTT() -  
> ApplyPTTAuthbuffer(authBuffer, authBufferLen);
```

## 2. 스트리밍 음성 인식 시작

이 API는 스트리밍 음성 인식을 시작하는 데 사용됩니다. 음성을 텍스트로 변환하여 얻은 텍스트는 콜백에서 실시간으로 반환됩니다. 인식할 언어를 지정하거나 음성으로 인식된 정보를 지정된 언어로 번역하고 번역을 반환할 수 있습니다. 녹음을 중지하려면 `StopRecording`을 호출합니다. 녹음이 중지된 후 콜백이 반환됩니다.

## 함수 프로토타입

```
ITMGPTT virtual int StartRecordingWithStreamingRecognition(const char*
filePath, const char* translateLanguage, const char* translateLanguage)
```

매개변수	유형	의미
filePath	char*	저장된 오디오 파일의 경로
speechLanguage	char*	오디오 파일이 텍스트로 변환될 언어이며, 매개변수는 <a href="#">Language Parameter Reference List</a> 참고
translateLanguage	char*	오디오 파일이 번역될 언어이며, 매개변수는 <a href="#">Language Parameter Reference List</a> 참고(현재 이 매개변수는 사용할 수 없으며, speechLanguage와 동일한 값을 입력하십시오)

## 예시 코드

```
ITMGContextGetInstance() ->GetPTT() -
>StartRecordingWithStreamingRecognition(filePath, "cmn-Hans-CN", "cmn-
Hans-CN");
```

## 스트리밍 음성 인식 콜백

스트리밍 음성 인식이 시작된 후 콜백 함수 OnEvent에서 콜백 메시지를 수신 대기해야 합니다. 이벤트 메시지는 `ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE` 입니다. 즉, 녹화가 중지되고 인식이 완료된 후 텍스트를 반환합니다. 이는 한 단락을 말한 이후에 인식된 텍스트를 반환하는 것과 같습니다. 이벤트 메시지는 실제 필요에 따라 OnEvent 함수에서 식별됩니다. 전달된 매개변수에는 다음 네 가지 메시지가 포함됩니다.

메시지 이름	의미
result	스트리밍 음성 인식 성공 여부를 판단하기 위한 반환 코드
text	음성에서 변환된 텍스트
file_path	저장된 녹음 파일의 로컬 경로
file_id	90일 동안 보관되는 녹음 파일의 백엔드 url 주소

### • 예시 코드

```

void UBaseViewController::OnEvent (ITMG_MAIN_EVENT_TYPE eventType,
const char *data) {

    FString jsonData = FString (UTF8_TO_TCHAR (data));
    TSharedPtr<FJsonObject> JsonObject;
    TSharedRef<TJsonReader>> Reader =
    TJsonReaderFactory<>::Create (FString (UTF8_TO_TCHAR (data)));
    FJsonSerializer::Deserialize (Reader, JsonObject);
    ...

    else if (eventType ==
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE)
    {

        int32 nResult = JsonObject->GetIntegerField (TEXT ("result"));
        FString text = JsonObject->GetStringField (TEXT ("text"));
        FString fileid = JsonObject->GetStringField (TEXT ("file_id"));
        FString file_path = JsonObject-
>GetStringField (TEXT ("file_path"));

        onPttStreamRecognitionCompleted (nResult, file_path, fileid,
text);
    }

    else if (eventType ==
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING)
    {

        int32 nResult = JsonObject->GetIntegerField (TEXT ("result"));
        FString text = JsonObject->GetStringField (TEXT ("text"));
        FString fileid = TEXT ("STREAMINGRECOGNITION_IS_RUNNING");
        FString file_path = JsonObject-
>GetStringField (TEXT ("file_path"));

        onPttStreamRecognitionisRunning (nResult, file_path, fileid,
text);
    }
}

```

## ● 예러 코드

에러 코드	의미	처리 방식
32775	스트리밍 음성을 텍스트로 변환하는 데 실패했지만 녹 음은 성공함	UploadRecordedFile API를 호출하여 녹음 파일을 업로 드한 다음 SpeechToText API를 호출하여 음성을 텍스트 로 변환

32777	스트리밍 음성을 텍스트로 변환하는 데 실패했지만 녹음 및 업로드는 성공함	반환된 메시지에는 업로드 성공 후 백엔드 url이 포함되어 있으며, SpeechToText API를 호출하여 음성을 텍스트로 변환
32786	스트리밍 음성-텍스트 변환 실패	스트리밍 녹화 중 스트리밍 녹화 API의 실행 결과가 반환될 때까지 기다리십시오

### 3. 녹음 중지

이 API는 녹음을 중지하는 데 사용됩니다. 비동기식이며 녹음이 중지된 후 녹음 완료에 대한 콜백이 반환됩니다. 녹음 파일은 녹음이 성공한 후에만 사용할 수 있습니다.

#### 함수 프로토타입

```
ITMGPTT virtual int StopRecording()
```

#### 예시 코드

```
ITMGContextGetInstance() -> GetPTT() -> StopRecording();
```

# Quick Integration of Sample Project

## Unreal Sample Project 빠른 실행

최종 업데이트 날짜: 2024-01-18 15:57:14

본문은 GME Unreal Sample Project를 빠르게 실행하고 예시 코드를 프로젝트에 통합하는 방법을 설명합니다.

### Unreal Sample Project 실행

#### 환경 요건

- Unreal Engine 4.22 이상.
- Microsoft Visual Studio.
- UnrealEngine 프로젝트를 실행할 수 있는 구성 환경.

#### 전제 조건

GME 음성 채팅 및 음성 메시지 서비스를 사전에 신청하고 AppId 및 Key를 받아야 합니다. GME 서비스 신청 방법은 [서비스 활성화](#)를 참고하십시오. appId는 콘솔 AppID에 해당하고, authKey는 콘솔의 권한 키에 해당합니다.

#### 작업 단계

##### 1단계: 프로젝트 다운로드

[SDK 다운로드 가이드](#)의 안내에 따라 Unreal Sample Project를 다운로드합니다. UE5와 UE4의 Demo 구성이 다르기 때문에 해당 엔진 버전에 대한 Sample Project를 다운로드해야 합니다.

OS/Engine	Update Time	SDK Download	Sample Project Download	Documents
Unity	January 18, 2023	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Quick Integration of SDK for Unity</a>
Unreal Engine 4.x	January 18, 2023	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Quick Integration of SDK for Unreal Engine</a>
Unreal Engine 5.x	January 18, 2023	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Quick Integration of SDK for Unreal Engine</a>
Cocos2D	January 18, 2023	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Getting Started</a>
Windows	January 18, 2023	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Native SDK Quick Access</a>
iOS	January 18, 2023	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Quick Integration of Native SDK</a>
Android	January 18, 2023	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Quick Integration of Native SDK</a>
macOS	January 18, 2023	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Quick Integration of Native SDK</a>
Web	2022-06-20	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">API Documentation</a>

## 2단계: 프로젝트 구성

다운로드 후 프로젝트 디렉터리를 열고 Source\UEDemo1 경로에서 UserConfig.cpp를 찾아 붉은색 상자의 appID와 appKey를 적용된 GME 콘솔 [서비스관리-애플리케이션 설정]에서 AppID와 권한 키로 변경합니다.

```
// ...
// ...
// ...
https://console.cloud.tencent.com/gamegme

std::string UserConfig::GetAppID() {
    FString appID;
    GConfig->GetString(*UserInfoSection, TEXT("AppID"), appID, GGameIni);
    if (appID.IsEmpty())
    {
        appID = "xxxxxxxx";
    }
    return TCHAR_TO_UTF8(*appID);
}

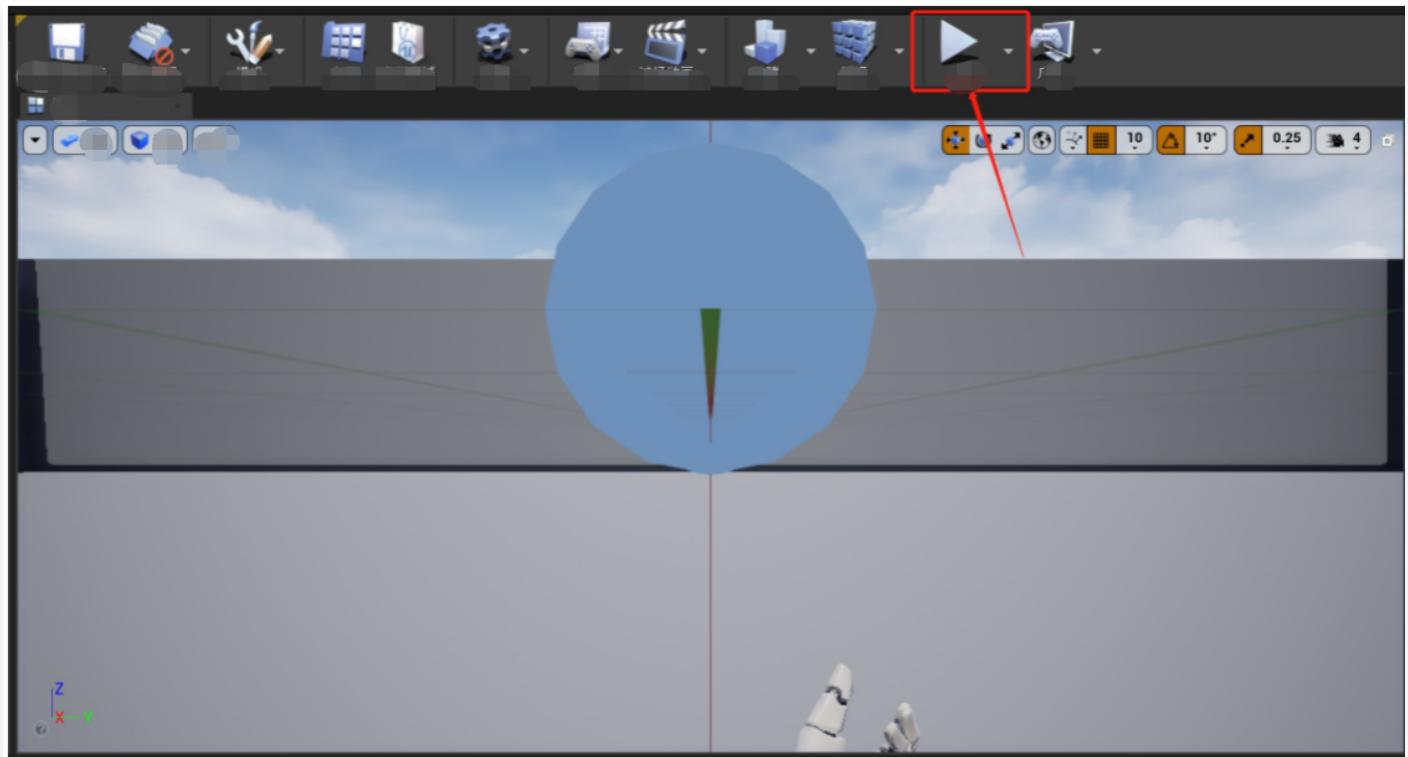
void UserConfig::SetAppKey(const std::string& appKey) {
    GConfig->SetString(*UserInfoSection, TEXT("AppKey"), UTF8_TO_TCHAR(appKey.c_str()), GGameIni);
    GConfig->Flush(false, GGameIni);
}

std::string UserConfig::GetAppKey() {
    FString appKey;
    GConfig->GetString(*UserInfoSection, TEXT("AppKey"), appKey, GGameIni);
    if (appKey.IsEmpty())
    {
        appKey = "xxxxxxxx";
    }
    return TCHAR_TO_UTF8(*appKey);
}
```

### 3단계: 컴파일 및 실행

#### 1) 프로그램 실행

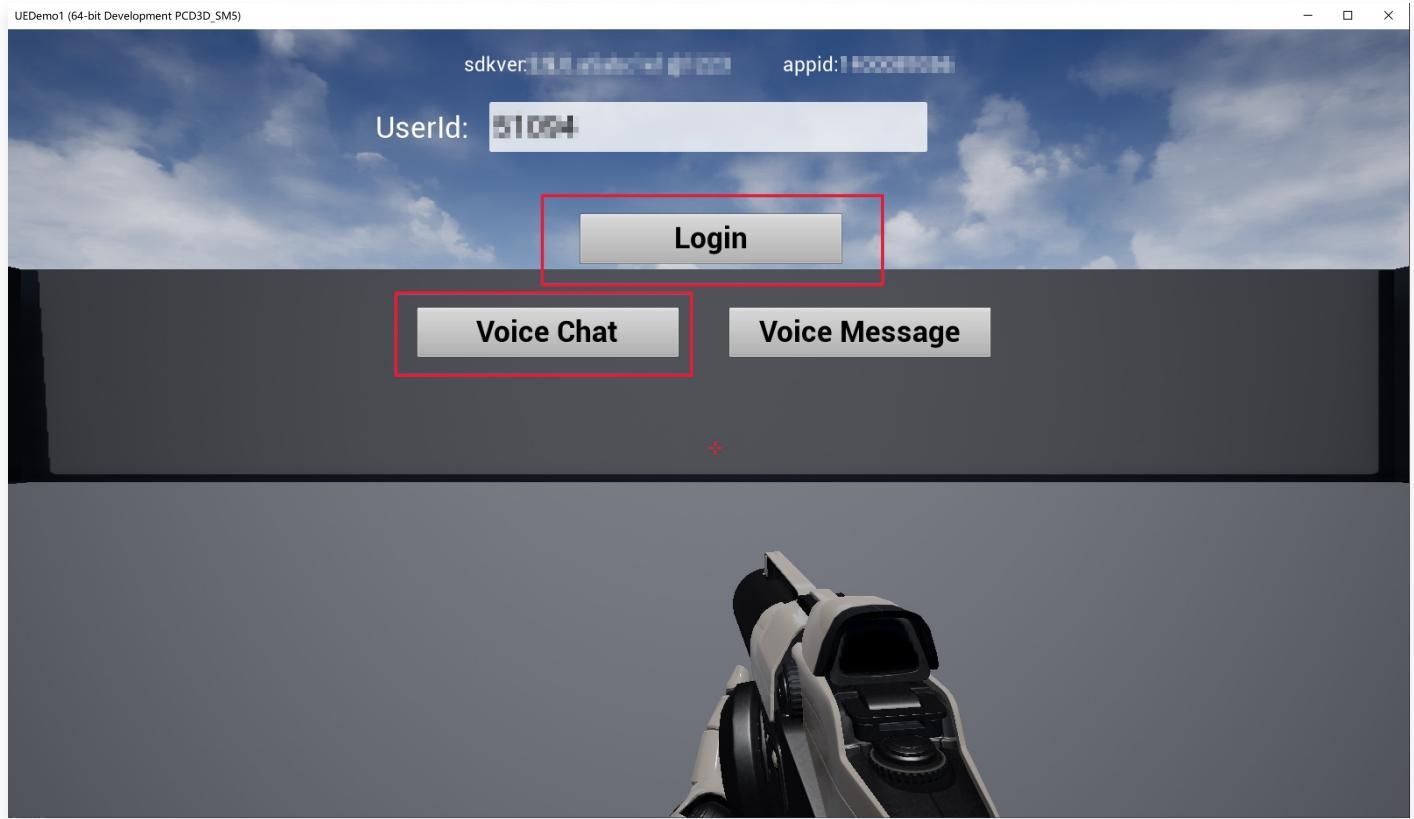
편집기에서 을(를) 클릭하여 프로그램을 실행합니다.



## 2) 초기화

- UserID: openid와 동일. openid는 애플리케이션에서 사용자의 고유 식별자이며 각 터미널의 openid는 고유해야 합니다.
- Voice Chat: 음성 채팅 기능 UI입니다.
- Voice Message: 음성 메시지 기능 UI입니다.

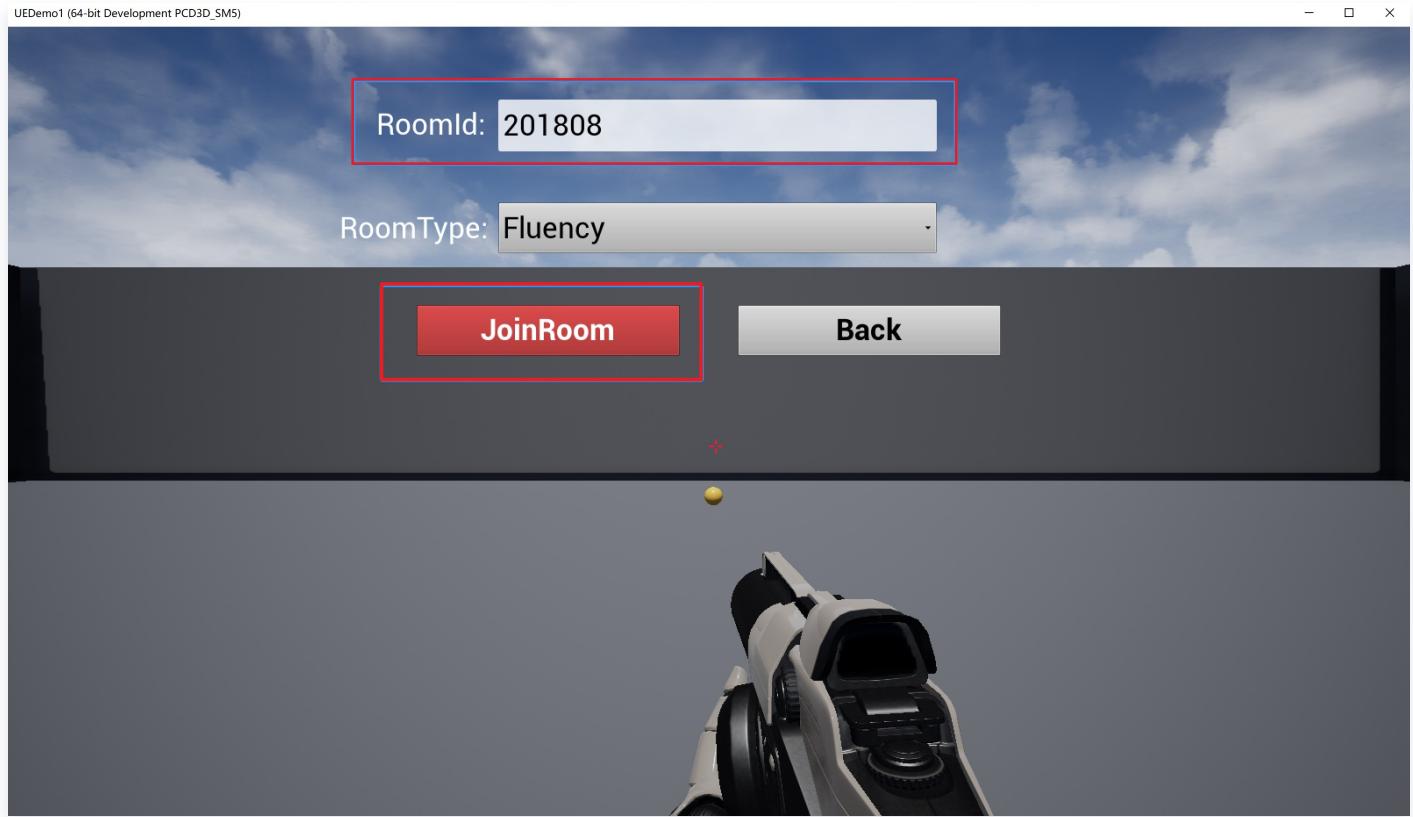
[Login](#)을 클릭하여 초기화한 후 **Voice Chat** 을 클릭하여 실시간 음성 채팅방 설정 페이지로 이동합니다.



### 3) 음성 채팅방 입장

- RoomId: 방 ID입니다. 같은 방에 있는 사용자는 음성으로 서로 통신할 수 있습니다.
- RoomType: Fluency를 사용하여 방에 입장합니다.
- JoinRoom: 음성 채팅 방에 입장합니다.
- Back: 이전 페이지로 돌아갑니다.

음성 채팅방 ID를 설정한 후, **JoinRoom**를 클릭하여 방에 입장합니다.



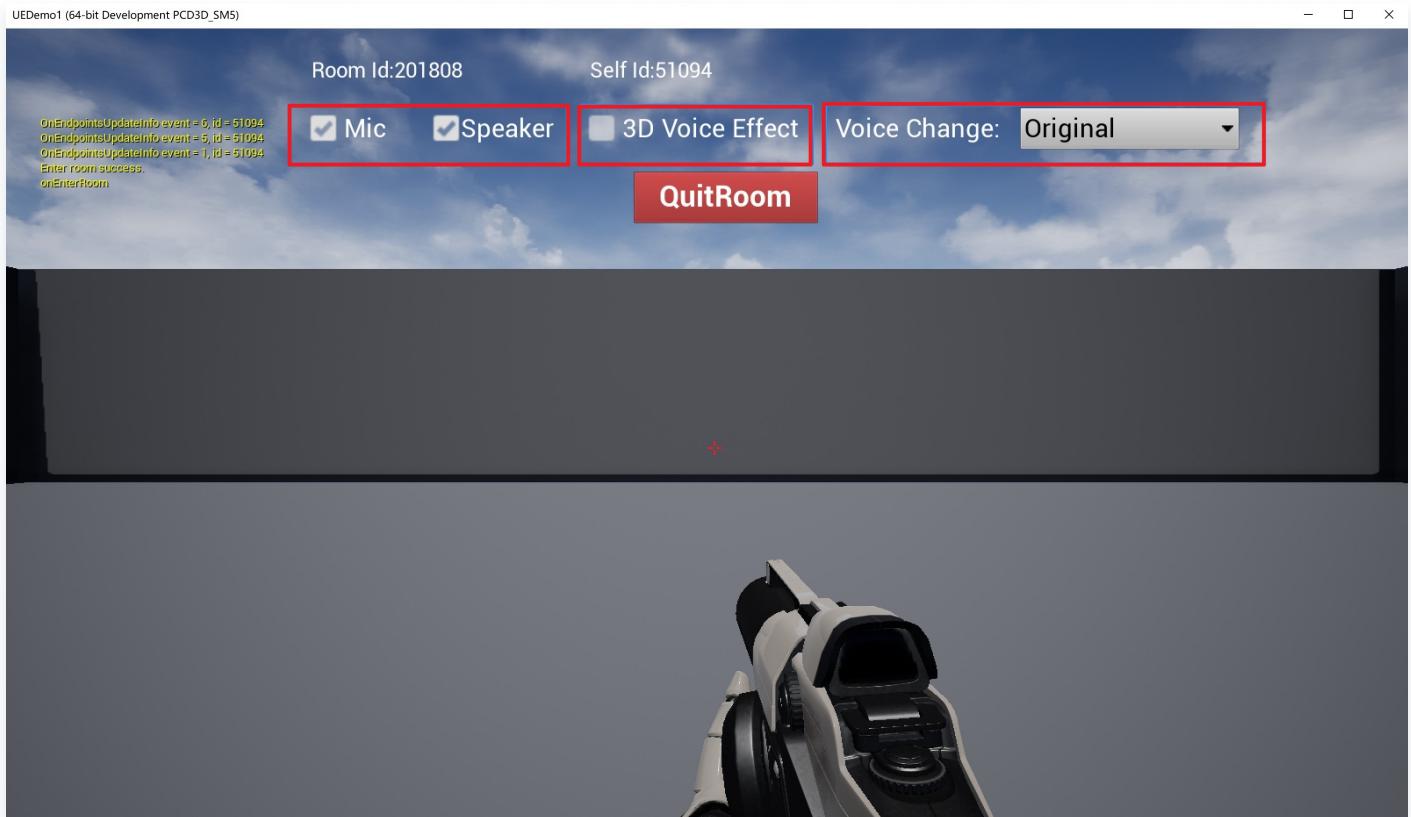
#### 4) 실시간 음성 채팅 사용

페이지에 방 입장을 위한 RoomID와 로컬 openID가 표시됩니다.

- Mic: 마이크를 켜려면 선택합니다.
- Speaker: 스피커를 켜려면 선택합니다.
- 3D Voice Effect: 3D 사운드 효과를 활성화하려면 선택합니다.
- Voice Change: 음성 변조 효과를 선택합니다.

로컬에서 Mic와 Speaker를 선택한 후, 다른 기기에서 위의 과정을 반복하여 같은 방에 들어가 Mic와 Speaker를 선택하면 양측의 통신이 가능합니다.

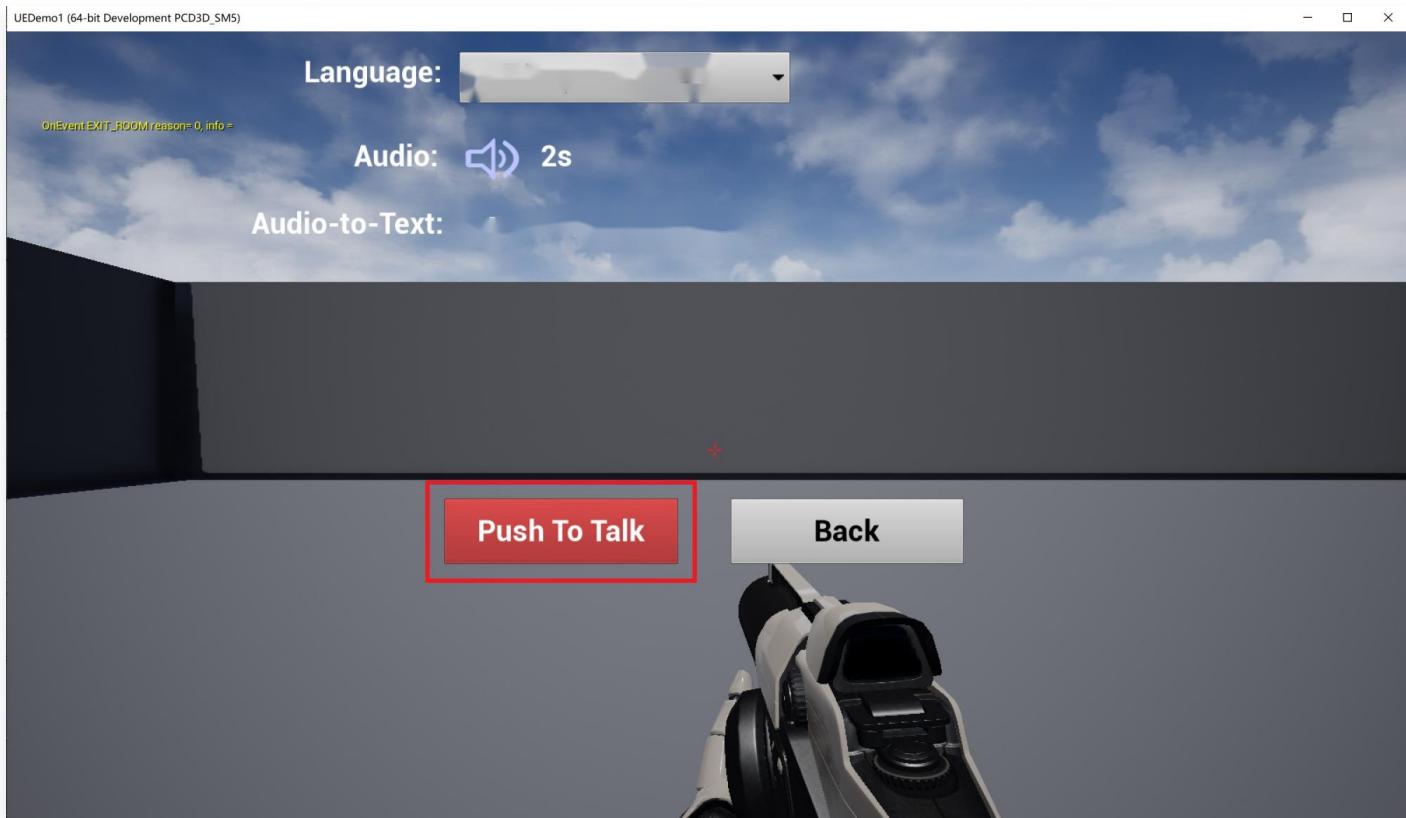
두 터미널 모두 3D Voice Effect를 선택한 경우, 키보드 [A] [S] [D] [W] 키를 사용하여 방향을 변경하여 음성의 3D 스테레오 효과를 경험할 수 있습니다.



## 5) 음성 메시지 사용

- Language: 텍스트 변환 대상 언어를 선택합니다. 예를 들어 중국어를 사용하는 경우 중국어를 선택합니다.
- Audio: 녹음 후 클릭하면 들을 수 있습니다.
- Audio-to-Text: 텍스트로 변환된 음성 메시지 내용입니다.
- Push To Talk: 길게 누르면 녹음됩니다.
- Back: 이전 페이지로 돌아갑니다.

Push to Talk를 길게 누르고 마이크에 대고 말합니다. 버튼을 놓으면 음성 메시지가 텍스트로 변환되어 UI에 표시됩니다.



## Sample Project 코드 소개

GME 실시간 음성 채팅을 사용하는 주요 프로세스는 Init > EnterRoom > EnableMic > EnableSpeaker입니다. Sample Project의 주요 코드는 `BaseViewController.cpp` 및 `ExperientialDemoViewController.cpp`에 있습니다.

### 초기화

초기화 코드는 `BaseViewController.cpp` 파일의 `InitGME` 함수에 있습니다. 초기화, 음성 메시지 인증 초기화, TMGDelegate 콜백 설정이 포함됩니다.

```
int UBaseViewController::InitGME(std::string sdkAppId, std::string
sdkAppKey, std::string userId) {

    int nAppid = atoi(sdkAppId.c_str());
```

```
int ret = ITMGContextGetInstance()->Init(sdkAppId.c_str(),
userId.c_str());

ITMGContextGetInstance()->SetTMGDelegate(this);

int RetCode = (int) ITMGContextGetInstance()->CheckMicPermission();
FString msg = FString::Printf(TEXT("check Permission retcode =%d"),
RetCode);
GEngine->AddOnScreenDebugMessage(INDEX_NONE, 10.0f, FColor::Yellow,
*msg);

char strSig[128] = {0};
unsigned int nLength = 128;
nLength = QAVSDK_AuthBuffer_GenAuthBuffer(nAppid, "0",
userId.c_str(), sdkAppKey.c_str(), (unsigned char *)strSig, nLength);
ITMGContextGetInstance()->GetPTT()->ApplyPTTAuthbuffer(strSig,
nLength);

m_appId = sdkAppId;
m_appKey = sdkAppKey;
m_userId = userId;
m_isEnableTips = false;
m_tipsMark = 0;
return ret;
}
```

GME를 사용하려면 UEDemoLevelScriptActor.cpp 스크립트의 Tick에 있는 Poll 함수를 주기적으로 호출해야 합니다.

```
void AUEDemoLevelScriptActor::Tick(float DeltaSeconds) {
Super::Tick(DeltaSeconds);

m_pTestDemoViewController->UpdateTips();
m_pCurrentViewController->UpdatePosition();
ITMGContextGetInstance()->Poll();
}
```

## 방 입장

방 입장 코드는 BaseViewController.cpp 파일의 EnterRoom 함수에 있습니다.

```
void UBaseViewController::EnterRoom(std::string roomID, ITMG_ROOM_TYPE roomType) {
    int nAppid = atoi(m_appId.c_str());
    UserConfig::SetRoomID(roomID);

    char strSig[128] = {0};
    unsigned int nLength = 128;
    nLength = QAVSDK_AuthBuffer_GenAuthBuffer(nAppid, roomID.c_str(),
    m_userId.c_str(), m_appKey.c_str(), (unsigned char *)strSig, nLength);
    GEngine->AddOnScreenDebugMessage(INDEX_NONE, 10.0f, FColor::Yellow,
    TEXT("onEnterRoom")));
    ITMGContextGetInstance()->EnterRoom(roomID.c_str(), roomType,
    strSig, nLength);
}
```

방 입장 콜백은 동일한 스크립트의 OnEvent 함수에 있습니다.

```
if (eventType == ITMG_MAIN_EVENT_TYPE_ENTER_ROOM) {
    int32 result = JsonObject->GetIntegerField(TEXT("result"));
    FString error_info = JsonObject-
    >GetStringField(TEXT("error_info"));
    if (result == 0) {
        GEngine->AddOnScreenDebugMessage(INDEX_NONE, 20.0f,
        FColor::Yellow, TEXT("Enter room success."));
    }
    else{
        FString msg = FString::Printf(TEXT("Enter room failed.
result=%d, info = %ls"), result, *error_info);
        GEngine->AddOnScreenDebugMessage(INDEX_NONE, 20.0f,
        FColor::Yellow, *msg);
    }
    onEnterRoomCompleted(result, error_info);
}
```

## 장치 활성화

방 입장 성공 후 장치 활성화 코드는 ExperientialDemoViewController.cpp에 있습니다.

```
void UEExperientialDemoViewController::onCheckMic(bool isChecked) {
```

```
//GEngine->AddOnScreenDebugMessage(INDEX_NONE, 10.0f,  
FColor::Yellow, L"onCheckMic");  
ITMGContext *pContext = ITMGContextGetInstance();  
if (pContext) {  
    ITMGAudioCtrl *pTmgCtrl = pContext->GetAudioCtrl();  
    if (pTmgCtrl) {  
        pTmgCtrl->EnableMic(isChecked);  
    }  
}  
}  
  
void UEExperientialDemoViewController::onCheckSpeaker(bool isChecked) {  
    //GEngine->AddOnScreenDebugMessage(INDEX_NONE, 10.0f,  
FColor::Yellow, L"onCheckSpeaker");  
    ITMGContext *pContext = ITMGContextGetInstance();  
    if (pContext) {  
        ITMGAudioCtrl *pTmgCtrl = pContext->GetAudioCtrl();  
        if (pTmgCtrl) {  
            pTmgCtrl->EnableSpeaker(isChecked);  
        }  
    }  
}
```

## 3D 음향 효과

3D 음향 효과 연결에 대해서는 [3D 음향 효과](#)를 참고하십시오. 프로젝트에서는 먼저 ExperientialDemoViewController.cpp의 코드를 사용하여 3D 음향 효과 기능을 초기화합니다.

```
void UEExperientialDemoViewController::onCheckSpatializer(bool isChecked)  
{  
    char buffer[256]={0};  
    //    sprintf(buffer, sizeof(buffer), "%s3d_model",  
    getFilePath().c_str());  
    sprintf(buffer, sizeof(buffer), "%sgme_2.8_3d_model.dat",  
    getFilePath().c_str());  
    int ret1 = ITMGContextGetInstance()->GetAudioCtrl()-  
    >InitSpatializer(buffer);  
    int ret2 = ITMGContextGetInstance()->GetAudioCtrl()-  
    >EnableSpatializer(isChecked, false);
```

```
FString msg = FString::Printf(TEXT("InitSpatializer=%d, EnableSpatializer ret=%d"), ret1, ret2);
GEngine->AddOnScreenDebugMessage(INDEX_NONE, 10.0f, FColor::Yellow, msg);
}
```

UEDemoLevelScriptActor.cpp 스크립트의 Tick에서 UpdatePosition 함수를 호출합니다.

```
void AUEDemoLevelScriptActor::Tick(float DeltaSeconds) {
    Super::Tick(DeltaSeconds);

    m_pTestDemoViewController->UpdateTips();
    m_pCurrentViewController->UpdatePosition();
    ITMGContextGetInstance()->Poll();
}

void UBaseViewController::UpdatePosition() {
    if (!m_isCreated)
        return;

    ITMGRoom *pTmgRoom = ITMGContextGetInstance()->GetRoom();
    if (!pTmgRoom)
    {
        return;
    }

    int nRange = GetRange();
    pTmgRoom->UpdateAudioRecvRange(nRange);

    FVector cameraLocation =
        UGameplayStatics::GetPlayerCameraManager(m_pActor->GetWorld(), 0)-
    >GetCameraLocation();
    FRotator cameraRotation =
        UGameplayStatics::GetPlayerCameraManager(m_pActor->GetWorld(), 0)-
    >GetCameraRotation();

    FString msg = FString::Printf(TEXT("location(x=%.2f, y=%.2f, z=%.2f), rotation(pitch=%.2f, yaw=%.2f, roll=%.2f)"),
        nRange, cameraLocation.X, cameraLocation.Y, cameraLocation.Z, cameraRotation.Pitch, cameraRotation.Yaw, cameraRotation.Roll));
}
```

```
    cameraLocation.X, cameraLocation.Y, cameraLocation.Z,
    cameraRotation.Pitch, cameraRotation.Yaw, cameraRotation.Roll);

    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 };

    pTmgRoom->UpdateSelfPosition(position, forward, right, up);
    SetPositionInfo(msg);
}
```

ExperientialDemoViewController.cpp에서 3D 효과를 활성화합니다.

```
void UExperientialDemoViewController::onCheckSpatializer(bool isChecked)
{
    char buffer[256] = {0};

    // snprintf(buffer, sizeof(buffer), "%s3d_model",
    // filePath().c_str());
    snprintf(buffer, sizeof(buffer), "%s3d_model.dat",
    filePath().c_str());

    int ret1 = ITMGContextGetInstance()->GetAudioCtrl()-
    >InitSpatializer(buffer);

    int ret2 = ITMGContextGetInstance()->GetAudioCtrl()-
    >EnableSpatializer(isChecked, false);

    FString msg = FString::Printf(TEXT("InitSpatializer=%d,
    EnableSpatializer ret=%d"), ret1, ret2);

    GEngine->AddOnScreenDebugMessage(INDEX_NONE, 10.0f, FColor::Yellow,
    msg);
}
```

