

GiGA Genie Inside Device SDK Developer Guide

1. 개요

1.1 문서 연혁

변경일	변경사항
2019.07.01	- 최초
2019.07.16	- Linux SDK 예제 추가
2019.08.01	- agent_setLocation 추가 - dialog_response 수정
2019.08.14	- Window SDK 예제 추가 - kws_setModelPath 추가
2019.09.24	- metaInfo 의 TTS 메시지를 title -> msg 로 변경함 - 컨텐츠 상세 내용을 전달하는 metaInfo에 대한 설명 추가

1.2 목적

본 문서는 (주)케이티(이하 KT)의 GiGA Genie AI 서비스를 3rd Party Application에서 이용할 수 있도록 제공하는 GiGA Genie Inside SDK를 사용하기 위한 연동 방법에 대해 기술한다.

1.3 적용범위

- 3rd Party Application에서 Inside SDK를 연동하는 방법을 설명한다.
- 3rd Party Application의 빌드 버전에 따라 일부 기능에 제약이 있을 수 있으며, 경우에 따라 별도 빌드된 Inside SDK를 제공 가능하다.
- 내용은 KT서비스 제공 계획의 변경 및 추가에 따라 추후 보완 및 변경될 수 있다.

1.4 지원 버전

Device SDK는 아래 버전으로 빌드 되었다.

- Android: Android 4.4 Version 부터 지원
 - minSDK Version = 19

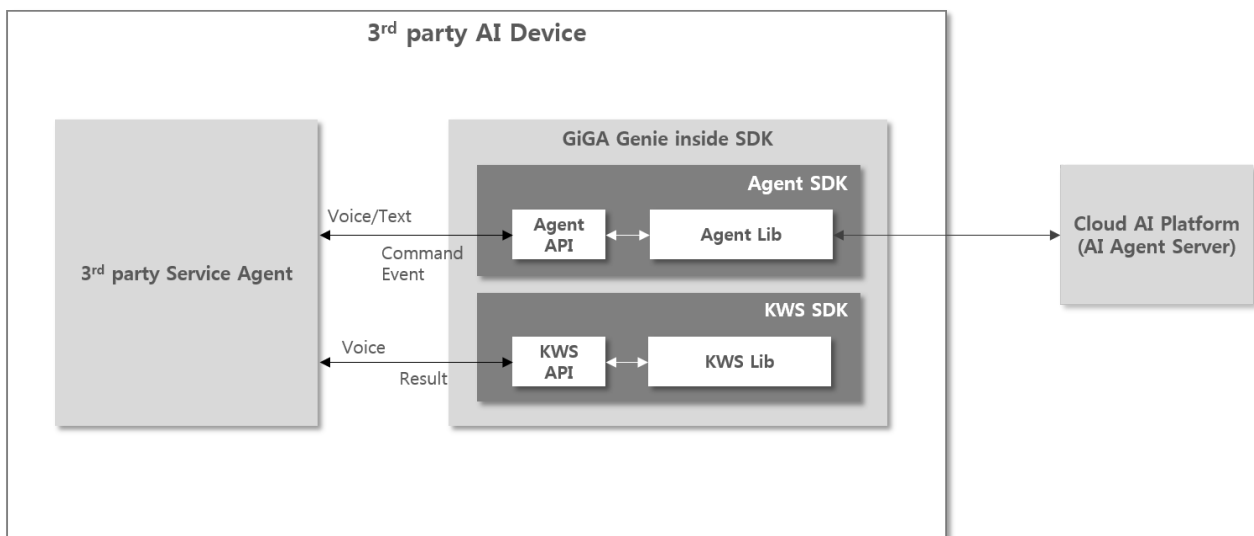
- targetSDK Version = 26
 - compileSdkVersion = 26
- Linux: Ubuntu 16.04 지원 (x86_64)
 - OS = Ubuntu 16.04 x86_64
 - Compiler = gcc 5.4.0
- Windows: Windows 7 부터 지원 (64bit)
 - OS = Windows 10 64bit
 - Compiler = Visual Studio 2017 15.9.12

1.5 약어 정리

- KWS: Keyword Spotting의 약어로 호출어 인식을 의미
- TTS: Text to Speech의 약어로 음성출력을 의미
- gRPC: gRPC Remote Procedure Call의 약어로 원격 함수를 호출할 수 있는 프로토콜

2. S/W Architecture

GiGA Genie Inside SDK(이하 G-SDK) 구성 및 3rd Party Service Agent와의 연동구조에 대해 설명한다.



2.1 G-SDK 구성

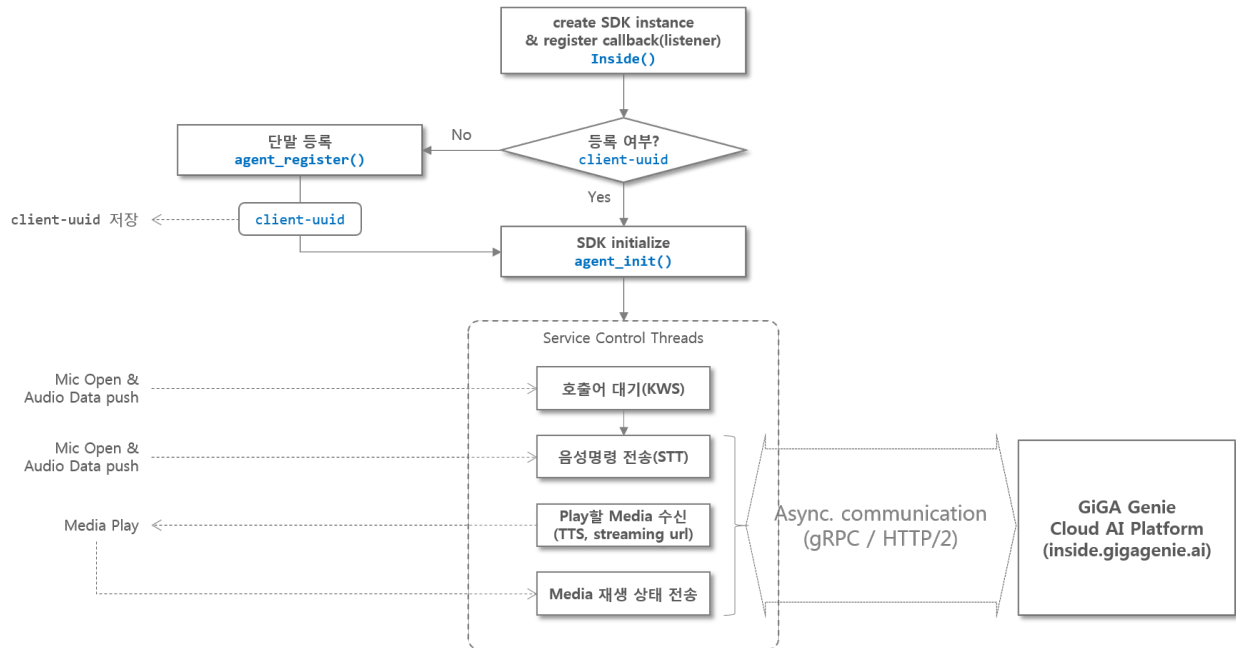
G-SDK는 Agent SDK와 KWS SDK로 구성되어 있다. 각 SDK의 역할과 API 구성에 대해 설명한다.

- Agent SDK
 - AI Agent 서버 연동(HTTPS, gRPC, WebSocket) 및 gRPC Connection 관리
 - 음성(PCM)/텍스트 데이터를 AI Agent 서버로 전송하고 그에 대한 응답을 3rd Party Service Agent로 전달
- KWS SDK
 - 3rd Party Service Agent로부터 음성 데이터(PCM)를 전달받고 설정된 호출어 ID에 따른 호출어 인식

여부를 판단

- 3rd Party Service Agent의 제어에 따라 호출어 인식 시작/종료 처리

2.2 G-SDK Flow Overview



2.2 연동 구조

2.2.1. Service Agent <-> Agent SDK

- Service Agent는 Agent SDK에서 제공하는 API를 통해 음성/텍스트 등을 AI Agent 서버로 전송할 수 있다.
- Service Agent는 callback 형태로 AI Agent 서버로부터 수신된 Response/Event/TTS 등을 Agent SDK로 부터 전달 받을 수 있다.

2.2.2. Agent SDK <-> AI Agent Server

- HTTP POST 기반 Register/Init/Unregister: Service Agent로 부터 전달받은 client-id, client-key, client-secret, userid 정보를 AI Agent 서버로 전달하여 uuid 정보를 발급 받거나 AI Agent 서버에 등록된 UUID 정보를 삭제할 수 있다.
- gRPC/WebSocket 프로토콜을 통해 음성/텍스트/TTS/이벤트 등을 송수신 한다.

2.2.3. Service Agent <-> KWS SDK

- KWS SDK를 사용하기 위해서는 Agent SDK를 반드시 init해야 한다.
- KWS SDK에서 제공하는 API를 통해 호출어 인식 시작/인식/중지 및 호출어 Id 변경을 할 수 있다.
- 호출어 인식 여부는 detection API의 리턴 값으로 판단할 수 있다.
- 호출어 인식을 위해 16KHz Signed 16bit Linear PCM 데이터가 KWS SDK로 전달 되어야 한다.

3. 디바이스 등록

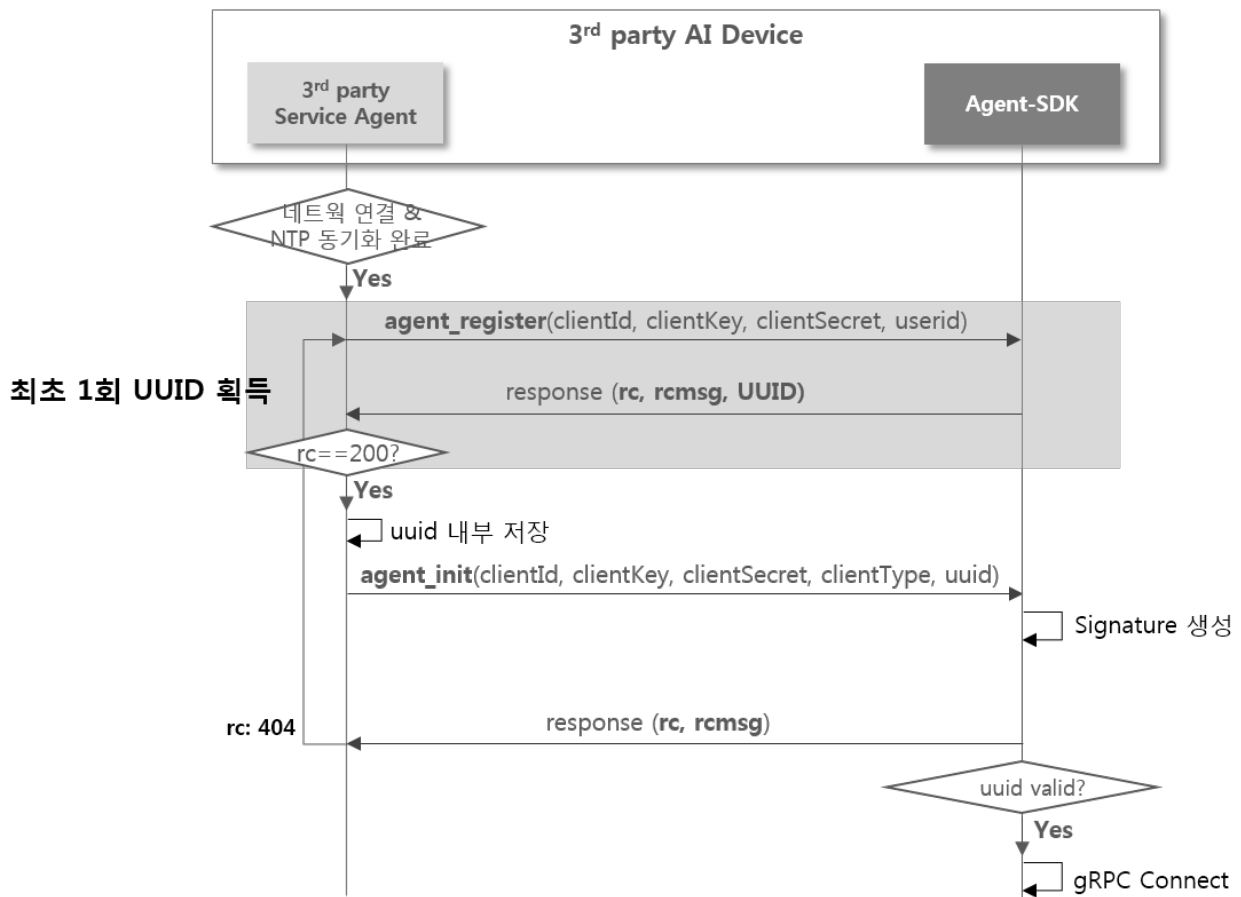
G-SDK를 이용하는 3rd Party Application은 API를 사용하기 위해 Client 정보 입력을 통해 디바이스를 등록 해야 한다. 디바이스 등록을 위해 필요한 값과 관련 절차는 아래에 설명한다.

3.1. 3rd Party Client 정보 발급

G-SDK 사용을 위해서는 SDK register 시 인증에 필요한 Client 정보를 입력해야 한다. Client 정보는 KT API Link (<https://apilink.kt.co.kr/>)에서 발급 받을 수 있다.

- client_type: 클라이언트 타입으로 inside SDK를 사용하는 경우 "GINSIDE"
- client-id: 클라이언트 아이디로, 단말당 1개의 client_id가 제공
- client-key: 클라이언트 키로, 단말당 개발/상용 client_key가 제공
- client-secret: 클라이언트 시크릿으로, 단말당 개발/상용 client_key가 제공

3.2. G-SDK Register

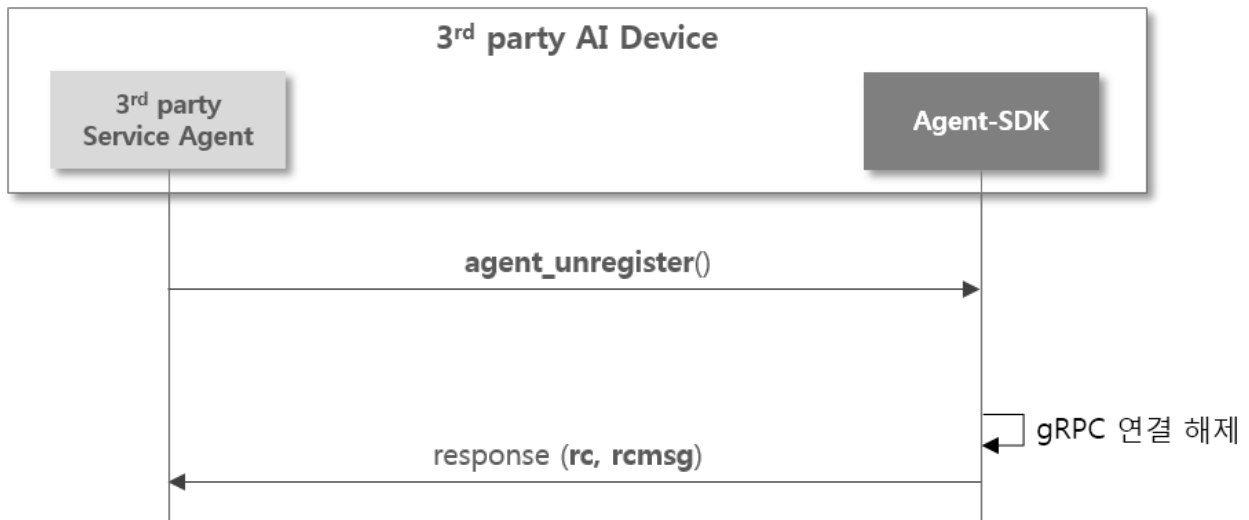


- userid: 사용자 별 Unique한 값을 입력해야 하며, 3rd Party Application에서 입력하지 않으면 Agent SDK에서 단말의 MAC 주소를 추출하여 입력한다.
- uuid: 서버로 부터 발급된 Unique한 Key 값이며, Agent SDK init시에 uuid를 함께 전달하여 등록된 단말임을 검증 받는다.

- Register 절차

- 1) 불필요한 uuid 중복 발급을 피하기 위해 Service Agent는 기 발급받은 uuid 정보를 client_id, client_key, client_secret, userid 정보와 함께 내부적으로 저장/관리 해야한다.
- 2) Service Agent는 디바이스 네트워크 연결 및 NTP 동기화가 완료된 이후에 기존에 발급된 uuid 가 없는 경우 register API를 통해 uuid를 발급요청 한다.
- 3) 기존에 발급받은 uuid 정보가 있거나 register API를 통해 uuid를 발급 받은 경우, init API를 통해 Agent-SDK 초기화를 요청한다.
- 4) Agent SDK는 전달받은 uuid 정보의 유효성 여부를 체크한 후 서버와 gRPC 연결을 시도한다.

3.3. G-SDK Unregister



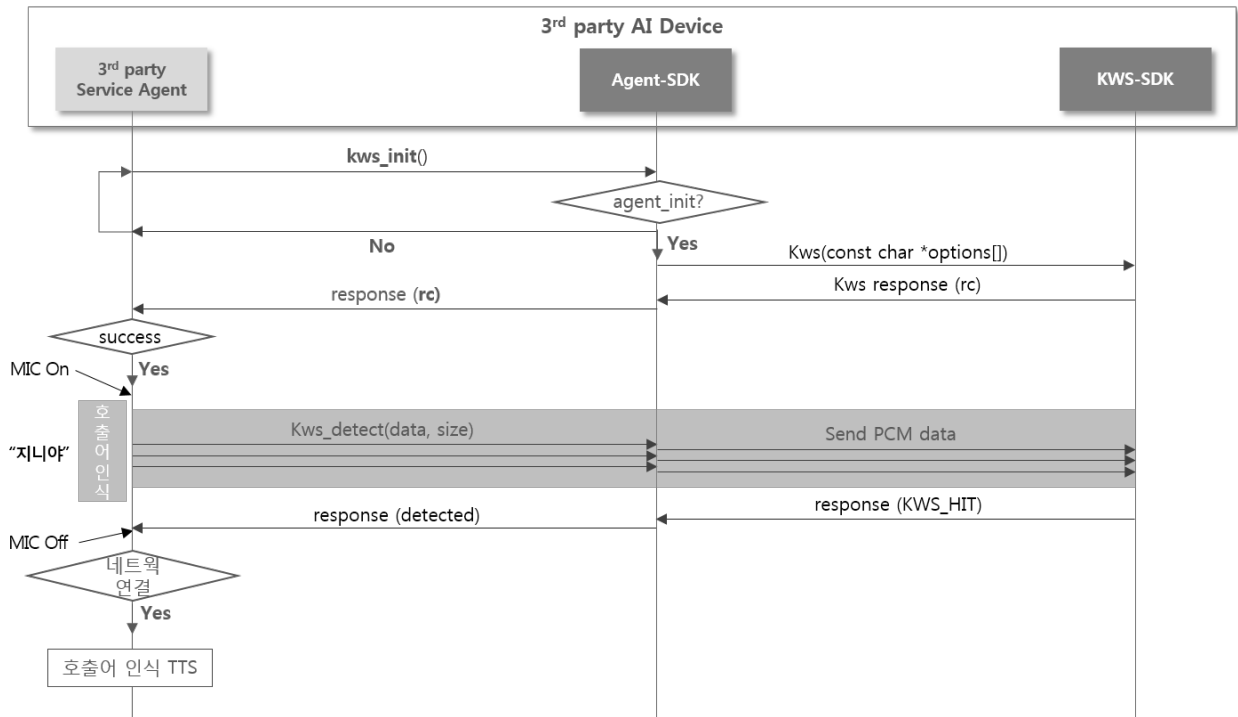
- Unregister 절차

- 1) 디바이스가 초기화 되거나 사용자 등록 정보가 초기화 되는 경우 Service Agent는 unregister API를 통해 서버에 등록된 정보(uuid)를 삭제해야 한다.
- 2) uuid 정보가 삭제된 경우 Agent-SDK는 gRPC 연결을 해제하고 Service Agent로 response를 전달한다.
- 3) Service Agent는 response를 수신하는 경우 필요하다면 디바이스를 무인증 상태로 전환하고 그에 해당하는 처리를 진행한다.

4. 음성인식 처리

4.1. 호출어 인식

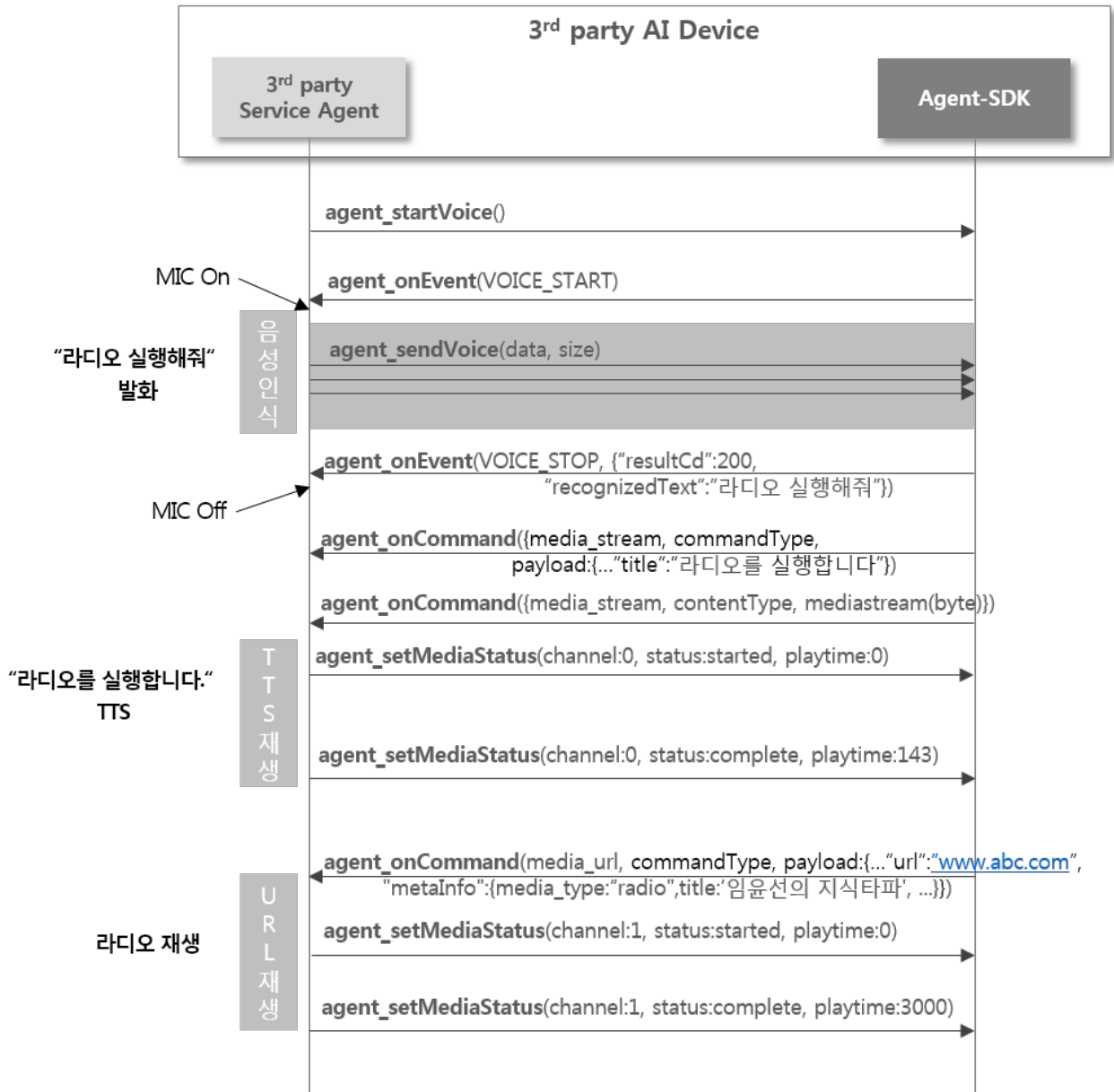
호출어 인식 절차는 아래와 같다.



- 1) Service Agent는 호출어 인식을 위해 KWS SDK 초기화를 요청한다. 만약 Agent SDK가 초기화 되어 있지 않다면 Agent SDK 초기화 후 KWS SDK 초기화를 요청해야 한다.
- 2) KWS SDK 초기화가 완료된 경우 호출어 인식을 위해 마이크를 open 하고 마이크에서 입력된 데이터를 정의한 포맷에 맞게 전달한다.
- 3) 호출어 인식 시 Service Agent는 기존에 동작 중이던 미디어 정보를 알맞게 처리해야 한다. 에러 케이스인 경우 TTS 재생을 시작하고 호출어 인식 모드로 다시 동작해야 한다.
- 4) 호출어가 인식된 경우 Service Agent는 음성 데이터 전송을 중지하고 마이크를 close하고 음성인식을 시작한다.
- 5) Service Agent에서의 호출어 인식 시작/종료 시점은 다음과 같다.
 - 디바이스 부팅 완료 후 호출어 인식 시작
 - 호출어가 인식된 경우 또는 디바이스 마이크 버튼 입력 시 호출어 인식 종료
 - 음성인식 완료 이벤트 수신 후 TTS 또는 미디어 재생 시작 시 호출어 인식 시작
 - 네트워크가 연결되지 않거나 정상적으로 register가 안된 상태에서는 에러 TTS 재생 시작 후 호출어 인식 시작

4.2 음성명령 인식

음성명령 인식 절차는 아래와 같다.



- 1) 호출어 인식이나 디바이스의 마이크 버튼에 의해 음성인식이 시작되어야 하는 경우 Service Agent는 agent_startVoice API를 통해 서버로 음성인식을 요청한다.
- 2) 음성인식 시작 메시지(START_VOICE)를 수신하면 Service Agent는 마이크 open 후 VOICE_STOP 이벤트 수신 전까지 음성 데이터를 Agent SDK로 전달한다.
- 3) 음성인식이 완료되면 서버는 음성인식 중단 메시지(VOICE_STOP)를 Service Agent로 전달한다.
- 4) VOICE_STOP 이벤트를 수신한 Service Agent는 음성 데이터 전송을 중단하고 메시지를 기다린다.
- 5) Service Agent는 TTS/미디어 재생 요청, 볼륨 제어 요청 등이 수신되면 요청에 대한 처리를 시작하고 호출어 인식 모드로 동작한다.

5. 미디어 재생

음성발화에 대한 응답으로 TTS(wav 또는 PCM 데이터)나 음악 재생 등을 위한 URL이 전달될 수 있고 Service Agent는 각 케이스에 따라 미디어를 재생해야 한다.

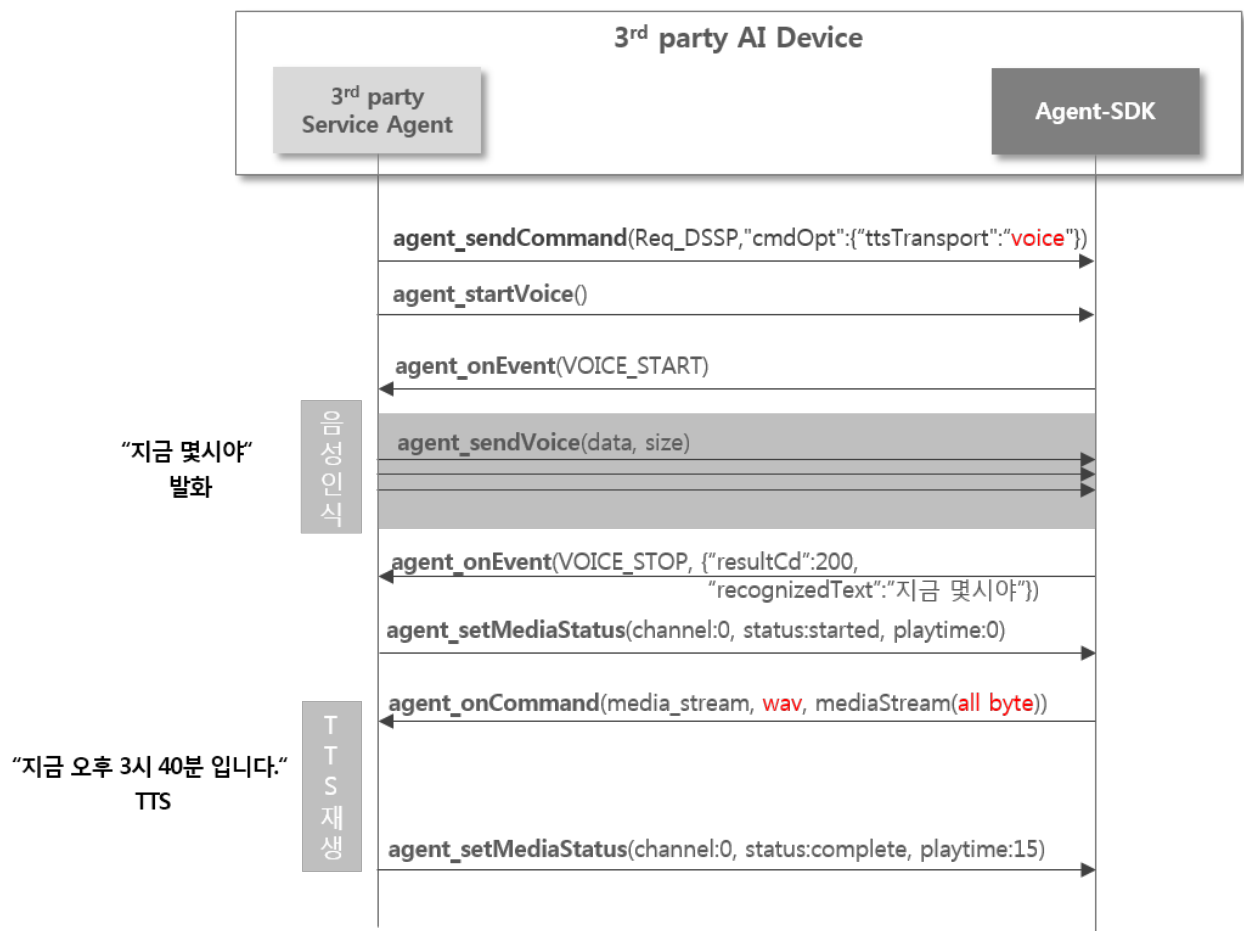
5.1. 재생 타입

5.1.1. URL

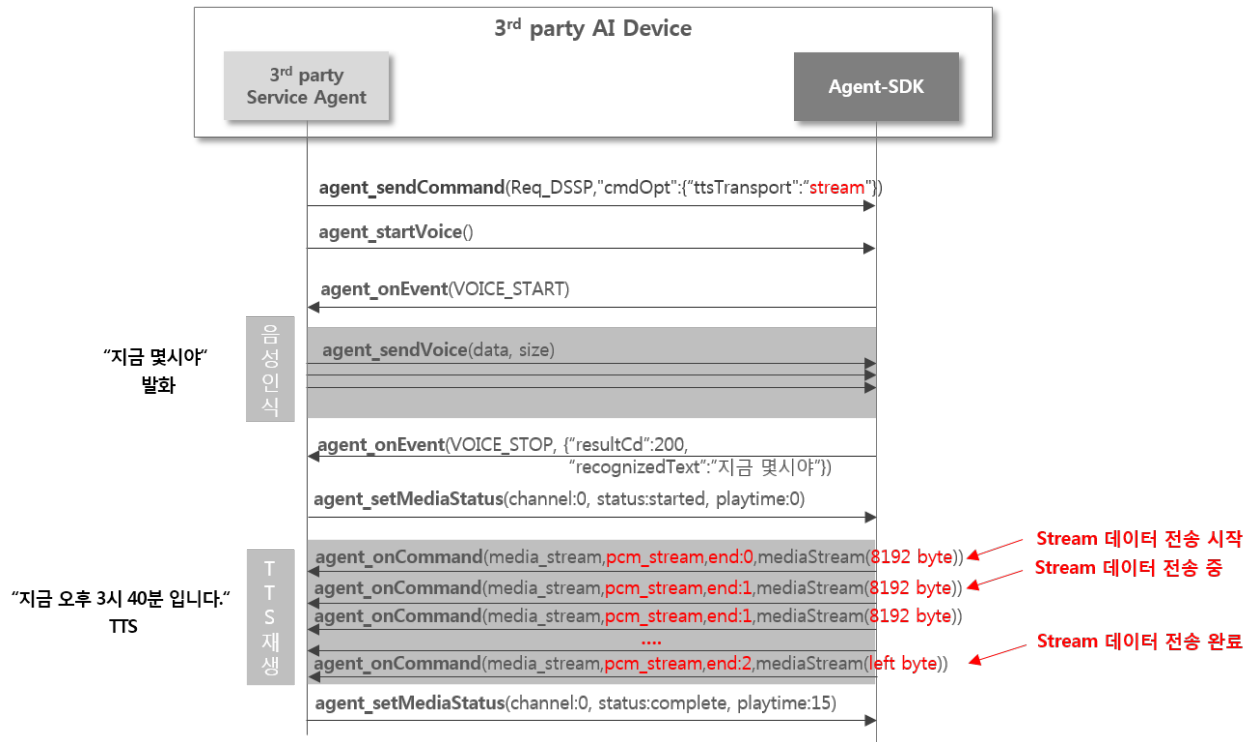
- 서버는 재생할 미디어 URL 정보를 전달하고 Service Agent는 전달받은 URL을 재생해야 한다.
- 'cmdOpt'에 'playNotiTime'(초 단위) 정보가 설정되어 전달되는 경우 playNotiTime 만큼 재생이 되면 'noti' 이벤트를 Agent-SDK로 전달해야 한다.

5.1.2. WAV / PCM data

- Wav 파일 전송 Flow



- Stream 파일 전송 Flow



- TTS의 경우 wav 파일을 한번에 전송 받거나(wav header 44bytes 포함), wav 파일의 헤더정보를 제외한 데이터를 일정한 사이즈로 분할하여 stream 형태로 전송 받을 수 있다.
- TTS 수신 방식을 변경하고자 할 때 Service Agent는 TTS 전송방식을 Agent-SDK로 전송해야 한다. 별도 TTS 수신 방식을 지정하지 않는 경우 wav 파일 전체가 전송된다.
- stream 형태로 TTS가 전달되는 경우 Service Agent에서는 Mono Channel, Signed 16-bit Little Endian 샘플 형식, 16KHz 샘플 주기의 PCM 데이터를 재생할 수 있어야 한다.
- 분할된 TTS 데이터는 'end' 정보(0:시작, 1:중간, 2:끝)를 통해 전송되는 데이터의 시작과 끝을 구별할 수 있다.

5.2. Stream ID

channel 정보로 미디어 재생을 제어할 수 있다.

- 재생되는 미디어를 구분하기 위해 stream ID가 사용되며 미디어 재생 요청 시 재생하고자 하는 미디어의 stream ID가 Req_PLMD 메시지를 통해 전달된다.
- TTS의 경우 0번을 사용하고 그 외 미디어는 1~10 사이의 값을 사용한다.
- TTS를 제외한 미디어는 동시 재생이 가능해야 한다.
 - 1번 재생 중 1번에 대한 제어명령없이 2번 재생요청이 전달되면 1번과 2번 미디어를 동시 재생

5.3 재생 상태 이벤트

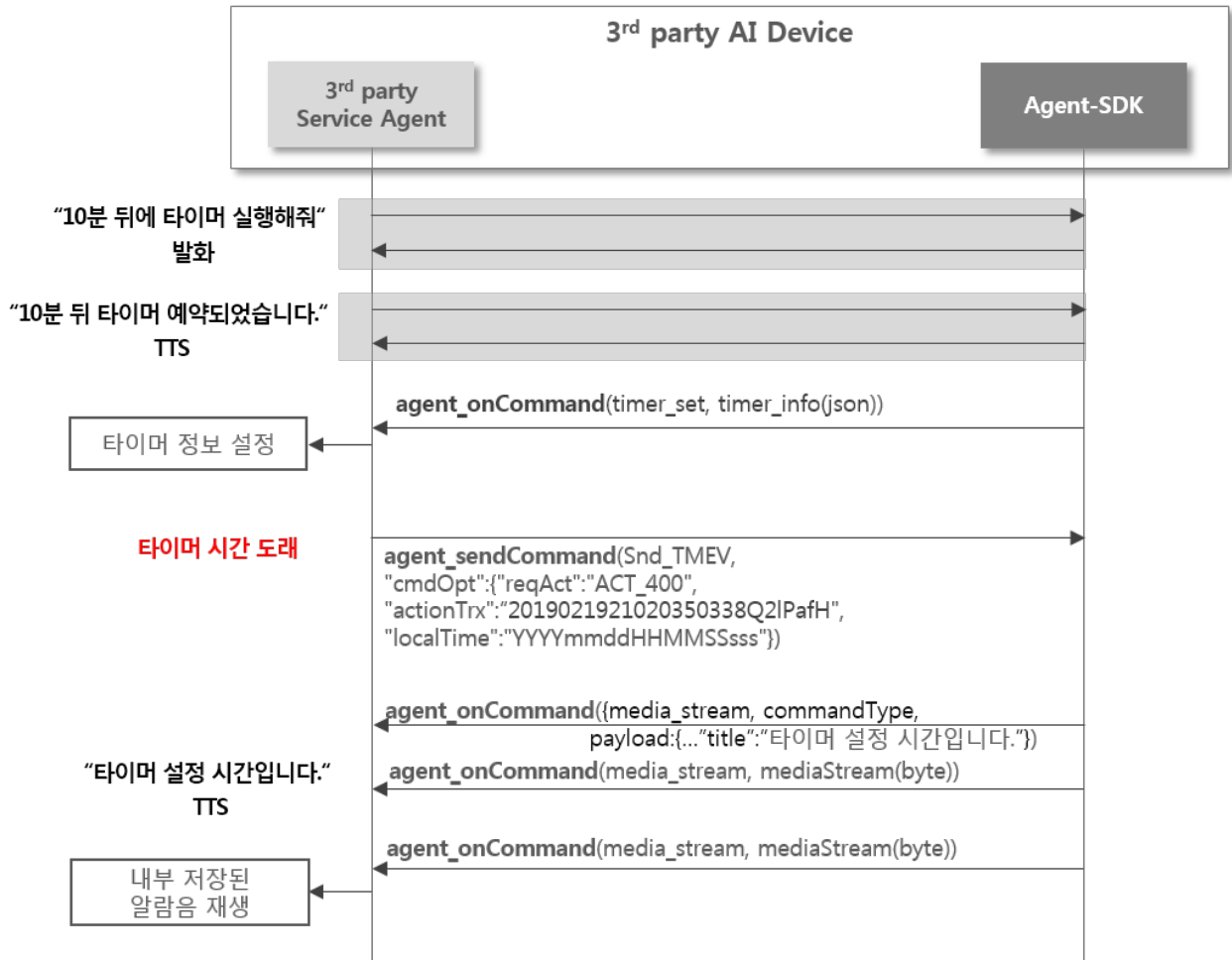
- Service Agent는 Agent-SDK로부터 미디어 재생요청을 받아 재생하는 경우 재생상태에 따라 "started", "paused", "stopped", "complete" 이벤트를 stream ID, 총 재생시간 정보와 함께 미디어 상태 이벤트를 전송해야 한다.
- 미디어 재생상태 이벤트가 전달되지 않는 경우 이후 서비스 진행에 문제가 발생할 수 있다.

5.4 지니뮤직 서비스



- 음악 또는 라디오 등 재생을 요청한 경우 해당 서비스 인증 여부에 따라 처리되는 시나리오가 다르다.
- 서비스 인증이 필요한 경우 인증을 위한 웹 페이지 URL이 전달되고 Service Agent는 해당 웹 페이지를 사용자에게 표시하여 인증이 가능하도록 해야 한다.
- 서비스 인증이 완료된 경우 서비스 시작을 알리는 TTS가 전달되면 Service Agent는 TTS 재생을 완료한 후 재생완료 이벤트를 Agent-SDK로 전달해야 한다.

6. 알람/타이머 처리



6.1. 알람

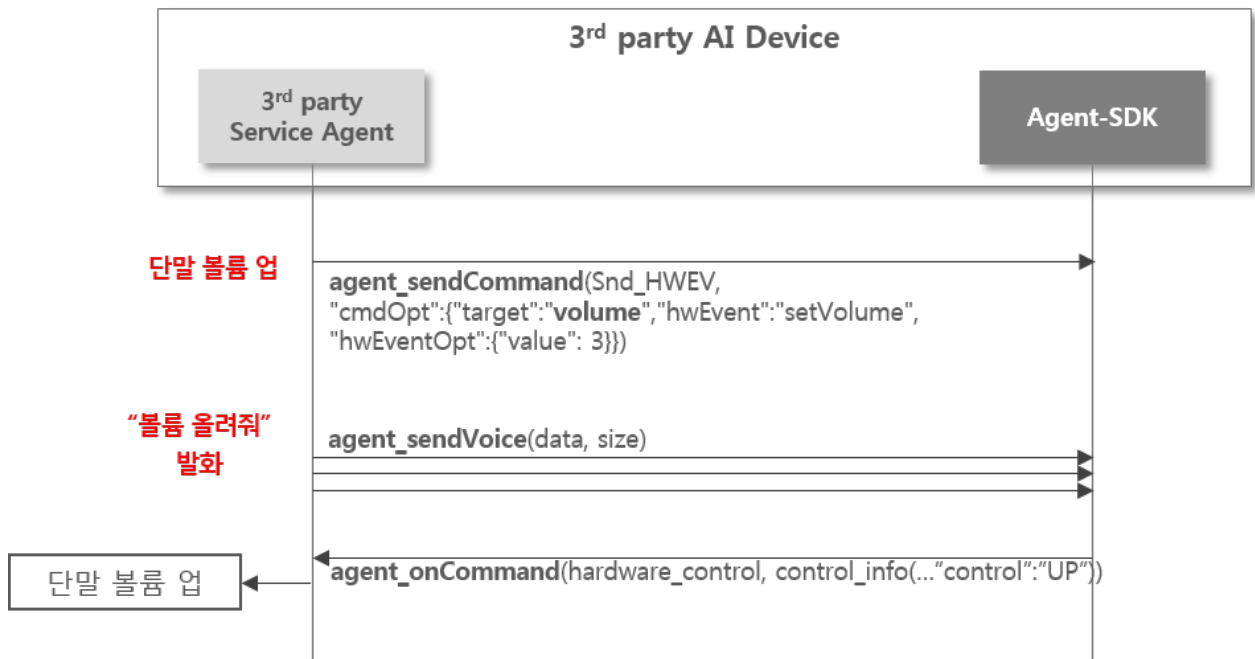
- Service Agent는 알람 설정 메시지를 수신하면 'setOpt'에 따라 reqAct, actionTrx, setTime 정보를 저장/관리해야 한다.
- 설정된 시간정보는 YYYYMMDDHHMMSS 형태로 전달되며 설정된 시간이 되면 reqAct, actionTrx, localTime(디바이스 시간)정보를 포함한 알람이벤트를 Agent-SDK로 전달해야 한다.
- 설정된 알람음 타입에 따라 음원 URL이 전달되거나 디바이스 내장 알람음 ID에 대한 정보가 Req_PLMD 메시지를 통해 전달되고, Service Agent는 전달된 'url' 정보를 기준으로 URL을 재생하거나 내장 알람음을 재생해야 한다.
- 알람음 반복 및 종료는 디바이스 자체 시나리오에 따라 처리한다.

6.2. 타이머

- 타이머의 경우 설정 시간(setTime)이 절대시간이 아닌 초단위의 상대시간으로 전달된다.
- Service Agent는 현재 디바이스 시간을 기준으로 전달된 setTime만큼 계산된 시간정보를 저장/관리해야 한다.
- 설정된 시간이 경과한 후 reqAct, actionTrx, localTime(디바이스 시간) 정보를 포함한 타이머 이벤트를 Agent-SDK로 전달해야 한다.
- 재생할 타이머 효과음 ID정보가 Req_PLMD 메시지의 'url' 파라미터에 설정되어 전달되고 Service Agent는 전달된 ID에 해당하는 타이머 효과음을 재생한다.

7. 단말 하드웨어 이벤트 처리

7.1 볼륨



사용자가 물리적 버튼을 눌렀거나 음성 발화를 통해 볼륨을 조절 하는 경우 아래와 같이 처리한다.

- 디바이스 볼륨이 변경되면 AI Agent 서버로 변경된 볼륨 값을 전송한다.
- 사용자가 음성발화를 통해 볼륨을 조절하면 서버는 `agent_onCommand`로 볼륨에 대한 이벤트를 전달한다.

7.2 버튼



- 미디어 제어(play/pause/forward/backward)에 관련된 버튼 입력 시 HW 이벤트를 Agent SDK로 전달하면, 서버에서 현재 재생 중인 미디어에 따라 제어 명령을 전달 받거나 이전곡/다음곡에 해당하는 음원 URL을 전달 받을 수 있다.

7.3 블루투스

```

sequenceDiagram
    participant AI_Device as 3rd party AI Device
    participant Agent_SDK as Agent-SDK
    participant Bluetooth as Bluetooth 기기

    AI_Device->>Bluetooth: 블루투스 기기 연결
    Bluetooth->>AI_Device: 노래 재생 요청
    AI_Device->>Agent_SDK: agent_sendCommand(Snd_HWEV,"cmdOpt":{"target":"bluetooth", "hwEvent":"BTEvent","hwEventOpt":{"event": 801}})
    Agent_SDK->>AI_Device: agent_startVoice()
    Agent_SDK->>AI_Device: agent_onEvent(VOICE_START)
    AI_Device->>Agent_SDK: agent_sendVoice(data, size)
    Agent_SDK->>AI_Device: agent_onCommand(hardware_control, command_type, control_info(...,"hwCmdOpt":{"control":"forward"}))
    AI_Device->>Bluetooth: 블루투스 기기 연결 해제
    AI_Device->>Agent_SDK: agent_sendCommand(Snd_HWEV,"cmdOpt":{"target":"bluetooth", "hwEvent":"BTEvent","hwEventOpt":{"event": 800}})
  
```

The diagram illustrates the interaction between a 3rd party AI Device, its Agent-SDK, and a Bluetooth device for music playback control. The process begins with the AI Device connecting to the Bluetooth device. Upon receiving a music playback request from the Bluetooth device, the AI Device sends a command to the Agent-SDK to start voice playback. The Agent-SDK then sends an event back to the AI Device. The AI Device subsequently sends voice data to the Agent-SDK. The Agent-SDK then sends a command back to the AI Device, which finally disconnects from the Bluetooth device and sends a command to the Agent-SDK to end the session.

	API	설명
Inside	Inside (Context context, InsideListener listener)	Inside 라이브러리 생성 및 리스너 등록 (Android)
String	agent_register (String client_id,String client_key,String client_secret,String user_id)	서버에 Authorize 요청 (UUID 생성 요청)
String	agent_init (String client_id, String client_key, String client_secret, String uuid)	입력받은 UUID로 서버에 확인 후 정상적으로 발급된 UUID 면 gRPC 연결
String	agent_unregister ()	UUID 삭제 요청
void	agent_setCommandEventCallback (agent_onCommand aoc, agent_onEvent aoe)	Agent SDK 콜백 함수 등록 (Linux, Window)
void	agent_setServerInfo (String serverIP, String grpcPort, String restPort)	연동서버 정보를 설정
void	agent_sendText (String msg)	텍스트 명령 요청
void	agent_startVoice ()	음성인식 시작 요청
void	agent_sendVoice (short[] data, int size)	음성인식을 위한 음성 stream 전송
void	agent_stopVoice ()	음성인식 취소
void	agent_getTTS (String msg)	입력된 텍스트의 TTS 스트림 전달
String	agent_serviceLogin (String service_type, String return_url)	OAuth 인증을 통해 서비스 로그인을 수행할 수 있는 페이지정보를 요청
void	agent_sendCommand (String cmd)	서버로 전달할 요청/이벤트
void	agent_onCommand (String cmd)	서버로부터 전달받은 음성 인식 해석 결과 콜백
void	agent_onEvent (int evt, String opt)	서버로 부터 전달받은 음성인식 시작,종료,서버 에러 이벤트
void	agent_reset ()	클라이언트 초기화
void	agent_debugmode (bool enable)	디버그 모드 (로그 보기)
void	agent_updateMediaStatus (int ch, String state, int playtime)	재생중인 미디어 상태 전송
String	agent_getVersion ()	Agent SDK 버전 정보 조회
void	agent_setLocation (String Longitude, String Latitude, String Address)	단말 위치 정보 입력

8.2 KWS SDK 목록

	API	설명
int	kws_init()	호출어 라이브러리 객체 생성
int	kws_detect (short[] data, int size)	호출어 인식을 위한 음성 데이터 stream 전송
void	kws_reset()	호출어 라이브러리 초기화
int	kws_error()	호출어 라이브러리 초기화나 연산 과정 에러 리턴
String	kws_getVersion()	호출어 라이브러리 버전 조회
int	kws_getKeyword()	현재 사용 중인 호출어 ID 조회
int	kws_setKeyword (int kwsId)	호출어 변경
int	kws_setModelPath (String path)	호출어 모델 경로 설정

9. Agent API 상세

9.1 Agent SDK 라이브러리 생성

3rd Party Application이 서버와 연동하기 위해서 객체를 생성한다. Agent SDK 를 사용하기 전에 반드시 호출되어야 한다. 해당 API는 **Android**에서만 제공된다.

- API
Inside(Context context, InsideListener listener)
- Parameter

이름	유형	설명
context	Context	-
listener	InsideListener	3rd Party Application이 Agent SDK 로부터 Notification이나 이벤트 등을 수신하기 위한 callback

3rd Party Application은 Notification이나 이벤트 등을 수신하기 위한 callback을 아래와 같이 등록한다.

- **agent_onCommand**(string cmd): [9.11 agent_onCommand 참고](#)
 - 음성인식 해석 결과 전달
 - 전달받은 actionType 종류에 따라 아래와 같은 시나리오로 처리하면 된다.
 - start_voice: 단말 Mic On 후 음성인식
 - dialog_response: 대화서버 응답을 파싱하여 서비스 앱의 Flow에 맞게 처리
 - media_stream: 전달받은 음성 stream을 재생
 - media_url: 전달받은 url을 통해 미디어 재생
 - hardware_control: 전달받은 하드웨어 제어 정보를 실행
 - timer_set: 전달받은 타이머 정보를 설정/해제

- `webview_url`: 전달받은 url을 웹뷰에서 실행
- **agent_onEvent**(int evt, string opt): [\[첨부 2\] onEvent 참고](#)
 - 음성인식 시작, 음성인식 종료, 서버 에러 이벤트 전달
 - 전달받은 이벤트를 수신하여 UI 및 음성인식 재시작 혹은 음성 명령어 모드를 동작 시킨다.
- 사용 예시
 - Android

```
import gigagenie.kt.com.inside.api.Inside;
public class MainActivity extends AppCompatActivity implements
    InsideListener;

InsideSDK insideSDK = new Inside(getApplicationContext(), this);
```

9.2 Agent SDK UUID 생성 요청

GiGA Genie 서비스를 사용하기 위해서 유니크한 uuid를 발급받는다. 발급받은 uuid는 3rd Party Application에 내부적으로 저장하고 있다가 `agent_init` 시에 사용한다. 만약 내부적으로 저장하고 있던 uuid가 삭제된 경우에는 해당 과정을 다시 거쳐야 한다.

- API
 - agent_register**(string client_id, string client_key, string client_secret, string user_id)
- Parameter

이름	유형	설명
client_id	string	기가지니 개발자 포털을 통해 발급 받은 클라이언트 아이디
client_key	string	기가지니 개발자 포털을 통해 발급 받은 클라이언트 키
client_secret	string	기가지니 개발자 포털을 통해 발급 받은 클라이언트 시크릿
user_id	string	사용자 식별을 위한 정보로, 만약 이 정보가 없는 경우 단말 MAC 주소 사용

- Result

이름	유형	설명
rc	int	결과 코드
rcmsg	String	결과 메시지
uuid	String	사용자별 Unique한 UUID

- rc는 다음과 같이 정의 된다.
 - 200: 성공

- 400: 파라미터 오류
- 401: 권한 없음(client_id, client_key불일치)
- 406: 일치하지 않는 정보
- 500: 시스템 에러 유형 1
- 501: 시스템 에러 유형 2
- -1: 알 수 없는 시스템 에러가 발생하였습니다
- -2: InsideListener 혹은 Context 가 존재하지 않습니다
- 사용 예시
 - Android

```
private final String ID = "YOUR-CLIENT-ID";
private final String KEY = "YOUR-CLIENT-KEY";
private final String SECRET = "YOUR-CLIENT-SECRET";
private final String UUID = ""; // USERID는 클라이언트 마다 다른 값이 설정되어야 함(예: 단말의 MAC 주소로 설정)

String ret = insideSDK.agent_register(ID, KEY, SECRET, USERID);
// ret = "{ \"rc\":200, \"rcMsg\":\"UUID 발급에 성공하였습니다.\", \"UUID\":\"...\"}"

JSONObject jsonObject = new JSONObject(ret);
int rc = jsonObject.getInt("rc");
String rcMsg = jsonObject.getString("rcMsg");

if(rc == 200) { // rc가 200인 경우에만 UUID 발급
    String UUID = jsonObject.getString("UUID");
}
```

- Linux

```
typedef struct {
    int rc;
    std::string rcmsg;
    std::string uuid;
} REGISTER_CODE;

std::string ID = "YOUR-CLIENT-ID";
std::string KEY = "YOUR-CLIENT-KEY";
std::string SECRET = "YOUR-CLIENT-SECRET";
std::string USERID = ""; // USERID는 클라이언트 마다 다른 값이 설정되어야 함(예: 단말의 MAC 주소로 설정)

REGISTER_CODE rc;
result = agent_register(ID, KEY, SECRET, USERID);
if (result.rc != 200) {
    printf("Error: agent_register got %d, %s\n", result.rc, result.rcmsg);
} else {
    printf("agent_register received an uuid [%s]\n", result.uuid);
}
```

```
}
```

- Windows

```
typedef struct {
    int rc;
    std::string rcmsg;
    std::string uuid;
} REGISTER_CODE;

REGESITER_CODE rc;

std::string ID = "YOUR-CLIENT-ID";
std::string KEY = "YOUR-CLIENT-KEY";
std::string SECRET = "YOUR-CLIENT-SECRET";
std::string USERID = ""; // USERID는 클라이언트 마다 다른 값이 설정되어야 함(예:
단말의 MAC 주소로 설정)

REGESITER_CODE rc;
result = agent_register(ID, KEY, SECRET, USERID);
if (rc.rc != 200) {
    sprintf(szLog, "Error: agent_register got rc=%d, rcmsg=%s\n", rc.rc,
rc.rcmsg.c_str());
}
else {
    sprintf(szLog, "agent_register successfully got UUID=%s\n",
rc.uuid.c_str());
}
```

9.3 Agent SDK UUID 확인 요청

Agent SDK를 사용하기 위해서 생성한다. 사용자가 입력한 UUID는 서버에서 정상적으로 발급된 UUID 인지 확인하는 과정을 거친다. 만약 rc가 404로 리턴된 경우에는 앞절에 설명한 uuid 생성 과정을 다시 거쳐야 한다.

- API

agent_init(string client_id, string client_key, string client_secret, string uuid)

- Parameter

이름	유형	설명
client_id	string	기가지니 개발자 포털을 통해 발급 받은 클라이언트 아이디
client_key	string	기가지니 개발자 포털을 통해 발급 받은 클라이언트 키
client_secret	string	기가지니 개발자 포털을 통해 발급 받은 클라이언트 시크릿
uuid	string	agent_register()를 통해 서버로 부터 발급받은 UUID

- Result

이름	유형	설명
rc	int	결과코드
rcmsg	String	결과 메시지

- rc는 다음과 같이 정의 된다.
 - 200: 성공
 - 404: UUID 찾을 수 없음
 - -1: 알 수 없는 시스템 에러가 발생하였습니다.
 - -2: InsideListener 혹은 Context 가 존재하지 않습니다.

- 사용 예시

- Android

```
String ret = insideSDK.agent_init(CLIENT_ID, CLIENT_KEY,
CLIENT_SECRET, uuid);
// ret = "{\"rc\":200,\"resMsg\":\"성공\"}"

JSONObject jsonObject = new JSONObject(ret);
int rc = jsonObject.getInt("rc");
String rcMsg = jsonObject.getString("rcMsg");

if(rc == 200) {
    // agent_init 성공
}
```

- Linux

```
REGESITER_CODE rc;
rc = agent_init(CLIENT_ID, CLIENT_KEY, CLIENT_SECRET, puuid);
if (rc.rc == 200) {
    printf("agent_init successfully got rc=%d\n", rc.rc);
}
else {
    if (rc.rc == 404) {
        printf("agent_init got 404. Please call agent_register()
again!\n");
    }
}
```

- Windows

```

REGISTER_CODE rc;
rc = agent_init(CLIENT_ID, CLIENT_KEY, CLIENT_SECRET, puuid);
if (rc.rc == 200) {
    sprintf("agent_init successfully got rc=%d\n", rc.rc);
}
else {
    sprintf("agent_init got 404. Please call agent_register()
again!\n");
    return;
}

```

9.4 Agent SDK UUID 삭제 요청

발급된 UUID를 삭제 요청한다.

- API

agent_unregister()

- Result

이름	유형	설명
rc	int	결과코드
rcmsg	String	결과 메시지

- rc는 다음과 같이 정의 된다.
 - 200: 성공
 - 404: Client 정보 찾을 수 없음
 - 501: 시스템 오류
 - -1: 알 수 없는 시스템 에러가 발생하였습니다
- 사용 예시
 - Android

```

String ret = insideSDK.agent_unregister();
// ret = "{\"rc\":200,\"resMsg\":\"unRegister에 성공하였습니다.\"}";

JSONObject jsonObject = new JSONObject(ret);
int rc = jsonObject.getInt("rc");
String rcMsg = jsonObject.getString("rcMsg");

if(rc == 200) {
    // agent_unregister 성공
}

```

- Linux

```
REGISTER_CODE rc;
rc = agent_unregister();
if (rc.rc == 200)
{
    // agent_unregister 성공
}
```

- Windows

```
std::string ret;

rc = agent_unregister();
if (rc.rc == 200) {
    // agent_unregister 성공
}
```

9.5 Agent SDK 콜백 등록

서버가 전송하거나 Agent SDK로 부터 API 호출에 대한 이벤트나 커맨드를 수신하기 위해 콜백 함수를 등록한다. 커맨드와 이벤트로 전달되는 데이터는 9.1절을 참고한다. 해당 API는 **Linux, Windows SDK**에서만 사용된다.

- API

agent_setCommandEventCallback(agent_onCommand aoc, agent_onEvent aoe)

- Parameter

이름	유형	설명
aoc	agent_onCommand	커맨드를 받기 위한 콜백 함수
aoe	agent_onEvent	이벤트를 받기 위한 콜백 함수

- 사용 예시

- Linux

```
REGISTER_CODE rc;
rc = agent_init(ID, KEY, SECRET, puuid);

if (rc.rc == 200) {
    agent_setCommandEventCallback(onCommand, onEvent);
}
```

- Windows

```
REGISTER_CODE rc;
rc = agent_init(ID, KEY, SECRET, puuid);

if (rc.rc == 200) {
    agent_setCommandEventCallback(this->onCommand, this->onEvent);
}
```

9.5 Agent SDK 텍스트 명령 요청

서버로 텍스트로 된 명령을 요청한다. 요청이 성공했을 때는 음성 스트림이 전달된다.

- API

agent_sendText(string msg)

- Input Parameter

이름	유형	설명
msg	string	명령 텍스트

- Result

해당 API 결과는 onCommand를 통해 전달된다.

- 사용 예시

- Android

```
insideSDK.agent_sendText("안녕하세요");

@Override
public void onCommand(String cmd) {
    // actionType 및 commandType 으로 분기 필요
    RsResult r = new Gson().fromJson(cmd, RsResult.class);
    String commandType = r.commandType;
    String actionType = r.actionType;

    if(actionType.equals("media_stream") {
        // 처리 필요
    }
}
```

- Linux

```
agent_sendText("안녕하세요");

void onCommand(std::string cmd) {
    cJSON *cmdp_jsonObj = cJSON_Parse(cmd.c_str());
    if (cmdp_jsonObj != NULL) {
        // actionType 및 commandType 으로 분기 필요
        cJSON *cmdp_actionType = cJSON_GetObjectItem(cmdp_jsonObj,
"actionType");
        std::string actionType = cmdp_actionType->valuelstring;
        if (strcmp(actionType.c_str(), "media_stream") == 0) {
            // 처리 필요
        }
    }
}
```

- Windows

```
agent_sendText("안녕하세요");

void CginsidetestDlg::onCommand(std::string cmd) {
    cJSON *cmdp_jsonObj = cJSON_Parse(cmd.c_str());
    if (cmdp_jsonObj != NULL) {
        // actionType 및 commandType 으로 분기 필요
        cJSON *cmdp_actionType = cJSON_GetObjectItem(cmdp_jsonObj,
"actionType");
        std::string actionType = cmdp_actionType->valuelstring;
        if (strcmp(actionType.c_str(), "media_stream") == 0) {
            // 처리 필요
        }
    }
}
```

9.6 Agent SDK 음성인식 시작 요청

음성인식 시작을 요청한다.

- API

agent_startVoice()

- Result

해당 API 결과는 onEvent를 통해 전달된다.

- 사용 예시

- Android

```

insideSDK.agent_startVoice();

@Override
public void onEvent(int evt, String opt) {
    switch (evt) {
        case Inside.VOICE_START:
            // 단말 마이크 on
            // agent_sendVoice 호출
            break;
        ....
    }
}

```

- Linux

```

agent_startVoice();

void onEvent(int evt, std::string opt)
{
    switch (evt) {
        case INSIDE_EVENT::VOICE_START:
            // 단말 마이크 on
            // agent_sendVoice 호출
            break;
        ....
    }
}

```

- Windows

```

agent_startVoice();

void CginsidetestDlg::onEvent(int evt, std::string opt) {
    switch (evt) {
        case INSIDE_EVENT::VOICE_START:
            // 단말 마이크 on
            // agent_sendVoice 호출
            break;
        ....
    }
}

```

9.7 Agent SDK 음성 데이터 전송

음성인식 시작 명령을 받으면 사용자가 발화한 음성 데이터 스트림을 서버로 전송한다. 이 때 3rd Party Application 은 음성인식 파형 등의 UI 작업을 수행하면 된다. 서버로 부터 음성인식 중지 명령이 오면 마이크를 닫는다.

- API

agent_sendVoice(short[] data, int size)

- Input Parameter

이름	유형	설명
data	array	녹음된 음성 스트림. 16kHz Signed 16bit Linear PCM 데이터
size	int	녹음된 음성 스트림 길이 (data 사이즈)

- Result

해당 API 결과는 onEvent를 통해 전달된다.

- 사용 예시

- Android

```
private static final int[] SAMPLE_RATE = new int[]{16000, 11025,
22050, 44100};
private static final int CHANNEL = AudioFormat.CHANNEL_IN_MONO;
private static final int ENCODING = AudioFormat.ENCODING_PCM_16BIT;

// VoiceRecorder Create
final AudioRecord audioRecord = new
AudioRecord(MediaRecorder.AudioSource.VOICE_RECOGNITION,
            sampleRate, CHANNEL, ENCODING, sizeInBytes);
return audioRecord;

// callback 부분에서 agent_sendVoice 호출
VoiceRecorder.Callback callback = new VoiceRecorder.Callback() {
    @Override
    public void onVoice(short[] data, int size) {
        insideSDK.agent_sendVoice(data, size);
    }
}

@Override
public void onEvent(int evt, String opt) {
    switch (evt) {
        case Inside.VOICE_STOP:
            // 단말 마이크 off
            // opt로 전달된 음성인식 결과 출력
            break;
        ....
    }
}
```

- Linux

```

while (...) {
    agent_sendVoice(buf, (int)nread);
}

void onEvent(int evt, std::string opt) {
    switch (evt) {
        case INSIDE_EVENT::VOICE_STOP:
            // 단말 마이크 off
            // opt로 전달된 음성인식 결과 출력
            break;
        ....
    }
}

```

o Windows

```

HRESULT __cdecl RecordCallback(BYTE* pRecordedData, int iSize) {
    short pcm_val[5000];
    unsigned long l_pcm_val = 0;
    int i;

    for (i = 0; i < iSize; i += 2) {
        pcm_val[i / 2] = pRecordedData[i] + pRecordedData[i + 1] * 256;
    }
    for (i = 0; i < iSize / 2; i++) {
        l_pcm_val += (pcm_val[i] > 0 ? pcm_val[i] : -1 * pcm_val[i]);
    }

    agent_sendVoice(pcm_val, iSize / 2);

    return 0;
}

void CginsidetestDlg::onEvent(int evt, std::string opt) {
    switch (evt) {
        case INSIDE_EVENT::VOICE_STOP:
            // 단말 마이크 off
            // opt로 전달된 음성인식 결과 출력
            break;
        ....
    }
}

```

9.8 Agent SDK 음성인식 중지 요청

서버로 음성인식 시작 요청 후 필요 시 음성 중지 요청을 한다.

- API

agent_stopVoice()

- 사용 예시
 - Android

```
insideSDK.agent_stopVoice();
```

- Linux

```
agent_stopVoice();
```

- Windows

```
agent_stopVoice();
```

9.9 Agent SDK TTS 변환 요청

입력한 텍스트를 TTS 스트림으로 변환한다.

- API

agent_getTTS(string msg)

- Parameter

이름	유형	설명
msg	string	TTS 로 변환할 텍스트

- Result

해당 API 결과는 onCommand를 통해 전달된다.

- 사용 예시
 - Android

```

insideSDK.agent_getTTS("안녕");

@Override
public void onCommand(String json) {
    // actionType 및 commandType 으로 분기 필요
    RsResult r = new Gson().fromJson(json, RsResult.class);
    String commandType = r.commandType;
    String actionType = r.actionType;

    if(actionType.equals("media_stream") {
        // 처리 필요
    }
}

```

o Linux

```

agent_getTTS("안녕");

void onCommand(std::string cmd) {
    cJSON *cmdp_jsonObj = cJSON_Parse(cmd.c_str());
    if (cmdp_jsonObj != NULL) {
        // actionType 및 commandType 으로 분기 필요
        cJSON *cmdp_actionType = cJSON_GetObjectItem(cmdp_jsonObj,
"actionType");
        std::string actionType = cmdp_actionType->valuestring;
        if (strcmp(actionType.c_str(), "media_stream") == 0) {
            //처리 필요
        }
    }
}

```

o Windows

```

agent_getTTS("안녕");

void CginsidemfctestDlg::onCommand(std::string cmd) {
    cJSON *cmdp_jsonObj = cJSON_Parse(cmd.c_str());
    if (cmdp_jsonObj != NULL) {
        // actionType 및 commandType 으로 분기 필요
        cJSON *cmdp_actionType = cJSON_GetObjectItem(cmdp_jsonObj,
"actionType");
        std::string actionType = cmdp_actionType->valuestring;
        if (strcmp(actionType.c_str(), "media_stream") == 0) {
            //처리 필요
        }
    }
}

```

9.10 Agent SDK 서비스 로그인

로그인이 필요한 서비스의 로그인페이지 정보를 요청한다. 해당 API를 호출하면 service_type에 따른 로그인 페이지 URL이 전달 되며 사용자가 로그인을 완료하면 입력한 return_url로 페이지가 redirect 된다.

- API

agent_serviceLogin(string service_type, string return_url)

- Parameter

이름	유형	설명
service_type	string	로그인 할 서비스 (예: 지니뮤직)
return_url	string	로그인 완료 후 redirection 할 URL

- Result

이름	유형	설명
rc	number	결과 코드
rcmsg	string	결과 메시지
oauth_url	string	login_type이 oauth 경우 전달됨

- rc는 다음과 같이 정의된다.
 - 200: 성공
 - 401: 허용되지 않는 호출
 - 404: 지원되지 않는 login_type 임
 - 500: 시스템 오류
 - -1: 알 수 없는 시스템 에러가 발생하였습니다.
- 사용 예시
 - Android

```
String ret = insideSDK.agent_serviceLogin("geniemusic", null);
// ret = "{\"rc\":200,\"rcmsg\":\"성공\",\"oauth_url\":\"http://music.login.com\"}"

JSONObject jsonObject = new JSONObject(ret);
int rc = jsonObject.getInt("rc");

if(rc == 200) {
    // oauth url 이동
}
```

- Linux

```
std::string strRet = agent_serviceLogin("geniemusic", "");

if (strRet.size() > 0) {
    cJSON *cmdp_jsonObj = cJSON_Parse(strRet.c_str());
    if (cmdp_jsonObj != NULL) {
        cJSON *cmdp_rc = cJSON_GetObjectItem(cmdp_jsonObj, "rc");

        if (cmdp_rc->valueint == 200) {
            cJSON *oauth_url =
cJSON_GetObjectItem(cmdp_jsonObj, "oauth_url");
            // oauth url 이동
        }
    }
}
```

- Windows

```
std::string strRet = agent_serviceLogin("geniemusic", "");

if (strRet.size() > 0) {
    cJSON *cmdp_jsonObj = cJSON_Parse(strRet.c_str());
    if (cmdp_jsonObj != NULL) {
        cJSON *cmdp_rc = cJSON_GetObjectItem(cmdp_jsonObj, "rc");

        if (cmdp_rc->valueint == 200) {
            cJSON *oauth_url =
cJSON_GetObjectItem(cmdp_jsonObj, "oauth_url");
            // oauth url 이동
        }
    }
}
```

9.11 agent_sendCommand

서버로 하드웨어 및 타이머 Event를 전달하거나 TTS 수신 방식을 설정하고자 할 때 사용한다. 단말단에서 Volume, BT/BLT, Timer, Button 이벤트가 발생한 경우 Agent 서버로 해당 내용을 전송해야 한다.

- API

agent_sendCommand(string cmd)

- Parameter

이름	유형	설명
cmd	string	Agent 서버로 전달할 요청/이벤트 (JSON포맷)

9.11.1 Snd_HWEV

단말 H/W 조작에 의해 발생한 Event를 전달한다.

- cmdOpt: json 형태의 string으로 다음의 값을 가진다.
 - target : String 으로 대상 H/W
 - hwEvent : String 으로 이벤트 유형
 - hwEventOpt : String으로 아래와 같은 유형이 존재한다.

target	hwEvent	hwEventOpt	설명
button	Btn_PU Btn_NX Btn_PV	-	Play/Pause 버튼이 눌러짐 Next 버튼이 눌러짐 Previous 버튼이 눌러짐
bluetooth	BTEvent	event	A2DP 연결 상태를 설정 event: 801(A2DP Play), 800(A2DP Not Connect)
volume	setVolume	value	볼륨 설정으로 value:3 와 같이 전달

sendCommand - Snd_HWEV 예시

```
{
  "commandType":"Snd_HWEV",
  "payload":{
    "cmdOpt": {"target":"volume", "hwEvent":"setVolume", "hwEventOpt":{"value":3}}
  }
}
```

단말 동작: 볼륨 3으로 변경시 서버로 이벤트 전달

9.11.2. Snd_TMEV

설정된 Timer 시간이 도래했을 때 Event를 전달한다.

- cmdOpt: json 형태의 string으로 다음의 값을 가진다.
 - recAct : String 으로 요청하는 Act 유형
 - actionTrx : String 으로 서버로 부터 요청받은 액션 Transaction
 - localTime : String 으로 단말 Event 생성 시간(YYYYMMDDHH24MISS)

sendCommand - Snd_TMEV 예시

```
{
  "commandType": "Snd_TMEV",
  "payload": {
    "cmdOpt": { "reqAct": "ACT_400", "actionTrx": "20190219210350afH",
    "localTime": "20190619103059189" }
  }
}
```

단말 동작: 설정한 타이머 시간이 도래했다는 이벤트를 서버로 전달

9.11.3. Req_DSSP

TTS 수신 방식을 설정하고자 할 때 Command를 전달한다.

- cmdOpt: json 형태의 string으로 다음의 값을 가진다.
 - ttsTransport: string으로 voice/stream 방식을 설정할 수 있다. 기본은 "voice"로 세팅.

sendCommand - Req_DSSP 예시

```
{
  "commandType": "Req_DSSP",
  "payload": {
    "cmdOpt": { "ttsTransport": "stream" }
  }
}
```

단말 동작: TTS수신 시 partial TTS 방식 (8192 byte)으로 변경

9.12 agent_onCommand

3rd Party Application에게 서버로부터 전달받은 응답, 이벤트 및 TTS 정보를 전달한다. 3rd Party Application은 해당 command를 받기 위한 리스너를 구현하여 해당 명령을 처리해야 한다.

- API

agent_onCommand(string cmd)

- Parameter

이름	유형	설명
cmd	string	Agent 서버로부터 전달받은 응답/이벤트/TTS (JSON포맷)

서버는 아래와 같이 6가지 Command를 단말로 전달할 수 있으며 각 actionType에 따라 payload에 전달 내용을 세팅할 수 있다.

9.12.1. start_voice

- actionType: start_voice
- cmdOpt: json 형태의 string으로 다음의 값을 가진다.
 - errorCode: number로 에러 코드
 - playNotiTime: number로 플레이 noti 시간
 - playTime: number로 플레이 시간

onCommand - start_voice 전달 예시

```
{
  "actionType": "start_voice",
  "payload": {
    "cmdOpt": { "errorCode": 0, "playNotiTime": 0, "playTime": 0 }
  }
}
```

연속대화의 경우 서버에서 start_voice를 전달

예) "일정 등록해줘" -> "일정을 등록하실 날짜를 말씀해주세요" -> onCommand로 "start_voice" 전달

단말 동작: Mic On 후 음성인식 시작

9.12.2. dialog_response

- actionType: dialog_response
- commandType: "Req_EXEC"
- dialogResponse: json 형태의 string으로 다음의 값을 가진다.
 - Intent: DialogKit에서 정의한 Intent
 - appInfo: DialogKit에서 정의한 NE 정보

onCommand - dialog_response 전달 예시

```
{
  "actionType": "dialog_response",
  "commandType": "Req_EXEC",
  "dialogResponse": {
    "Intent": "USERGUIDE_ENTER", "appInfo": { "NE-USERGUIDE": "사용자가이드",
    "VSid": "00006" }
  }
}
```

단말 동작: "dialogResponse" 내 json 형식의 대화 답변을 파싱하여 단말 custom 대화 처리

9.12.3. media_stream

TTS 재생하는 text 정보를 아래와 같이 Req_PLMD로 먼저 전달 후 실제 스트림을 전달한다.

- actionType: media_stream
- commandType: "Req_PLMD"
- cmdOpt: json 형태의 string으로 다음의 값을 가진다.
 - channel: number로 출력할 채널 번호
 - metaInfo: String으로 TTS 재생하는 Text 정보를 mesg로 전달
 - actOnOther: String으로 다른 출력 미디어에 대한 action으로 아래와 같다.

Action	Description
mute	다른 미디어가 재생중이면 mute 시키고, TTS 출력 완료 후에도 mute 상태 유지
muteR	다른 미디어가 재생중이면 mute 시킴, TTS 출력 완료 후 unmute 시킴
pause	다른 미디어가 재생중이면 pause 시키고, TTS 출력 완료 후에도 pause 상태 유지
pauseR	다른 미디어가 재생중이면 pause 시킴, TTS 출력 완료 후 다른 미디어 resume 시킴
Stop	다른 미디어가 재생중이면 stop 시킴, 다른 미디어 완전 재생 종료
VolDown	다른 미디어가 재생중이면 volume-down, TTS 출력 완료 후에도 volume-down 상태 유지
VolDownR	다른 미디어가 재생중이면 volume-Down, TTS 출력 완료 후 기존 볼륨 복구

실제 TTS mediastream은 아래와 같이 base64 encoding되어 전달되며 3rd Party Application에서 base64 decoding을 해서 사용한다.

- contentType: String으로 TTS 전송 방식이 설정됨. 다음의 값을 가진다.
 - pcm_stream: 8192 byte 씩 partial TTS 전송 (44 byte header 제외)
 - wav: 전체 wav 파일 (44 byte header 포함)
- end: String으로 **contentType**이 "**pcm_stream**"인 경우에만 사용. 0: 전송 시작, 1: 전송 중, 2: 전송 완료
- mediastream: String으로 음성 byte array가 전달됨

onCommand - media_stream 전달 예시

```
{
  "actionType": "media_stream",
  "commandType": "Req_PLMD",
  "payload": {
    "cmdOpt": { "channel": 0, "actOnOther": "stop", "metaInfo": { "mesg": "지니뮤직을 실행합니다" } }
  }
}
{
  "actionType": "media_stream",
  "contentType": "pcm_stream",
  "end": 0,
  "mediastream": "UklGRkTfAABXQVZFZm10IBAAAAABAAEAgD4AAAB9AABBCC....."
}
```

단말 동작: UI 작업이 필요한 경우 metaInfo 내 mesg로 전달 받은 TTS text 정보를 화면에 표시 후, mediastream으로 전달 받은 TTS stream을 재생

9.12.4. media_url

곡이 내려오는 경우 아래와 같이 URL이 전달된다. media_url이 전달될 때 playNotiTime에 설정된 값이 있으면, agent_updateMediaStatus API로 설정된 playNotiTime에 서버로 noti를 해주어야 한다.

- actionType: media_url
- commandType: "Req_PLMD"
- cmdOpt: json 형태의 string으로 다음의 값을 가진다.
 - channel: number로 출력할 채널 번호
 - actOnOther: String으로 다른 출력 콘텐츠에 대한 action (10.2.3 표 참고)
 - url: String으로 재생할 URL
 - playNotiTime: 미디어 재생일 경우 number로 재생 시간 noti해주는 주기 (지니뮤직은 60초 설정됨) (단위: second)
 - metaInfo: String으로 콘텐츠 메타 정보가 설정됨. 지니뮤직, 팟캐스트, 라디오, 캘린더, 날씨, 지역 정보, 위키디피아, 환율, 버스도착안내, 길안내, 알람/타이머에 대한 정보 설정 [9.12.8 metaInfo 상세설명 참고](#)

onCommand - media_url 전달 예시

```
{
  "actionType": "media_url",
  "commandType": "Req_PLMD",
  "payload": {
    "cmdOpt": { "channel": 1, "actOnOther": "pause", "metaInfo": { "mesg": "TTS 메시지",
      "actFeel": "Happy", "sentencePattern": "Describe", "contentType": "music",
      "infoType": "text", "infoDetail": { "title": "곡명", ... } }
    }
  }
}
```

단말 동작: UI가 필요한 경우 metaInfo로 전달 받은 미디어 정보를 화면 표시 후, 전달 받은 url을 재생

사용자가 발화로 음악 중지를 요청한 경우 재생중인 곡을 제어하는 명령이 내려온다. 제어 명령은 아래와 같이 미디어 처리 액션이 내려온다.

- `commandType`: "Req_UPMD"
- `cmdOpt`: json 형태의 string으로 다음의 값을 가진다.
 - `channel`: number로 출력할 채널 번호
 - `playTime`: number로 플레이 시간 (단위: millisecond)
 - `act`: String으로 미디어에 대한 변경 action

Action	Description
pause	일시중지(향후 resume 받아서 resume가능한 상태 유지)
stop	중지(향후 resume 명령 불가)
resume	pause 시 resume 처리
seek	특정 시간대로 이동(이 경우 playTime 전달)

onCommand - media_url 전달 예시

```
{
  "actionType": "media_url",
  "commandType": "Req_UPMD",
  "payload": {
    "cmdOpt": { "channel": 1, "act": "pause", "playTime": -1 }
  }
}
```

단말 동작: act로 전달받은 액션을 수행 (재생중인 음악 재생 중지)

9.12.5. hardware_control

- `actionType`: hardware_control
- `commandType`: "Req_HWCL"
- `cmdOpt`: json 형태의 string으로 다음의 값을 가진다.
 - `target`: String으로 대상 H/W (예: volume, bluetooth)
 - `hwCmd`: String으로 요청 Command(예: setVolume)
 - `hwCmdOpt`: String으로 요청 Command Option
 - `trxid`: 요청 Transaction Id

target	hwCmd	hwCmdOpt	설명
bluetooth	controlClassicSvc	control: play(재생), stop(중지), pause(일시정지), rewind(처음으로), forward(다음곡), backward(이전곡)	Bluetooth 제어 명령을 설정한다.
volume	setVolume	control: UP(올림), DN(내림), MT(뮤트), UMT(뮤트 해 제), ST(값으로 설정)	볼륨 제어명령 을 설정한다.

onCommand - hardware_control 전달 예시

```
{
  "actionType": "hardware_control",
  "commandType": "Req_HWCL",
  "payload": {
    "cmdOpt": { "target": "volume", "hwCmd": "setVolume", "hwCmdOpt":
    { "control": "UP", "trxid": "2019021921020350338Q2IPafH" }
  }
}
```

단말 동작: 전달받은 볼륨 제어 정보를 수행 (볼륨 업)

9.12.6. timer_set

- actionType: timer_set
- commandType: "Req_STTM"
- cmdOpt: json 형태의 string으로 다음의 값을 가진다.
 - setOpt: String으로 설정 옵션(set: 설정, clear: 해제)
 - reqAct: String으로 요청하는 Act 유형(예: ACT_400)
 - actionTrx: String으로 설정 액션 Transaction
 - setTime: String으로 설정 시간(YYYYMMDDHH24MISS)

onCommand - timer_set 전달 예시

```
{
  "actionType": "timer_set",
  "commandType": "Req_STTM",
  "payload": {
    "cmdOpt": { "setOpt": "set", "reqAct": "ACT_400",
    "actionTrx": "2019021921020350338Q2IPafH", "setTime": "20190619103059189" }
  }
}
```

단말 동작: 단말에 6월 19일 10시 30분 59초에 타이머 설정

9.12.7. webview_url

- actionType: webview_url
- commandType: "Req_OAuth"
- cmdOpt: json 형태의 string으로 다음의 값을 가진다.
 - oauth_url: String으로 로그인 페이지 URL

onCommand - webview_url 전달 예시

```
{
  "actionType": "webview_url",
  "commandType": "Req_OAuth",
  "payload": {
    "cmdOpt": { "oauth_url": "http://www.genie.com/login" }
  }
}
```

단말 동작: 전달받은 url을 웹뷰로 실행

9.12.8. metaInfo 상세설명

- Req_PLMD에 메타 정보는 다음과 같은 형태로 전송된다.

```
{
  "payload": {
    "cmdOpt": {
      "channel": 1,
      "actOnOther": "pause",
      "metaInfo": {
        ...
      }
    }
  }
}
```

- metaInfo 구조는 다음과 같다.
 - TTS 메시지 외에 메타 정보가 없는 경우

```
{
  "metaInfo": {
    "mesg": "TTS 메시지",
    "actFeel": "Happy",
    "sentencePattern": "Describe",
    "contentType": "wikipedia"
  }
}
```

- 콘텐츠에 대한 상세 내용을 텍스트로 전달하는 경우

```
{
  "metaInfo": {
    "mesg": "TTS 메시지",
    "actFeel": "Happy",
    "sentencePattern": "Describe",
    "contentType": "music",
    "infoType": "text",
    "infoDetail": {
      "title": "곡명",
      ...
    }
  }
}
```

- 콘텐츠에 대한 상세 내용을 url로 전달하는 경우

```
{
  "metaInfo": {
    "mesg": "TTS 메시지",
    "actFeel": "Happy",
    "sentencePattern": "Describe",
    "infoType": "url",
    "infoDetail": {
      "url": "http://agent.gigagenie.ai/[uuid]",
      "ctx": "contextUuid"
    }
  }
}
```

- 서비스 별 contentType은 다음 표를 참고한다.

서비스	contentType 값
지니뮤직	music
팟캐스트	podcast
라디오	radio
캘린더	calendar
날씨	weather
지역정보	localInfo
위키피디아	Wikipedia
환율	exchangeRate
버스도착안내	busArrivalInfo
길안내	directions
알람/타이머	* TBD

- 서비스 별 infoDetail 구조는 [\[첨부 2\] 서비스 별 infoDetail](#) 구조를 참고한다.

9.13 agent_onEvent

3rd Party Application에게 SDK 이벤트를 전달한다. 음성인식 시작/종료, 기타 에러코드 등을 전달한다. 3rd Party Application은 해당 event를 전달받으면 상황에 따른 적절한 예외처리 또는 UI 표기를 할 수 있다.

- API

agent_onEvent(int evt, string opt)

- Input Parameter

이름	유형	설명
evt	int	이벤트명
opt	string	이벤트에 따라 전달되는 추가 정보 (JSON포맷)

- 사용 예시

- Android

```
@Override
public void agent_onEvent(int evt, String opt) {
    switch (evt) {
        case Inside.VOICE_START:
            // 마이크 On
            Inside.sendVoice();
    }
}
```



```

        ...
        break;
    case Inside.VOICE_STOP:
        // 마이크 off
        ...
        break;
    case Inside.SERVER_ERROR:
        // 에러 처리
        ...
        break;
    }
}

```

◦ Linux

```

void init() {
    rc = agent_init(ID, KEY, SECRET, puuid);
    if (rc.rc == 200) {
        agent_setCommandEventCallback(onCommand, onEvent);
    }
}

void onEvent(int evt, std::string opt) {
    switch (eventMask) {
        case INSIDE_EVENT::VOICE_START:
            // 마이크 on
            Inside.sendVoice();
            ...
            break;
        case INSIDE_EVENT::VOICE_STOP:
            // 마이크 off
            ...
            break;
        case INSIDE_EVENT::SERVER_ERROR:
            // 에러 처리
            ...
            break;
    }
}

```

◦ Windows

```

rc = agent_init(ID, KEY, SECRET, puuid);
if (rc.rc == 200) {
    agent_setCommandEventCallback(onCommand, onEvent);
}

void CginsidemfctestDlg::onEvent(int evt, std::string opt) {

```

```

switch (eventMask) {
    case INSIDE_EVENT::VOICE_START:
        // 마이크 On
        Inside.sendVoice();
        ...
        break;
    case INSIDE_EVENT::VOICE_STOP:
        // 마이크 Off
        ...
        break;
    case INSIDE_EVENT::SERVER_ERROR:
        // 에러 처리
        ...
        break;
}
}

```

9.14 Agent SDK 리셋

Agent SDK를 초기화하기 위해서 사용한다. 해당 API 호출 시 3rd Party Application은 인스턴스 초기화 및 미디어 플레이어 초기화 등 모든 값을 초기화 하여야 한다. 이후 다시 Agent SDK 를 사용하기 위해서는 Agent Init를 호출하여야 한다.

- API
 - agent_reset()**
- 사용 예시
 - Android

```

if(insideSDK != null)
    insideSDK.agent_reset();
insideSDK = null;
// 미디어 플레이어 등 초기화 필요
// 테스트앱의 MainActivity resetInsideSDK() 메소드 참조

```

- Linux

```

agent_reset();
// 미디어 플레이어 등 초기화 필요
// 테스트앱의 MainActivity resetInsideSDK() 메소드 참조

```

- Windows

```
agent_reset();  
// 미디어 플레이어 등 초기화 필요  
// 테스트앱의 MainActivity resetInsideSDK() 메소드 참조
```

9.15 Agent SDK 디버그 모드 설정

Agent SDK 디버그 모드를 설정한다. true로 설정 시 Log를 확인할 수 있다.

- API
agent_debugmode(boolean enable)
- Parameter

이름	유형	설명
enable	boolean	0:디버그 모드 해제, 1: 디버그 모드

- 사용 예시
 - Android

```
insideSDK.agent_debugmode(1);
```

- Linux

```
agent_debugmode(true);
```

- Windows

```
agent_debugmode(true);
```

9.16 Agent SDK 미디어 상태 업데이트

재생중인 미디어 상태 정보를 전달한다. 재생되는 모든 미디어의 상태를 SDK 로 전달하여야 하며, 미디어 재생 완료 후 이전에 재생하던 미디어가 있을 시에는 onCommand로 다른 미디어를 제어하는 명령이 전달 될 수 있다.

- API
agent_updateMediaStatus(int ch, string state, int playtime)
- Parameter

이름	유형	설명
ch	int	서버에서 전달받은 채널 값
state	string	미디어의 상태를 전달한다. 상태는 다음과 같다. started(시작됨), paused(중지됨), resumed(재시작됨), complete(플레이 완료), stopped(외부 요인에 의한 종료), noti(playNotiTime에 의한 noti)
playtime	int	미디어 재생 시간. 시작 등 값이 없을 때는 0 전달

- Result

해당 API 결과는 onCommand를 통해 전달될 수 있다.

- 사용 예시

- Android

```
insideSDK.agent_updateMediaStatus(0,"started",0); //TTS 재생 시작
insideSDK.agent_updateMediaStatus(0,"noti",30); //agent_onCommand로
"playNotiTime"이 30으로 내려왔다면 noti 시간이 도래했을 때 서버로 noti 정보 전달
```

- Linux

```
agent_updateMediaStatus(0,"started",0); //TTS 재생 시작
```

- Windows

```
agent_updateMediaStatus(0,"started",0); //TTS 재생 시작
```

9.17 Agent SDK 서버 정보 변경

통신할 서버의 정보를 설정한다. 해당 API 호출 후에는 반드시 다시 init 과정을 거쳐야 한다.

- API

agent_setServerInfo(string serverIP, string grpcPort, string restPort)

- Input Parameter

이름	유형	설명
serverIP	String	통신할 서버의 주소
grpcPort	String	grpc 통신 시 사용할 Port
restPort	String	rest 통신 시 사용할 Port

- 사용 예시

- Android

```
if(insideSDK != null) {
    // insideSDK 가 존재한다면 인스턴스 재생성 및 인증
} else {
    // 존재하지 않다면, 서버 정보만 업데이트
    insideSDK.agent_setServerInfo("dev.gigagenie.ai", "50089",
    "10089");
}
```

- Linux

```
agent_setServerInfo("dev.gigagenie.ai", "50089", "10089");
```

- Windows

```
agent_setServerInfo("dev.gigagenie.ai", "50089", "10089");
```

9.18 Agent SDK 버전 조회

Agent SDK 버전을 조회한다.

- API

agent_getVersion()

- 사용 예시

- Android

```
string ver = insideSDK.agent_getVersion(); // ver = "0.9.0beta"
```

- Linux

```
printf("ver = %s\n", agent_getVersion()); // ver = "0.9.0beta"
```

- Windows

```
sprintf("ver = %s\n", agent_getVersion()); // ver = "0.9.0beta"
```

9.19 Agent SDK 위치 정보 설정

Agent SDK 로 위치 정보를 설정한다. 위치 정보 값을 초기화 하려면 모든 값을 빈값으로 세팅한다.

- API

agent_setLocation(String Longitude, String Latitue, String Address)

- Input Parameter

이름	유형	설명
Longitude	String	경도 값
Latitude	String	위도 값
Address	String	주소 값

- 사용 예시

- Android

```
// 위치값 설정
insideSDK.agent_setLocation("127.029000", "37.4713370", "서울시 서초구 태봉로");

// 위치값 초기화
insideSDK.agent_setLocation("", "", "");
```

- Linux

```
agent_setLocation("127.029000", "37.4713370", "서울시 서초구 태봉로");
```

- Windows

```
agent_setLocation("127.029000", "37.4713370", "서울시 서초구 태봉로");
```

10. KWS API 상세

10.1 KWS SDK 초기화

호출어 라이브러리 객체를 생성한다. 해당 API는 agent_init 완료 후 사용할 수 있다.

- API

kws_init(int kwsId)

- Parameter

이름	유형	설명
kwsId	int	Optional 값으로 세팅할 키워드 (dafault: 기가지니) 0: 기가지니, 1: 지니야, 2: 친구야, 3: 자기야

- Result

이름	유형	설명
result_code	int	객체 생성 성공 여부 0:success, -1: fail, -2: unauthorized

- 사용 예시

- Android

```
insideSDK.kws_init();
or
insideSDK.kws_init(0);
```

- Linux

```
int ret = kws_init();
```

- Windows

```
int ret = kws_init();
```

10.2 KWS SDK 호출어 데이터 전송

호출어 인식을 위한 음성 데이터 stream을 전송한다. 호출어가 검출되면 서버로 음성인식 시작 요청을 한다.

- API

int **kws_detect**(short[] data, int size)

- Parameter

이름	유형	설명
data	short[]	16kHz Signed 16bit Linear PCM 데이터
size	int	PCM 입력 sample 개수 (data 사이즈)

- Result

이름	유형	설명
result_code	int	호출어 인식 결과

- result_code는 다음과 같다.

- -1: 실패
 - 0: 호출어 감지되지 않음
 - 1: 호출어 감지 시작
 - 2: 호출어 감지 중

- 3: 호출어 감지 종료
- 4: 호출어 검출 성공
- 사용 예시
 - Android

```
AudioRecord recorder = new audioRecord(...);
recorder.startRecording();
while(!stop) {
    ...
    ret = insideSDK.detect(data, size);
    if (ret == 4) {
        // 호출어 인식됨
    }
}
```

- Linux

```
ret = kws_detect(pcm_buf, pcm_size);
if (ret == KWS_DET_DETECTED) {
    printf("Now Detected!\n");
}
```

- Windows

```
ret = kws_detect(pcm_val, iSize / 2);
if (ret == KWS_DET_DETECTED) {
    printf("Now Detected!\n");
}
```

10.3 KWS SDK 리셋

호출어 라이브러리 객체를 초기화한다.

- API
 - void **kws_reset()**
- 사용 예시
 - Android

```
insideSDK.kws_reset();
```

- Linux

```
kws_reset();
```


- Windows

```
kws_reset();
```

10.4 KWS SDK 에러

호출어 라이브러리 초기화나 연산 과정의 에러를 리턴받는다. 정상일 경우 0이 리턴 된다.

- API
int **kws_error()**
- Result

이름	유형	설명
error_code	int	호출어 라이브러리 에러코드

- 사용 예시
 - Android

```
int ret = insideSDK.kws_getError();
```

- Linux

```
int ret = kws_error();
```

- Windows

```
int ret = kws_error();
```

10.5 KWS SDK 버전 조회

호출어 라이브러리 버전 정보를 조회한다.

- API
String **kws_getVersion()**
- Result

이름	유형	설명
version_code	string	호출어 라이브러리 버전 정보

- 사용 예시

- Android

```
String version = insideSDK.kws_getVersion();
```

- Linux

```
std::string version = kws_getVersion();
```

- Windows

```
std::string version = kws_getVersion();
```

10.6 KWS SDK 키워드 조회

현재 호출어 라이브러리에서 사용 중인 호출어 ID를 조회한다.

- API

int **kws_getKeyword()**

- Result

이름	유형	설명
kwsId	int	호출어 ID 0: 기가지니, 1: 지니야, 2: 친구야, 3: 자기야

- 사용 예시

- Android

```
int kwsId = insideSDK.kws_getKeyword();
```

- Linux

```
int kwsId = kws_getKeyword();
```

- Windows

```
int kwsId = kws_getKeyword();
```

10.7 KWS SDK 키워드 변경

사용할 호출어를 변경한다.

- API

int **kws_setKeyword**(int kwsId)

- Parameter

이름	유형	설명
kwsId	int	변경할 호출어 ID 0: 기가지니, 1: 지니야, 2: 친구야, 3: 자기야

- Result

값	유형	설명
0	int	변경 성공

- 사용 예시

- Android

```
int ret = insideSDK.kws_setKeyword(0);
```

- Linux

```
int ret = kws_setKeyword(1);
```

- Windows

```
int ret = kws_setKeyword(1);
```

10.8 KWS SDK 모델 경로 변경

Linux, Window에서만 제공되는 함수로, 호출어 모델의 디렉토리를 설정하는 함수이다. 기본적으로 모델 파일들이 실행 파일의 경로인 ./conf에 존재한다고 가정한다. 만약 경로를 변경하고 싶다면 kws_init() 함수를 호출하기 전에 해당 함수를 호출하여야 한다.

- API

int **kws_setModelPath**(string path)

- Parameter

이름	유형	설명
path	string	호출어 모델 파일이 존재하는 디렉토리를 설정하는 함수로서 절대 및 상대 경로 모두 사용 가능

- Result

값	유형	설명
0	int	성공
1	int	실패 - path가 존재하지 않거나 디렉토리가 아닌 경우

- 사용 예시

- Linux

```
int ret = kws_setModelPath("./kws/conf");
```

- Windows

```
int ret = kws_setModelPath("./kws/conf");
```

11. 개발 참고 사항 및 이슈 사항

개발 참고 사항 및 이슈 사항에 대해 설명한다.

11.1. 예외처리

11.2. FAQ

[첨부 1] onEvent

음성인식 시작/종료 이벤트 및 서버 에러 발생 시 서버에서 단말로 이벤트를 전달한다.

evt	opt	설명	3rd Party 처리
VOICE_START	-	음성인식 시작	Mic on 후 sendVoice 시작
VOICE_STOP	{"resultCd":200, "recognizedText":"안녕하세요"}	음성인식 종료, 음성인식 결과 전달	Mic off
SERVER_ERROR	{"errorCode":900, "errorMsg":"음성인식 서버 접속 실패"}	서버 에러	에러코드에 맞게 재시도 or 클라이언트 초기화 후 재시도

SERVER_ERROR 이벤트는 다음과 같은 errorCode를 가진다.

- 900: 음성인식 서버 접속 실패
- 901: 음성인식 시작 후 5초 이내에 실제 음성 감지가 발생하지 않는 경우

- 902: 음성인식 시작 후 10초 이내에 음성 끝 감지가 발생하지 않는 경우
- 903: 음성인식 라이브러리 초기화 실패
- 910: TTS 서버 접속 실패
- 911: TTS 변환 요청 실패
- 912: TTS 요청 텍스트의 포맷이 잘못되었거나 빈 값인 경우
- 920: 대화서버 접속 실패
- 921: 클라이언트 타입에 상응하는 대화서버 프로파일 없음
- 930: Filter 서버 접속 실패

[첨부 2] 서비스 별 infoDetail 구조

metalInfo 내 infoDetail은 서비스 별로 아래와 같은 구조를 가진다.

- 지니뮤직

```
{
  "title": "곡명",
  "artist": "아티스트명",
  "duration": "재생 시간",
  "imageUrl": "앨범 이미지 URL"
}
```

- 라디오

```
{
  "title": "라디오 채널 이름",
}
```

- 팟캐스트

```
{
  "title": "에피소드명",
  "publisher": "팟캐스트 제공자 (채널 이름)",
  "description": "채널 설명",
  "category": "카테고리",
  "imageUrl": "앨범 이미지 URL",
  "pubDate": "방송일",
  "duration": "재생 시간"
}
```

- 알람/타이머 (TBD)

```
{
}
```

- 캘린더(TBD)

```
{  
}
```

- 날씨

날씨 정보의 기본 metaInfo 구조는 다음과 같다.

```
{  
  "metaInfo": {  
    "mesg": "오늘 서초구 날씨입니다...",  
    "actFeel": "Neutral",  
    "sentencePattern": "What_question",  
    "contentType": "weather",  
    "infoType": "text",  
    "infoDetail": {  
      "type": "today",  
      "description": "오늘 서울특별시 서초구 태봉로 날씨입니다",  
      ...  
    }  
  }  
}
```

날씨 정보 metaInfo 내 전달되는 infoDetail에는 오늘 날씨(w3), 내일 날씨(w4), 특정일 날씨(w5), 주말 날씨(w6, w7) 이번주/다음주 날씨(w8)정보가 상세히 전달된다.

1. 오늘 날씨(Today Type : w3)의 infoDetail 설명
 - weatherList[]는 총 8개의 배열로 구성된다.

값	설명	대화서버 데이터명
type	"today", 날씨 정보 종류	totype
description	날짜, 일자, 지역 메시지	tomsg
area	지역	toarea
forecast	날씨 예보(맑음/흐림/구름조금 등)	towfk
forecastDetail	날씨 예보 내용	tocast
icon	날씨 아이콘 코드 (1~22)	toicon
temp	현재온도 (°C)	totemp
tempHigh	최고온도 (°C)	totmx
tempLow	최저온도 (°C)	totmn
precipitationChance	강수확률 (%)	topop
precipitation	강수량 (mm)	torainfall
windSpeed	풍속 (m/s)	tows
windDirection	풍향 코드	towd
humidity	습도 (%)	toreh
pm10	미세먼지 수치 ($\mu\text{g}/\text{m}^3$)	topm10
pm25	초미세먼지 수치 ($\mu\text{g}/\text{m}^3$)	topm25
ozone	오존 수치 (ppm)	too3
uv	자외선 수치	toh0
weatherList[]	요청 시간 이후의 시간대별(3시간 단위) 날씨 정보 배열	
weatherList[n].time	시간대	titime1
weatherList[n].tempHigh	예상 최고 온도 (°C)	titmx1
weatherList[n].icon	날씨 아이콘 코드 (1~22)	tiicon1
weatherList[n].forecast	날씨 예보 코드	tiwfk1
weatherList[n].precipitationChance	예상 강수확률 (%)	tipop1
updated.time	오늘 시간대별 날씨 업데이트 시간 (MM/DDHH:mm)	totimeupdate
updated.day	오늘 상세날씨 업데이트 시간	todayupdate
updated.pm	미세먼지/초미세먼지 업데이트 시간	topmupdate
updated.dust	황사 업데이트 시간	todustupdate
updated.ozon	오존 업데이트 시간	toozonupdate
updated.uv	자외선 업데이트 시간	touvupdate

○ 예시

```
{
  "type": "today",
  "description": "오늘 서울특별시 서초구 태봉로 날씨입니다",
}
```

```

"area": "서울특별시 서초구 태봉로",
"forecast": "구름조금",
"forecastDetail": "맑음",
"icon": 2,
"temp": 27,
"tempHigh": 26,
"tempLow": 13,
"precipitationChance": 10,
"precipitation": 35,
"windSpeed": 3,
"windDirection": 7,
"humidity": 30,
"pm10": 42,
"pm25": 15,
"ozone": 0.027,
"uv": 0,
"weatherList": [{
  "time": "지금",
  "tempHigh": 24,
  "icon": 1,
  "forecast": "맑음",
  "precipitationChance": 0
}, {
  "time": "21시",
  "tempHigh": 18,
  "icon": 2,
  "forecast": "구름조금",
  "precipitationChance": 0
}, {
  "time": "내일 0시",
  "tempHigh": -2.0,
  "icon": 1,
  "forecast": "맑음",
  "precipitationChance": 0
}],
"updated": {
  "time": "05/21 10:00",
  "day": "05/21 10:00",
  "pm": "05/21 10:00",
  "dust": "05/21 10:00",
  "ozon": "05/21 10:00",
  "uv": "05/21 10:00"
}
}

```

2. 내일/모레 날씨(Tomorrow Type : w4)의 infoDetail 설명

- 기본적으로 오전 날씨를 보여준다. (오전 날씨 데이터는 9시 기준)
- 오후 날씨 정보는 weatherPm{}으로 구성된다. (오후 날씨 데이터는 15시 기준)

- weatherList[]는 총 8개의 배열로 구성된다.

값	설명	대화서버 데이터명
type	"tomorrow", 날씨 정보 종류	totype
description	날짜, 일자, 지역 메시지	tomsg
area	지역	toarea
forecast	날씨 예보(맑음/흐림/구름조금 등)	towfk
forecastDetail	날씨 예보 내용	tocast
icon	날씨 아이콘 코드 (1~22)	toicon
temp	내일 오전 온도 (°C)	totemp
precipitationChance	강수확률 (%)	topop
windSpeed	풍속 (m/s)	tows
windDirection	풍향 코드	towd
humidity	습도 (%)	toreh
pm10	미세먼지 수치 ($\mu\text{g}/\text{m}^3$)	topm10
pm25	초미세먼지 수치 ($\mu\text{g}/\text{m}^3$)	topm25
ozone	오존 수치 (ppm)	too3
uv	자외선 수치	toh0
weatherPm {}	오후(15시 기준) 날씨 정보 객체	
weatherPm.forecast	오후 날씨 예보(맑음/흐림/구름조금 등)	towfk
weatherPm.icon	오후 날씨 아이콘 코드 (1~22)	toicon
weatherPm.temp	오후 온도 (°C)	totemp
weatherPm.precipitationChance	오후 강수확률 (%)	topop
weatherPm.windSpeed	오후 풍속 (m/s)	tows
weatherPm.windDirection	오후 풍향 코드	towd
weatherPm.humidity	오후 습도 (%)	tohum
weatherPm.pm10	오후 미세먼지 수치 ($\mu\text{g}/\text{m}^3$)	topm10
weatherPm.pm25	오후 초미세먼지 수치 ($\mu\text{g}/\text{m}^3$)	topm25
weatherPm.ozone	오후 오존 수치 (ppm)	too3
weatherPm.uv	오후 자외선 수치	toh0
weatherList[]	시간대별(3시간 단위) 날씨 정보 배열	
weatherList[n].time	시간대	titime1
weatherList[n].tempHigh	예상 최고 온도 (°C)	titmx1
weatherList[n].icon	날씨 아이콘 코드 (1~22)	tiicon1
weatherList[n].forecast	날씨 예보	tiwfk1
weatherList[n].precipitationChance	예상 강수확률 (%)	tipop1
updated.time	내일 시간대별 날씨 업데이트 시간 (MM/DDHH:mm)	totimeupdate

updated.day	내일 상세날씨 업데이트 시간	todayupdate
updated.pm	미세먼지/초미세먼지 업데이트 시간	topmupdate
updated.dust	황사 업데이트 시간	todustupdate
updated.ozon	오존 업데이트 시간	toozonupdate
updated.uv	자외선 업데이트 시간	touvupdate

○ 예시

```
{
  "type": "tomorrow",
  "description": "내일 서울특별시 서초구 태봉로 날씨입니다",
  "area": "서울특별시 서초구 태봉로",
  "forecast": "맑음",
  "forecastDetail": "맑음",
  "icon": 1,
  "temp": 21,
  "precipitationChance": 10,
  "windSpeed": 3,
  "windDirection": 4,
  "humidity": 30,
  "pm10": 42,
  "pm25": 15,
  "ozone": 0.027,
  "uv": 0,
  "weatherPm": {
    "forecast": "구름조금",
    "icon": 2,
    "temp": 28,
    "precipitationChance": 0,
    "windSpeed": 3.3,
    "windDirection": 7,
    "humidity": 30,
    "pm10": 36,
    "pm25": 8,
    "ozone": 0.015,
    "uv": 0
  },
  "weatherList": [{
    "time": "지금",
    "tempHigh": 26,
    "icon": 1,
    "forecast": "맑음",
    "precipitationChance": 0
  }, {
    "time": "0시",
    "tempHigh": 15,
    "icon": 1,
    "forecast": "맑음",
```

```

        "precipitationChance": 0
    }, {
        "time": "3시",
        "tempHigh": 13,
        "icon": 1,
        "forecast": "맑음",
        "precipitationChance": 0
    }, {
        "time": "6시",
        "tempHigh": 13,
        "icon": 1,
        "forecast": "맑음",
        "precipitationChance": 0
    } ],
    "updated": {
        "time": "05/21 10:00",
        "day": "05/21 10:00",
        "pm": "05/21 10:00",
        "dust": "05/21 10:00",
        "ozon": "05/21 10:00",
        "uv": "05/21 10:00"
    }
}

```

3. 특정일 날씨(Day Type : w5)의 infoDetail 설명

- 오전 날씨 데이터는 9시 기준, 오후 날씨 데이터는 15시 기준이다.

값	설명	대화서버 데이터명
type	"day", 날씨 정보 종류	totype
description	날짜, 일자, 지역 메시지	tomsg
area	지역	toarea
forecast	날씨 예보(맑음/흐림/구름조금 등)	towfk
forecastDetail	날씨 예보 내용	tocast
icon	날씨 아이콘 코드 (1~22)	toicon
temp	내일 오전 온도 (°C)	totemp
precipitationChance	강수확률 (%)	topop
windSpeed	풍속 (m/s)	tows
windDirection	풍향 코드	towd
humidity	습도 (%)	toreh

pm10	미세먼지 수치 ($\mu\text{g}/\text{m}^3$)	topm10
pm25	초미세먼지 수치 ($\mu\text{g}/\text{m}^3$)	topm25
ozone	오존 수치 (ppm)	too3
uv	자외선 수치	toh0
weatherPm {}	오후(15시 기준) 날씨 정보 객체	
weatherPm.forecast	오후 날씨 예보(맑음/흐림/구름조금 등)	towfk
weatherPm.icon	오후 날씨 아이콘 코드 (1~22)	toicon
weatherPm.temp	오후 온도 ($^{\circ}\text{C}$)	totemp
weatherPm.precipitationChance	오후 강수확률 (%)	topop
weatherPm.windSpeed	오후 풍속 (m/s)	tows
weatherPm.windDirection	오후 풍향 코드	towd
weatherPm.humidity	오후 습도 (%)	tohum
weatherPm.pm10	오후 미세먼지 수치 ($\mu\text{g}/\text{m}^3$)	topm10
weatherPm.pm25	오후 초미세먼지 수치 ($\mu\text{g}/\text{m}^3$)	topm25
weatherPm.ozone	오후 오존 수치 (ppm)	too3
weatherPm.uv	오후 자외선 수치	toh0

○ 예시

```
{
  "type": "day",
  "description": "05월 28일 서울특별시 서초구 태봉로 날씨입니다.",
  "area": "서울특별시 서초구 태봉로",
  "forecast": "맑음",
  "forecastDetail": "맑음",
  "icon": 1,
  "temp": 21,
  "precipitationChance": 10,
  "windSpeed": 3,
  "windDirection": 4,
  "humidity": 30,
  "pm10": 42,
  "pm25": 15,
  "ozone": 0.027,
  "uv": 0,
  "weatherPm": {
    "forecast": "구름조금",
```

```

        "icon": 2,
        "temp": 28,
        "precipitationChance": 0,
        "windSpeed": 3.3,
        "windDirection": 7,
        "humidity": 30,
        "pm10": 36,
        "pm25": 8,
        "ozone": 0.015,
        "uv": 0
    }
}

```

4. 주말 날씨(Weekend Type : w6, w7)의 infoDetail 설명

- 기본적으로 토요일 날씨를 보여주며 일요일 날씨 정보는 weatherSun {}으로 구성되어 있다.

값	설명	대화서버 데이터명
type	"day", 날씨 정보 종류	totype
description	날짜, 일자, 지역 메시지	tomsg
area	지역	toarea
forecast	날씨 예보(맑음/흐림/구름조금 등)	towfk
forecastDetail	날씨 예보 내용	tocast
icon	날씨 아이콘 코드 (1~22)	toicon
tempHigh	토요일 최고온도 (°C)	totmx
tempLow	토요일 최저온도 (°C)	totmn
precipitationChance	토요일 강수확률 (%)	topop
weatherSun {}	주말[토요일, 일요일] 날씨 정보 객체	
weatherSun.forecast	일요일 날씨 예보(맑음/흐림/구름조금 등)	towfk
weatherSun.icon	일요일 날씨 아이콘 코드 (1~22)	toicon
weatherSun.tempHigh	일요일 최고온도 (°C)	totmx
weatherSun.tempLow	일요일 최저온도 (°C)	totmn
weatherSun.precipitationChance	일요일 강수확률 (%)	topop
weatherSun.windSpeed	일요일 예상 풍속 (m/s)	tows

○ 예시

```
{
  "type": "weekend",
  "description": "이번주 주말 서울특별시 서초구 태봉로 날씨입니다",
  "area": "서울특별시 서초구 태봉로",
  "forecast": "맑음",
  "forecastDetail": "맑음",
  "icon": 1,
  "temp": 21,
  "precipitationChance": 10,
  "windSpeed": 3,
  "weatherSun": {
    "forecast": "구름조금",
    "icon": 2,
    "tempHigh": 3.0,
    "tempLow": -4.0,
    "precipitationChance": 0,
    "windSpeed": 3.3,
  }
}
```

5. 이번주/다음주 날씨(Week Type : w8)의 infoDetail 설명

- weatherList[]는 이번주 날씨의 경우 8개(8일분=오늘, 내일, 이후 6일분), 다음주 날씨의 경우 6개(오늘, 다음 주 5일분) 으로 제공된다.

값	설명	대화서버 데이터명
type	"day", 날씨 정보 종류	totype
description	날짜, 일자, 지역 메시지	tomsg
area	지역	toarea
weatherList[]	일별 날씨 정보 배열	
weatherList[n].time	일자	titime1
weatherList[n].tempHigh	예상 최고온도 (°C)	titmx1
weatherList[n].tempLow	예상 최저온도 (°C)	titmn1
weatherList[n].icon	날씨 아이콘 코드 (1~22)	tiicon1
weatherList[n].forecast	날씨 예보 코드	tiwfk1
weatherList[n].precipitationChance	예상 강수확률 (%)	tipop1

○ 예시

```
{
```

```
"type": "week",
"description": "다음주 서울특별시 서초구 태봉로 날씨입니다",
"area": "서울특별시 서초구 태봉로",
"weatherList": [{
    "time": "오늘",
    "tempHigh": 25,
    "tempLow": 12,
    "icon": 2,
    "forecast": "구름조금",
}, {
    "time": "05/27 (월)",
    "tempHigh": 22,
    "tempLow": 17,
    "icon": 1,
    "forecast": "맑음",
}, {
    "time": "05/28 (화)",
    "tempHigh": 23,
    "tempLow": 14,
    "icon": 1,
    "forecast": "맑음",
}, {
    "time": "05/29 (수)",
    "tempHigh": 26,
    "tempLow": 15,
    "icon": 2,
    "forecast": "구름조금",
}, {
    "time": "05/30 (목)",
    "tempHigh": 28,
    "tempLow": 16,
    "icon": 2,
    "forecast": "구름많음"
}]
}
```