
GiGA Genie Inside Device SDK Developer Guide

GiGA Genie

2020.06

KT AI연구소

목차

1.1 문서 연혁	4
1.2 목적	9
1.3 적용범위	9
1.4 지원 버전	9
1.5 용어 정리	10
2. S/W Architecture	10
2.1 연동 구조	11
2.2 Device SDK Flow Overview	12
2.3 Service Agent의 Mic / Media 제어	12
2.4 연동 서버 정보	15
2.5 제공하는 미디어 콘텐츠 형식	16
3. 디바이스 등록	17
3.1 Client Key 발급 및 이용 절차	17
3.2 Device SDK Register	18
3.3 Device SDK Unregister	19
4. 음성인식 Flow	20
4.1 호출어 인식	20
4.2 음성명령 인식	21
5. 미디어 재생 Flow	24
5.1 Channel ID	24
5.2 재생 상태 이벤트	24
5.3 재생 타입	25
6. 부가 서비스 Flow	27

6.1 지니뮤직 서비스	27
6.2 타이머 서비스	28
6.3 콕콕 114 / 길찾기	29
6.4 단말 특화 서비스	29
7. 단말 하드웨어 이벤트 Flow	31
7.1 볼륨	31
7.2 버튼	32
7.3 블루투스	33
8. API 목록	34
8.1 Agent SDK API 목록	34
8.2 KWS SDK API 목록	35
9. Agent API 상세	36
9.1 Inside	36
9.2 agent_register	37
9.3 agent_init	40
9.4 agent_setCommandEventCallback	42
9.5 agent_onCommand	43
9.6 agent_onEvent	53
9.7 agent_updateMediaStatus	55
9.8 agent_sendText	56
9.9 agent_startVoice	58
9.10 agent_sendVoice	59
9.11 agent_stopVoice	62
9.12 agent_getTTS	62
9.13 agent_sendCommand	64

9.14 agent_serviceLogin	67
9.15 agent_serviceLoginStatus	69
9.16 agent_serviceLogout	71
9.17 agent_debugmode	72
9.18 agent_setServerInfo	73
9.19 agent_getVersion	74
9.20 agent_setLocation	75
9.21 agent_unregister	76
9.22 agent_reset	78
10. KWS API 상세	80
10.1 kws_init	80
10.2 kws_detect	81
10.3 kws_reset	82
10.4 kws_error	83
10.5 kws_getVersion	84
10.6 kws_getKeyword	84
10.7 kws_setKeyword	85
10.8 kws_setModelPath	86
11. 개발 참고 사항 및 이슈 사항	88
11.1 예외처리	88
11.2 FAQ	89
[첨부 1] metaInfo 상세설명	93
[첨부 2] 날씨 infoDetail 구조	103

1. 개요

1.1 문서 연혁

변경일	변경사항
2019.07.01	- 최초 (V1.0.0 가이드)
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 -> mesg 로 변경함 - 콘텐츠 상세 내용을 전달하는 metaInfo 에 대한 설명 추가
2019.10.30	- agent_onEvent 내 Event 추가 - 연동 서버 정보 추가
2019.11.15	- agent_onEvent 의 SERVER_ERROR 의 errorCode 추가
2019.12.09	- metaInfo 내 infoDetail 에 지역정보, 길안내 추가
2019.12.19	- agent_getTTS 시 TTS 최대 길이(1024) 추가
2020.01.28	- agent_setServerInfo 예제 IP, Port 정보 현행화
2020.02.05	- 미디어 재생 Flow, 부가서비스 Flow 추가 및 Agent API 상세 설명 순서 변경
2020.02.28	- kws_setModelPath 수정 (Android 호출어 ModelPath 설명 추가)
2020.03.31	- agent_onEvent 내 "GO_TO_STANDBY" Event 추가
2020.05.15	- 개정 (V1.1.0 가이드)

[V1.1.0 주요 변경 사항]

※ NOTICE : SDK v1.0.x를 이용하여 만들어진 Application에서 SDK v1.1.x를 이용하기 위해서는 아래 변경사항에 대해서 반드시 수정해야 합니다.

구분	V1.0.x	V1.1.0
agent_onCommand() 의 parameter 변경	agent_onCommand(string cmd) - cmd : string 으로 변환된 JSON 으로 actionType 을 포함하고 있음 예) cmd = { "actionType" : "media_stream", "payload" : { "cmdOpt" : { ... } } } 	agent_onCommand(string actionType , string payload) - actionType : 명령 유형 string - payload : 각 actionType 의 부가정보를 담은 JSON String 예) actionType = "play_media" payload = { "cmdOpt" : { ... } }
agent_onCommand() 의 actionType name 변경	actionType 변경 내역 (9.3 의 actionType List Table 참고)	
	media_stream media_url media_stream(contentType : wav) dialog_response media_url(comandType: Req_UPMD) hardware_control	play_media play_media media_data exec_dialogkit control_media control_hardware

구분	V1.0.x	V1.1.0
	timer_set webview_url	set_timer webview_url
음성인식 시작 (Mic ON 후 sendVoice)	agent_onEvent() callback 의 evt : VOICE_START 이벤트로 처리	VOICE_START 이벤트 삭제 agent_onCommand 의 actionType = start_voice 로 변경 예) pseudo-code agent_onCommand(actionType, payload) { if actionType == "start_voice" : Mic-ON & agent_sendVoice() }
음성인식 종료 (Mic OFF, 결과 Text 확인)	agent_onEvent() callback 의 evt : VOICE_STOP 이벤트로 처리	VOICE_STOP 이벤트 삭제 agent_onCommand 의 actionType = stop_voice 로 변경 예) pseudo-code agent_onCommand(actionType, payload) { if actionType == "stop_voice" : Mic-OFF }
TTS 재생 (Wav 파일로 재생할 경우)	Step1: agent_onCommand()에서 cmd 의 actionType = media_stream 으로 TTS Text 확인	Step1: agent_onCommand()에서 actionType = play_media 로 TTS Text 확인 (9.3.4)

구분	V1.0.x	V1.1.0
	Step2: agent_onCommand()에서 cmd 의 actionType = media_stream , contentType 이 wav 인 데이터 전달받아 재생	Step2: agent_onCommand()에서 actionType = media_data 로 전달되는 TTS wav 데이터 전달받아 재생 (9.3.5)
Media URL 재생	agent_onCommand()에서 cmd 의 actionType = media_url 로 전달되는 미디어 url 확인 & 재생	agent_onCommand()에서 actionType = play_media 로 전달되는 미디어 url 확인&재생 (9.3.4)
재생중인 Media 제어	agent_onCommand()에서 actionType = media_url , commandType = Req_UPMD 로 전달되는 cmdOpt 에서 미디어 제어정보 확인	agent_onCommand()에서 actionType = control_media 로 재생 중인 미디어 제어(9.3.6)
단말 특화 서비스(Dialog Kit)	agent_onCommand(string cmd)에서 cmd 내 actionType = dialog_resonse 일때 JSON 의 dialogResponse 정보를 사용하여 단말 특화 서비스 구현 예) cmd = { "actionType" : "dialog_response", "commandType" : "Req_EXEC", " dialogResponse " : { "Intent" : "lightTurnOff", "applInfo" : {"NE-LIGHT" : "전등"}, "__systemMesg" : "전등을 끄습니다"	agent_onCommand(string actionType, string palyload)에서 actionType = exec_dialogkit 일때 payload JSON 의 dssMsg 정보를 사용하여 단말 특화 서비스 구현 예) action_type = "exec_dialogkit" payload = { "cmdOpt" : { "execType" : "dialogHandler", " execOpt " : { "action" : "kt.action.dialog.custom", " dssMsg " : { "Intent" : "lightTurnOff",

구분	V1.0.x	V1.1.0
	<pre> }, } </pre>	<pre> "applInfo" : {"NE-LIGHT" : "전등"}, "__systemMesg" : "전등을 켜습니다" } } } } } </pre>
agent_getTTS() 사용방법 변경	void agent_getTTS(string msg) 후 agent_onCommand() callback 의 actionType = media_stream 을 통해 wav 데이터 받아서 재생	string agent_getTTS(string msg) 의 return 값(string)으로 base64 encoded string 전달 받음. Base64 decoding 하여 TTS wav 바로 재생
서비스 로그인 확인 (OAuth access_token 확인)	N/A	agent_serviceLoginStatus API 추가 (9.21)
서비스 로그아웃 (OAuth 인증정보 삭제)	N/A	agent_serviceLogout API 추가 (9.22)

1.2 목적

본 문서는 (주)케이티(이하 KT)의 GiGA Genie AI 서비스를 3rd Party Application(이하 Service Agent) 에서 이용할 수 있도록 제공하는 GiGA Genie Inside SDK 를 사용하기 위한 연동 방법에 대해 기술한다.

1.3 적용범위

- Service Agent 에서 Inside SDK 를 연동하는 방법을 설명한다.
- Service Agent 의 빌드 버전에 따라 일부 기능에 제약이 있을 수 있으며, 경우에 따라 별도 빌드된 Inside SDK 를 제공 가능하다.
- KT 서비스 제공 계획의 변경 및 추가에 따라 추후 보완 및 변경될 수 있다.

1.4 지원 버전

Device SDK 는 아래 버전으로 빌드 되었다.

- **Android: Android 4.4 Version 부터 지원**

minSDK Version = 19

targetSDK Version = 26

compileSdkVersion = 26

- **Linux: Ubuntu, CentOS, Raspbian, Yocto 지원**

OS = Ubuntu x86_64, CentOS x86_64, Raspbian armv7l, Yocto aarch64(G-INSIDE reference H/W)

Compiler = gcc 5.4.0

- **Windows: Windows 7 부터 지원 (64bit)**

OS = Windows 10 64bit

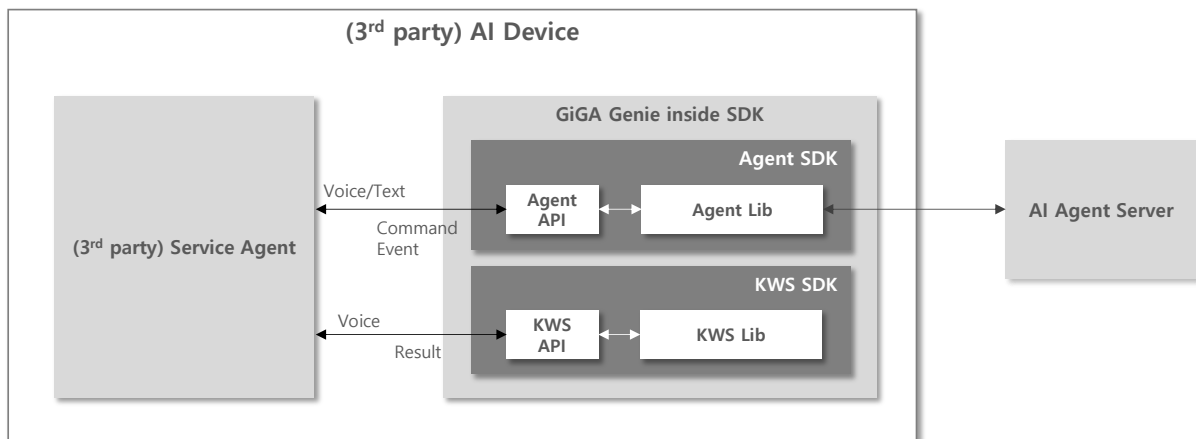
Compiler = Visual Studio 2017 15.9.12

1.5 용어 정리

- KWS: Keyword Spotting 의 약어로 호출어 인식을 의미
- TTS: Text to Speech 의 약어로 음성출력을 의미
- gRPC: gRPC Remote Procedure Call 의 약어로 원격 함수를 호출할 수 있는 프로토콜
- AI Agent Server : GiGA Genie AI 서비스를 제공하는 서버
- Service Agent : GiGA Genie Inside SDK 를 사용/구현한 3rd Party Application
- 단말 특화 서비스: 기가지니 INSIDE 적용 단말에서 제공하는 자체 서비스
- UUID(Unique User Identification) : AI Device 를 Unique 하게 구분하는 ID

2. S/W Architecture

GiGA Genie Inside SDK(이하 Device SDK) 구성 및 (3rd Party) Service Agent 와의 연동구조는 아래와 같다.



GiGA Genie Inside Device SDK 는 Agent SDK 와 KWS SDK 로 구성되어 있으며, 각 SDK 의 역할과 API 구성은 다음과 같다.

● Agent SDK

- AI Agent Server 연동(HTTPS, gRPC, WebSocket) 및 gRPC Connection 관리
- 음성(PCM)/텍스트 데이터를 AI Agent Server 로 전송하고 그에 대한 응답을 Service Agent 로 전달

● KWS SDK

- Service Agent 로 부터 음성 데이터(PCM)를 전달받고 설정된 호출어 ID 에 따른 호출어 인식 여부를 판단
- Service Agent 의 제어에 따라 호출어 인식 시작/종료 처리

2.1 연동 구조

2.1.1 Service Agent <-> Agent SDK

- Service Agent 는 Agent SDK 에서 제공하는 API 를 통해 음성/텍스트 등을 AI Agent Server 로 전송할 수 있다.
- Service Agent 는 callback 형태로 AI Agent Server 로부터 수신된 Response/Event/TTS 등을 Agent SDK 로 부터 전달 받을 수 있다.

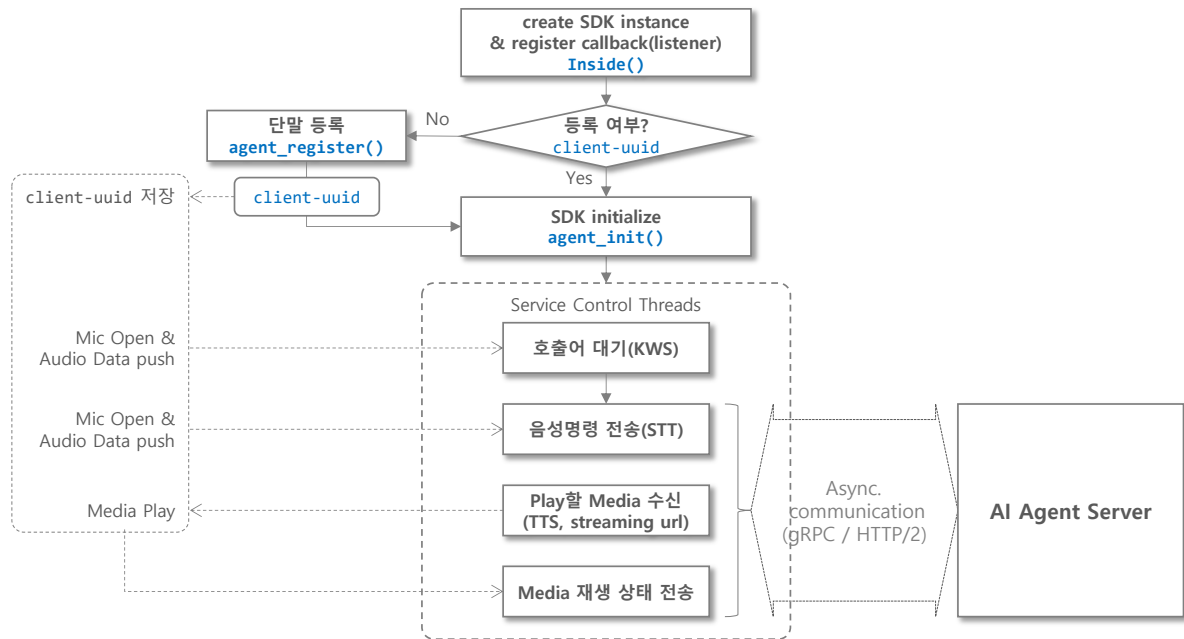
2.1.2 Agent SDK <-> AI Agent Server

- HTTP POST 기반 Register/Init/Unregister: client-id, client-key, client-secret, userid 정보를 AI Agent Server 로 전달하여 uuid 정보를 발급받거나 AI Agent Server 에 등록된 UUID 정보를 삭제할 수 있다.
- gRPC/WebSocket 프로토콜을 통해 음성/텍스트/TTS/이벤트/미디어 콘텐츠를 송수신 한다.
- 음성 인식을 위해 16KHz Signed 16bit Linear PCM 데이터를 Server 로 전달한다.

2.1.3 Service Agent <-> KWS SDK

- KWS SDK 를 사용하기 위해서는 Agent SDK 를 반드시 init 해야 한다.
- KWS SDK 에서 제공하는 API 를 통해 호출어 인식 시작/인식/중지 및 호출어 Id(기가지니, 지니야, 친구야, 자기야) 변경을 할 수 있다.
- 호출어 인식 여부는 detection API 의 리턴 값으로 판단할 수 있다.
- 호출어 인식을 위해 16KHz Signed 16bit Linear PCM 데이터가 KWS SDK 로 전달 되어야 한다.

2.2 Device SDK Flow Overview



- [Device SDK for Android]

SDK 객체를 생성하면서 InsideListener 콜백함수를 등록한다.(9.1 Inside)
단말 정보를 등록하여 UUID 를 발급 받은 후 (9.2 agent_register)
Service Agent 내 UUID 를 저장하고 AI Agent Server 와 연결한다.(9.3 agent_init)

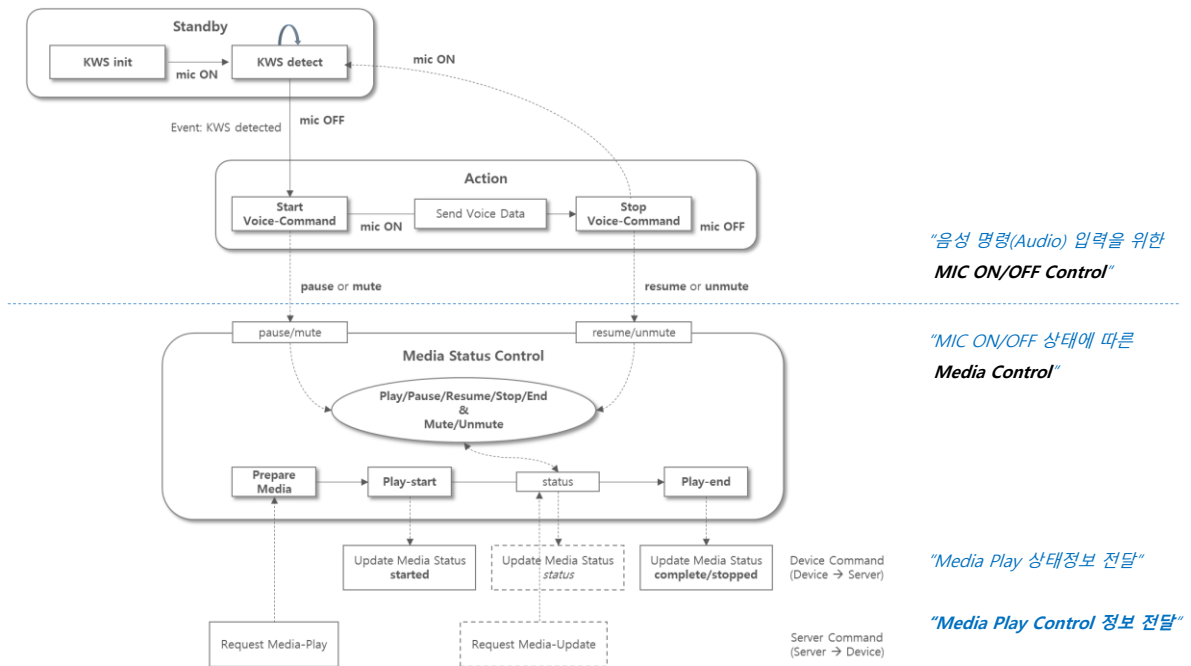
- [Device SDK for Linux/Windows]

단말 정보를 등록하여 UUID 를 발급 받은 후 (9.2 agent_register)
Service Agent 내 UUID 를 저장하고 AI Agent Server 와 연결한다.(9.3 agent_init)
서버 이벤트를 수신하기 위한 콜백 함수를 등록 한다.(9.4 agent_setCommandEventCallback)

2.3 Service Agent의 Mic / Media 제어

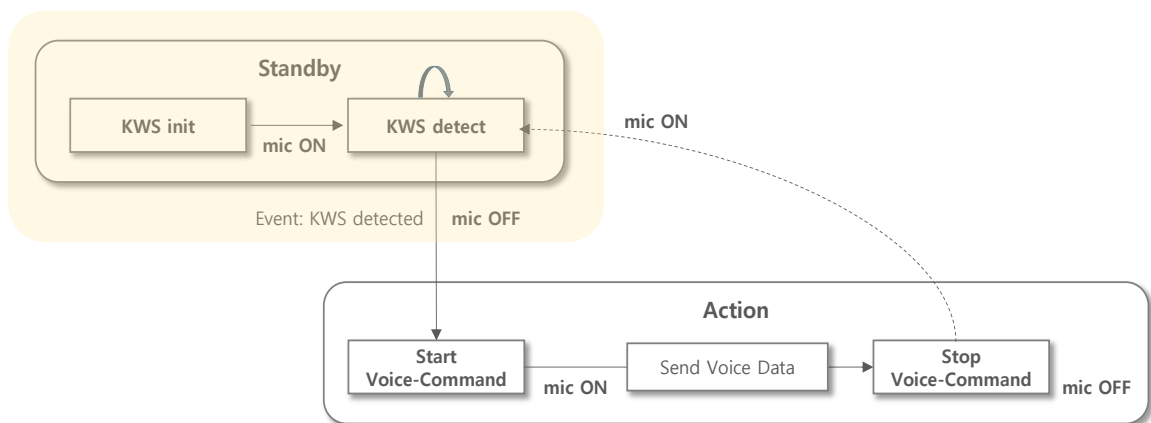
Service Agent 는 "음성명령(Audio) 입력을 위한 MIC ON/OFF 제어"와 "MIC ON/OFF 상태에 따른 Media 제어"를 해야 한다.

■ Service Status (Overview)



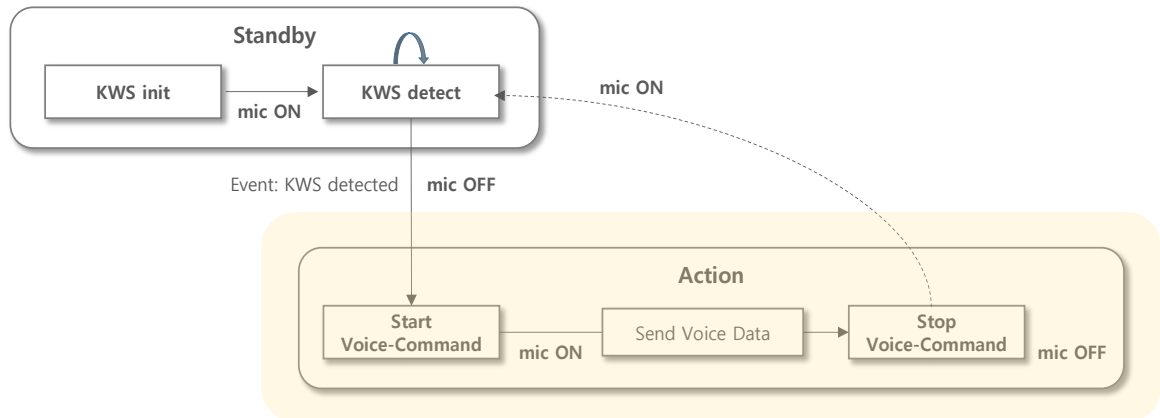
■ Standby 상태

- 사용자 Event 대기 상태
- 호출어 인식 또는 특정 Button Push Event Trigger 를 기다리는 상태



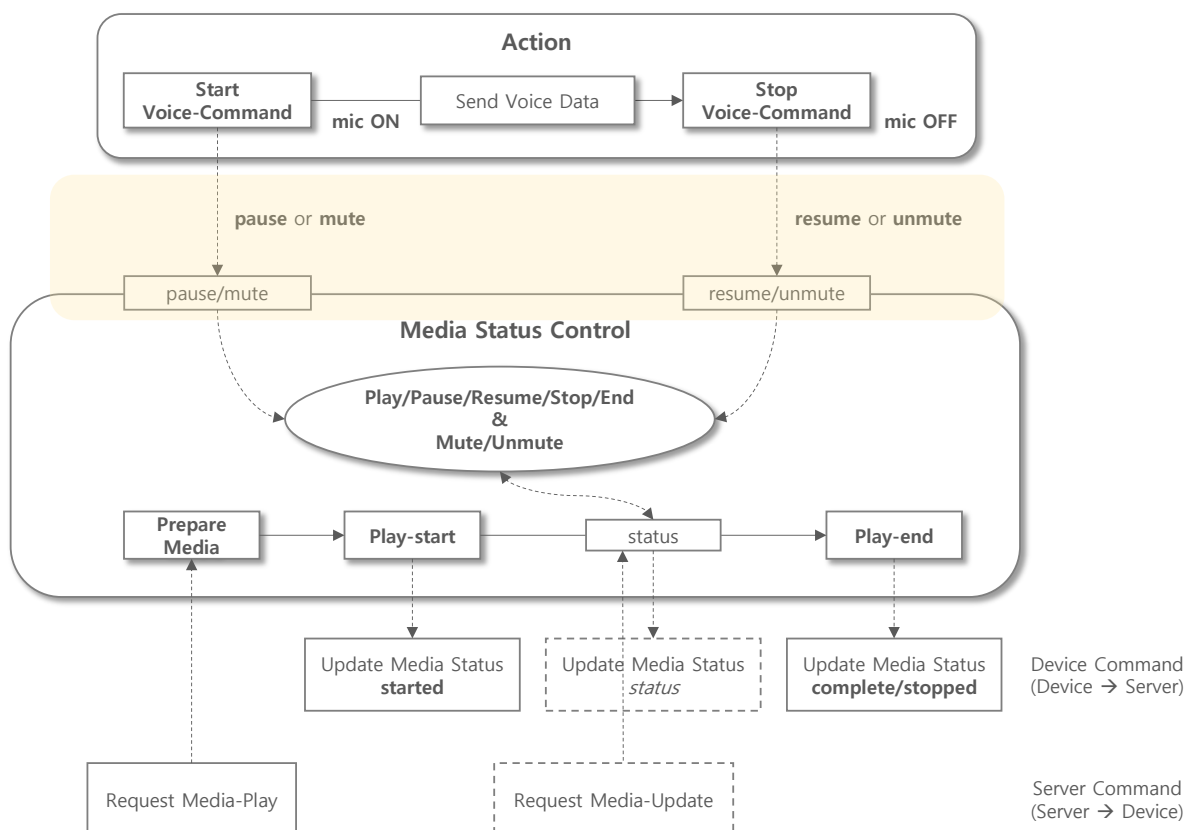
■ Action 상태

- 사용자 음성명령어(ex. TEXT 명령어) 처리하는 상태
- 음성명령어 입력을 위해 MIC ON, 음성명령어 입력 완료 시 MIC OFF 필요



■ Action 에서 Media Status Control 필요

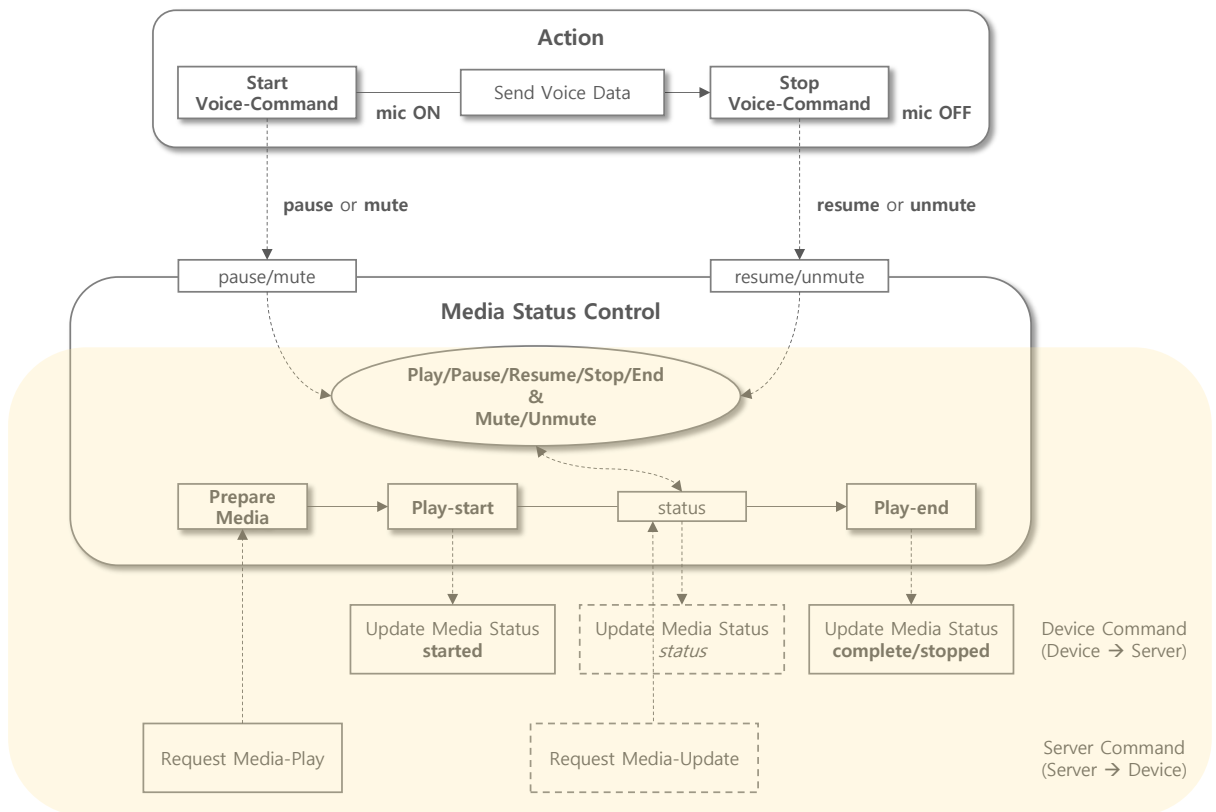
- 미디어 재생 중에도 사용자 음성명령어 처리가 가능해야 하며 이 때 미디어 서비스에 대해 명령어 인식 전 "pause/mute", 명령어 종료 후 "resume/unmute" 를 해야 한다.



■ Media State Control 상태

- 서버에서 미디어 재생 요청 명령이 전달되면 미디어 재생

- 단말에서 미디어 재생 시작/종료/중지 시 서버로 미디어 상태 정보를 전송
- 단말에서 미디어 재생 중 playNotiTime 주기로 미디어 상태 정보를 전송
- 서버에서 미디어 서비스 제어 요청 (pause/stop/resume/seek) 명령이 전달 되면 해당 Action 을 수행하고 미디어 제어 결과 상태 정보를 전송



2.4 연동 서버 정보

기본적으로 상용 서버와 연동하게 되어 있으며, setServerInfo API 로 연동 서버 주소를 변경할 수 있다. 연동 서버 주소는 아래와 같다.

- Host Address: inside-dev.gigagenie.ai
- gRPC Port: 50109 , Rest Port: 30109

상용서버 정보는 다음과 같다.

- Host Address: inside.gigagenie.ai
- gRPC Port: 50051, Rest Port: 9080

2.5 제공하는 미디어 콘텐츠 형식

GiGA Genie Inside SDK 를 적용하는 3rd Party 디바이스는 다음과 같은 코덱 지원이 필요하다.

종류	지원 리스트
오디오	MPEG1 L1/2/3, AAC, WMA, Vorbis, FLAC
컨테이너	MP4, WMV, MKV, MPEG2-TS, ASF, 3GP
스트리밍	HTTP, HLS

서비스 별 제공하는 오디오 파일 형식은 다음과 같습니다.

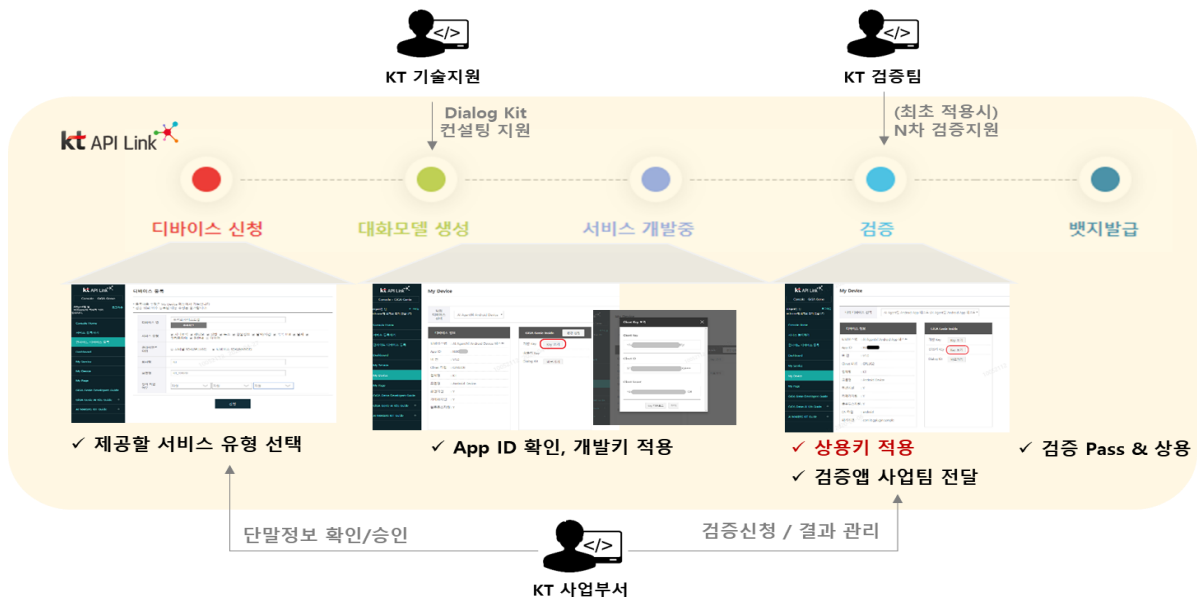
- 뉴스 : .mp3, .wav, .mp4
- 라디오 : .m3u8
- 지니뮤직 / 팟캐스트: mp3

3. 디바이스 등록

Device SDK 를 이용하는 Service Agent 는 API 를 사용하기 위해 Client 정보 입력을 통해 디바이스를 등록 해야 한다. 디바이스 등록을 위한 절차는 아래에 설명한다.

3.1 Client Key 발급 및 이용 절차

Device SDK 사용을 위해서는 SDK 등록 시(9.2 agent_register) 인증에 필요한 Client 정보를 입력해야 한다. Client 정보는 아래 절차에 따라 발급 받은 후 사용한다.



1. API Link(<https://apilink.kt.co.kr>) 에서 회원가입
2. 사업 제휴 신청 및 디바이스 등록 (Console > GiGA Genie > 인사이드 디바이스 등록)
3. 디바이스 등록 완료 후 My Device 에서 등록한 디바이스 정보 및 개발키 발급 확인 (Console > GiGA Genie > My Device)

- client-id: 클라이언트 아이디로, 단말당 1 개의 개발/상용 client_id 제공
- client-key: 클라이언트 키로, 단말당 개발/상용 client_key 제공
- client-secret: 클라이언트 시크릿으로, 단말당 개발/상용 client_secret 제공

4. 발급된 개발키로 개발 서버 연동 및 테스트

- setServerInfo API 로 개발서버 연동(개발서버 host/port 는 변경될 수 있음)
- 발급된 개발키 (client_id, client_key, client_secret)로 개발 테스트

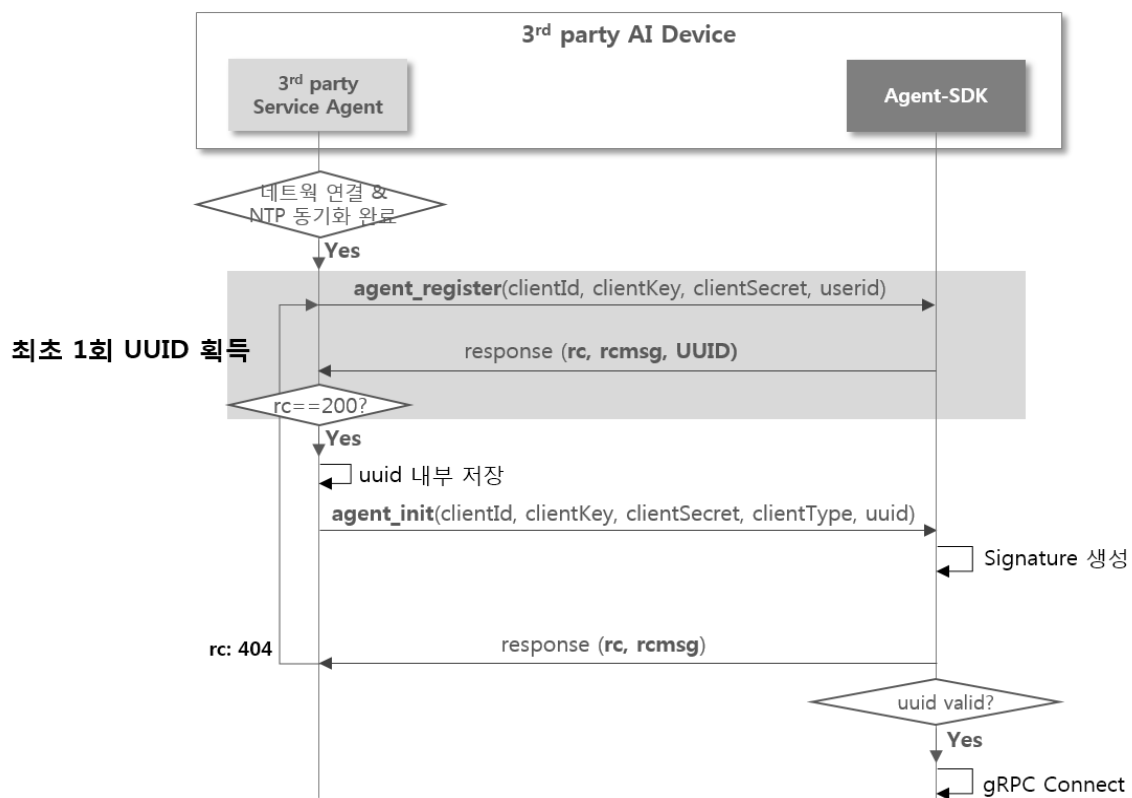
5. 개발 완료 후 상용 검증 신청 (Console > GiGA Genie > My Device)

6. 발급된 상용키로 상용 서버 연동 및 테스트

- setServerInfo 를 하지 않으면 default 로 상용서버로 연동
- 발급된 상용키 (client_id, client_key, client_secret)로 상용 적용

3.2 Device SDK Register

단말등록 및 UUID 발급/저장, Agent Server 와 gRPC 연결 절차는 다음과 같다.



Register 절차

- 1) 단말 등록을 위해 `agent_register` API 호출이 필요하다. 이때 결과 값으로 단말별 Unique ID(UUID)가 전달된다. 발급받은 UUID는 Service Agent에서

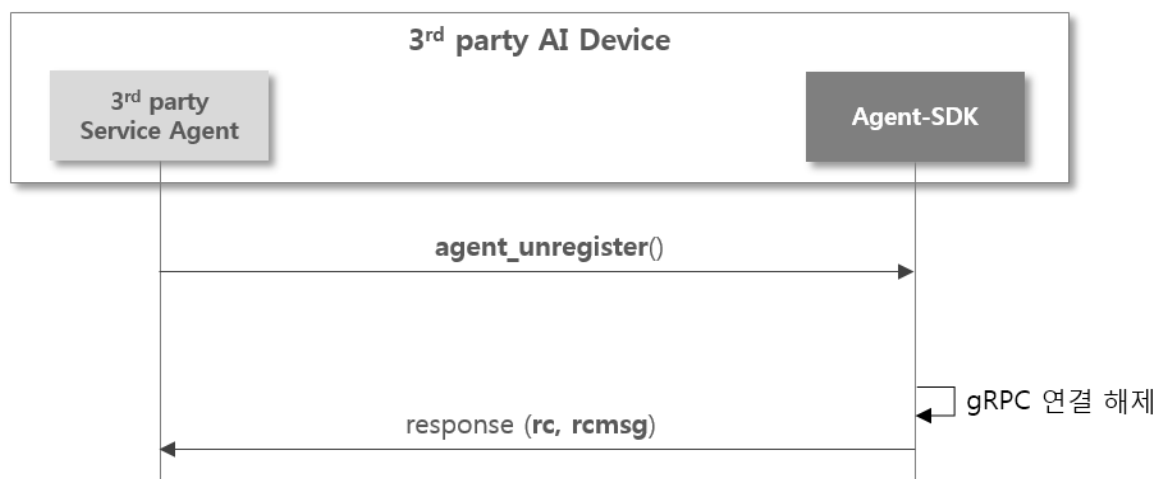
저장/관리한다. (9.2 agent_register 참고) 이전에 발급받은 UUID가 있는 경우 agent_register 절차는 생략 가능하다.

2) 단말 등록 후에는 agent_init API 호출을 통해 Agent Server와 연결한다. (9.3 agent_init 참고)

3) UUID는 필요에 따라 동일 UUID로 다른 Device로 접속하는 형태도 가능하나 동시에 여러 단말에서 접속할 수 없다. (Agent Server에서는 서비스 단말의 성격과 사업 제휴에 따라 발급 가능한 UUID 개수를 제한할 수 있다.)

3.3 Device SDK Unregister

Agent Server에 등록된 단말을 해제하기 위한 절차는 다음과 같다.



Unregister 절차

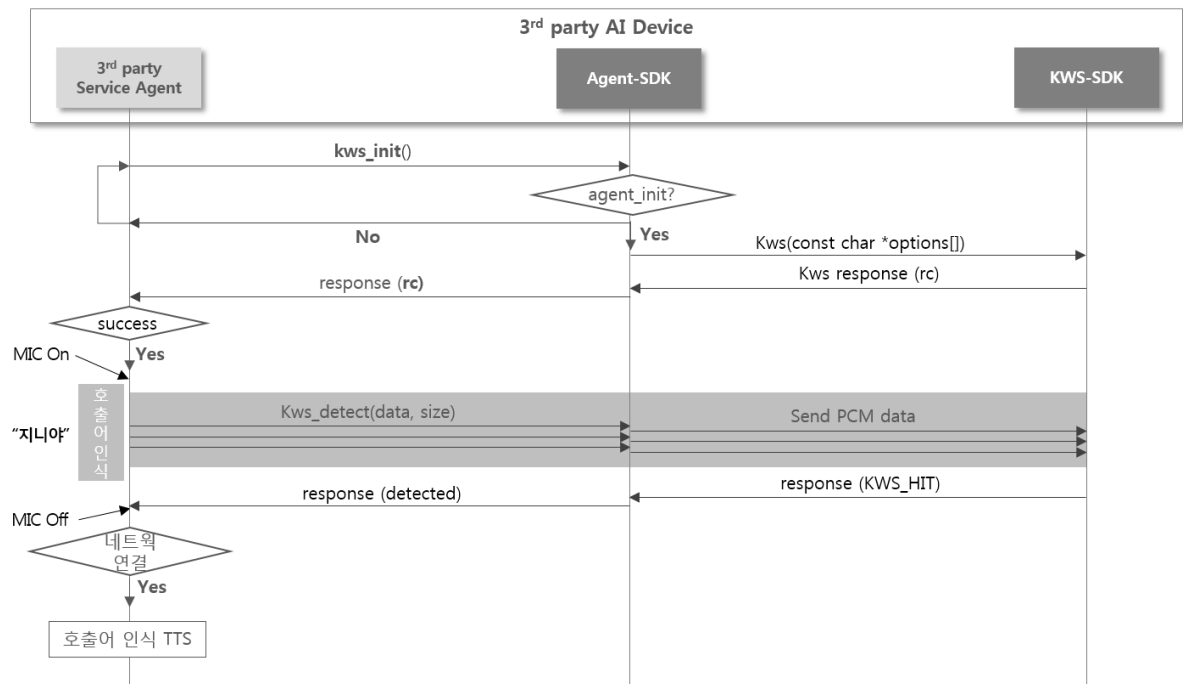
1) 디바이스가 초기화 되거나 단말의 사용자 등록 정보가 초기화 되는 경우 Service Agent는 agent_unregister API를 통해 서버에 등록된 정보(UUID)를 삭제해야 한다. (9.21 agent_unregister 참고)

2) UUID 정보가 삭제되면 SDK는 서버 연결을 해제한다. Service Agent는 서버 연결 해제 후 단말에서 관리하던 사용 정보/이력이 있으면 모두 삭제한다

4. 음성인식 Flow

4.1 호출어 인식

호출어 인식 절차는 아래와 같다.

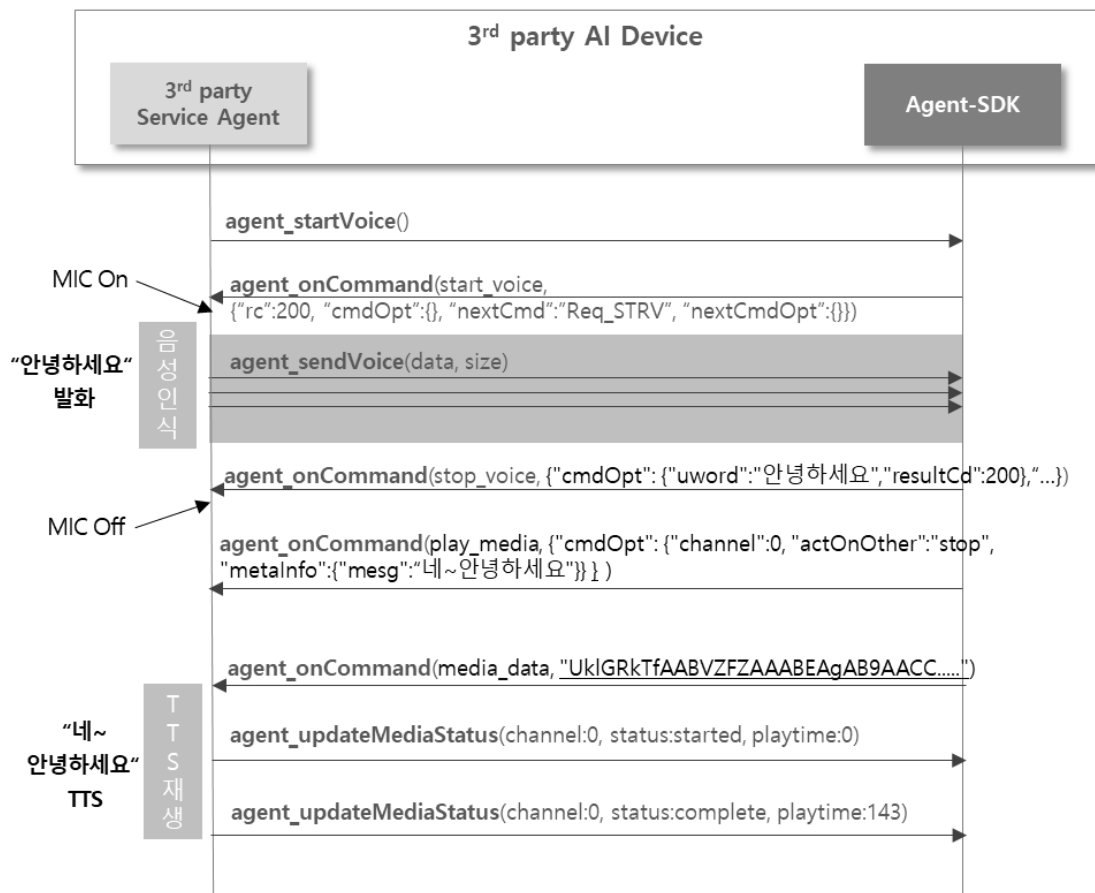


- 1) Service Agent 는 호출어 인식을 위해 KWS SDK 초기화를 요청한다. 만약 Agent SDK 가 초기화 되어 있지 않다면 Agent SDK 초기화 후 KWS SDK 초기화를 요청해야 한다. (10.1 kws_init 참고)
- 2) KWS SDK 초기화가 완료된 경우 호출어 인식을 위해 마이크를 open 하고 마이크에서 입력된 데이터를 전달한다.(10.2 kws_detect 참고)
- 3) 호출어가 인식되면 Service Agent 는 호출어 인식을 위한 음성 데이터 전송을 중지하고 마이크를 close 한다.
- 4) Service Agent 에서의 호출어 인식 시작/종료 시점은 다음과 같다.
 - 디바이스 부팅 완료 후 호출어 인식 시작
 - 호출어가 검출된(detected) 경우/ 디바이스 마이크 버튼 입력 시 호출어 인식 종료

- 음성인식 완료 수신/ TTS 또는 미디어 재생 시작 시 호출어 인식 시작
- 네트워크가 연결되지 않거나 정상적으로 register 가 안된 상태에서는 예러 TTS 재생 시작 후 호출어 인식 시작

4.2 음성명령 인식

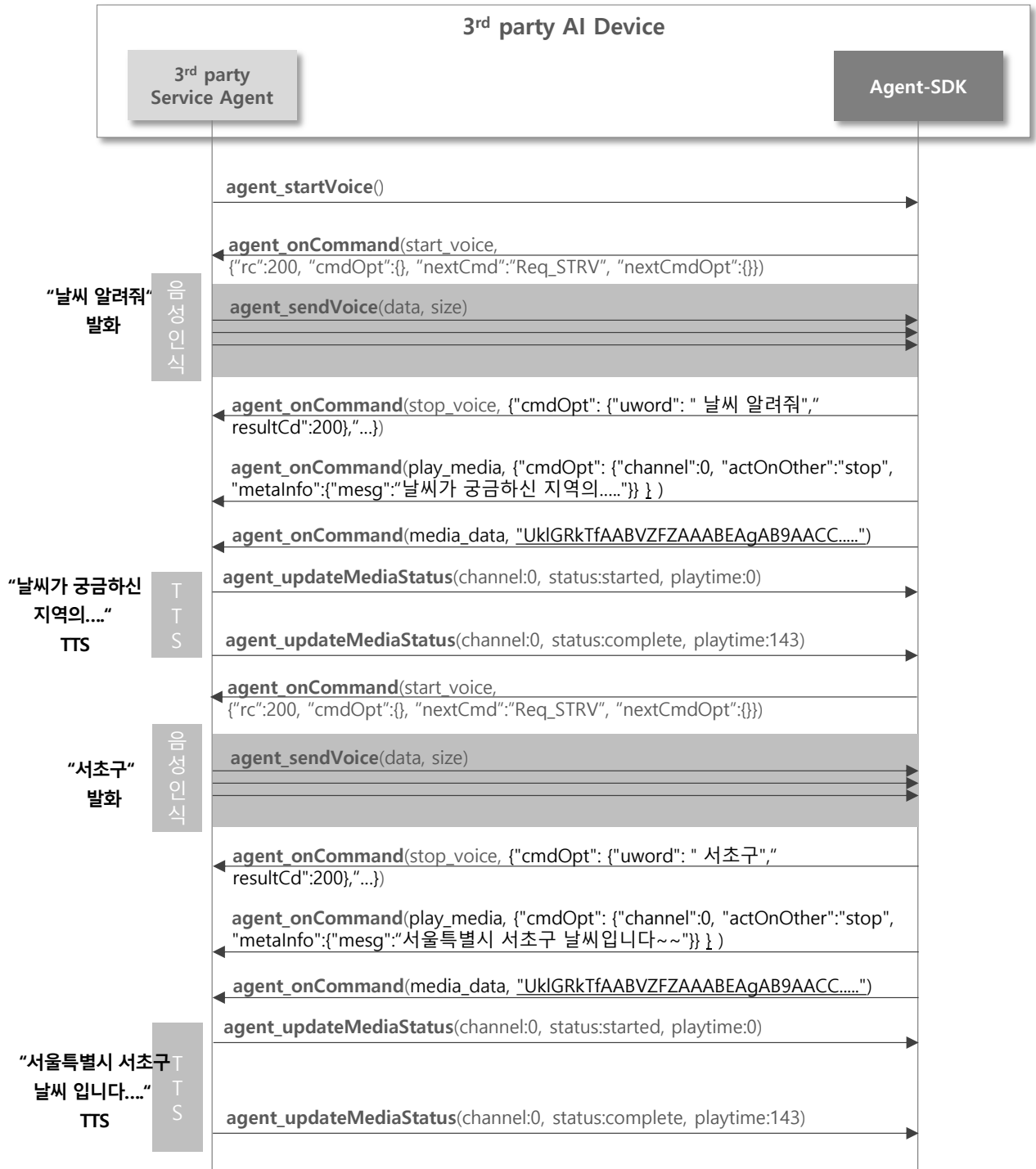
음성명령 인식 절차는 아래와 같다.



- 1) 호출어 인식 후 또는 단말의 마이크 버튼을 눌러 음성인식이 시작되는 경우 Service Agent 는 agent_startVoiceAPI 를 통해 서버로 음성인식 시작을 요청한다. (9.9 agent_startVoice 참고)
- 2) 서버로부터 agent_onCommand 의 actionType : start_voice 를 수신하면 Service Agent 는 마이크 Open 후 agent_sendVoice API 를 통해 음성 데이터를 전송한다. (9.5 agent_onCommand, 9.9 agent_sendVoice 참고)

- 3) 서버로부터 agent_onCommand 의 actionType : stop_voice 를 수신하면 Service Agent 는 음성 데이터 전송을 중단하고 다음 서버 메시지를 기다린다. . (9.5 agent_onCommand 참고)
- 4) Service Agent 는 TTS/미디어 재생 요청, 볼륨 제어 요청 등의 명령어가 수신되면 해당 요청에 대한 처리를 시작하고 호출어 인식 또는 특정 Button 이벤트를 대기하는 Standby 상태가 된다. (2.3 Service Agent 의 Mic / Media 제어 참고)

연속으로 음성명령을 발화하는 절차는 다음과 같다.



5. 미디어 재생 Flow

음성발화에 대한 응답으로 TTS(media_data:wav)나 음악, 라디오 등 미디어 재생을 위한 URL(media_url)이 전달될 수 있고 Service Agent 는 각 케이스에 따라 미디어를 재생해야 한다.

5.1 Channel ID

서버는 음성발화에 대한 응답을 보낼 때 미디어 서비스를 재생해야 하는 channel 정보와 함께 전송한다. 단말에서는 미디어 재생을 위해서 channel 정보로 미디어 재생을 제어할 수 있다.

- 재생되는 미디어를 구분하기 위해 Channel ID 가 사용되며 미디어 재생 요청 시 재생하고자 하는 미디어의 Channel ID 는 agent_onCommand 의 actionType : play_media 를 통해 전달된다. (9.5 agent_onCommand 참고)
- TTS 의 경우 0~9 번을 사용하고 그 외 미디어는 101~999 사이의 값을 사용한다.
- 전달 받은 Channel 정보로 미디어 재생상태를 업데이트 해야 한다. (9.7 agent_updateMediaStatus 참고).
- TTS 를 제외한 미디어는 동시 재생이 가능해야 한다.
 - Channel 101 번 재생 중 Channel 101 번에 대한 제어 명령없이 Channel 102 번 재생요청이 전달되면 Channel 101 번과 Channel 102 번 미디어를 동시 재생

5.2 재생 상태 이벤트

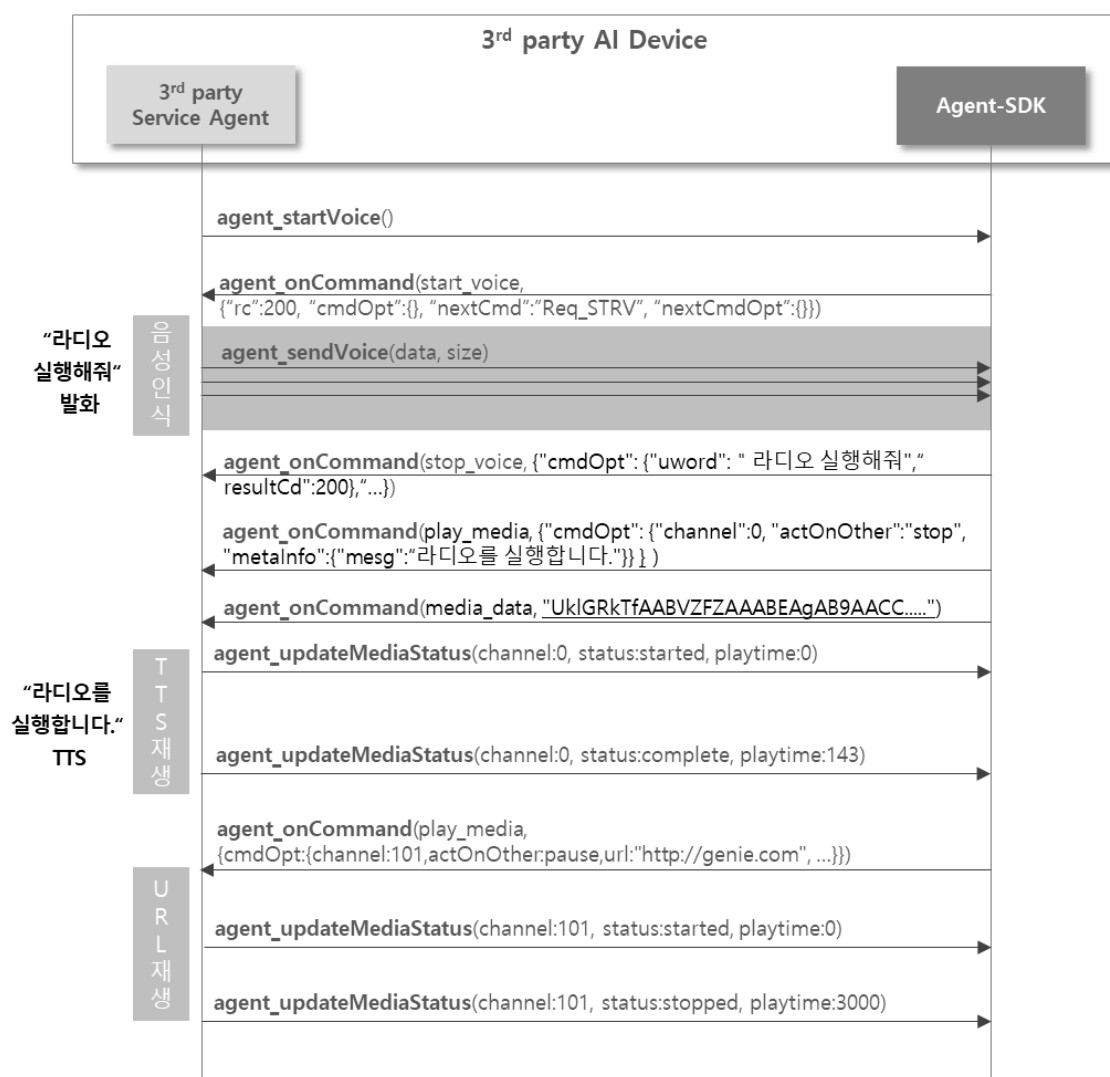
- Service Agent 는 미디어 재생요청을 받아 재생하는 경우 재생상태에 따라 "started", "paused", "stopped", "complete" 이벤트를 channel ID, 총 재생시간 정보와 함께 미디어 상태 이벤트를 전송해야 한다.

- 미디어 재생상태 이벤트가 전달되지 않는 경우 이후 서비스 진행에 문제가 발생할 수 있다.

5.3 재생 타입

5.3.1 media URL

■ mediaURL Flow

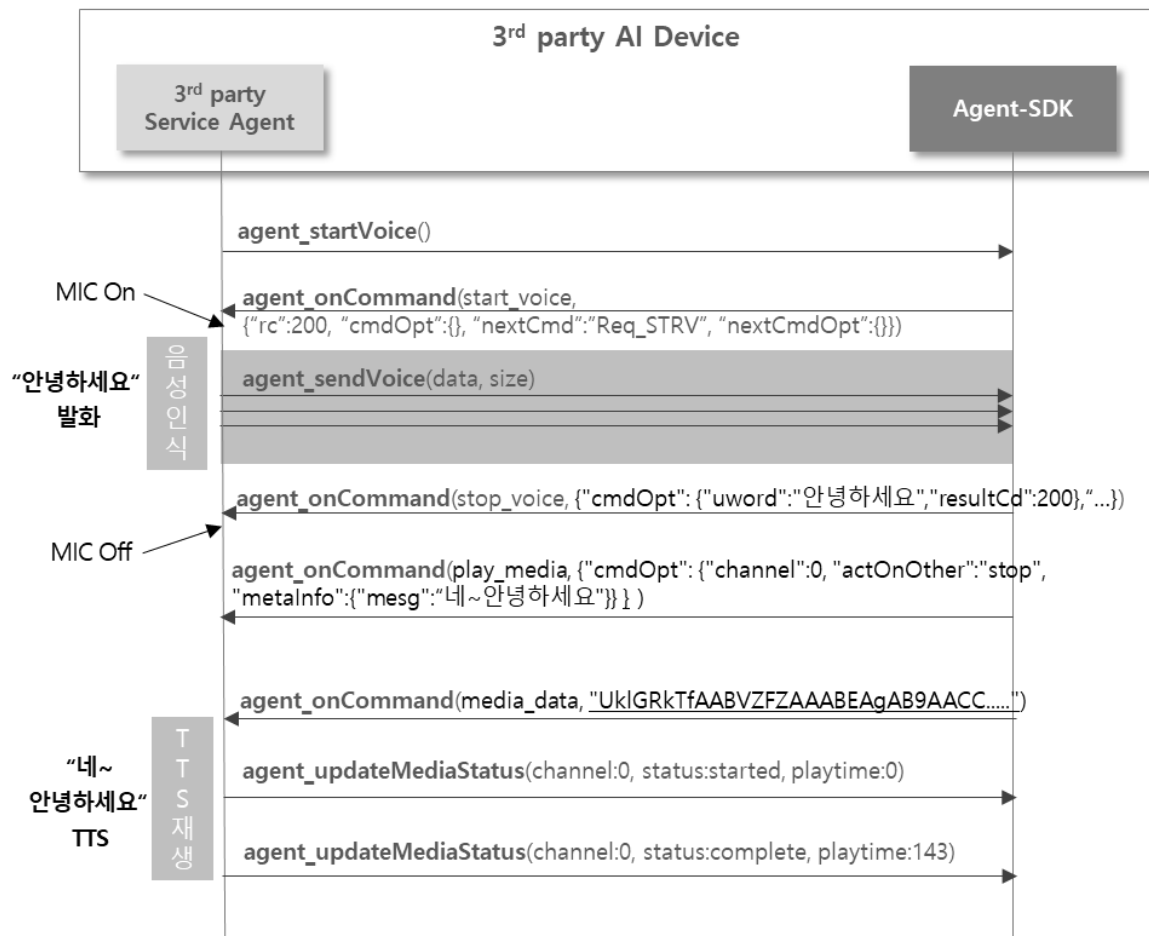


- 미디어 서비스의 경우 서버는 재생할 미디어 URL 정보를 전달하고 Service Agent 는 전달받은 URL 을 재생해야 한다.
- agent_onCommand 의 actionType : play_media(9.5.3) 규격을 참고한다.

- 해당 서비스 : 라디오, 팟캐스트, 지니뮤직, 뉴스 서비스

5.3.2 media Stream

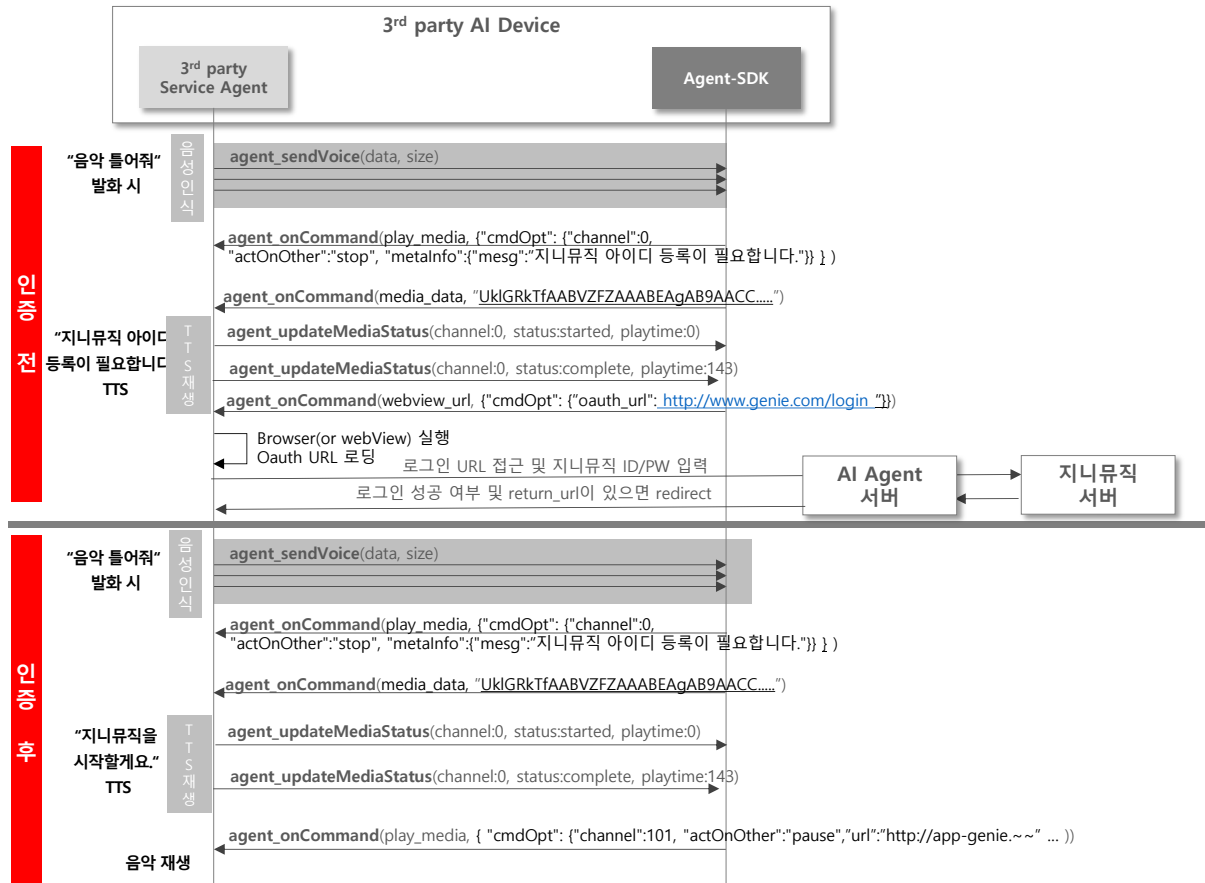
■ wav 파일 전송 Flow



- 서버에서 TTS 재생 contentType 이 wav 로 전달될 경우 TTS 메시지 정보와 다른 출력 미디어에 대한 action 정보가 먼저 내려오고 실제 mediastream 정보가 이어서 내려온다.
- agent_onCommand 의 actionType : play_media(9.5.3) 와 actionType : media_data(9.5.4) 규격을 참고한다.

6. 부가 서비스 Flow

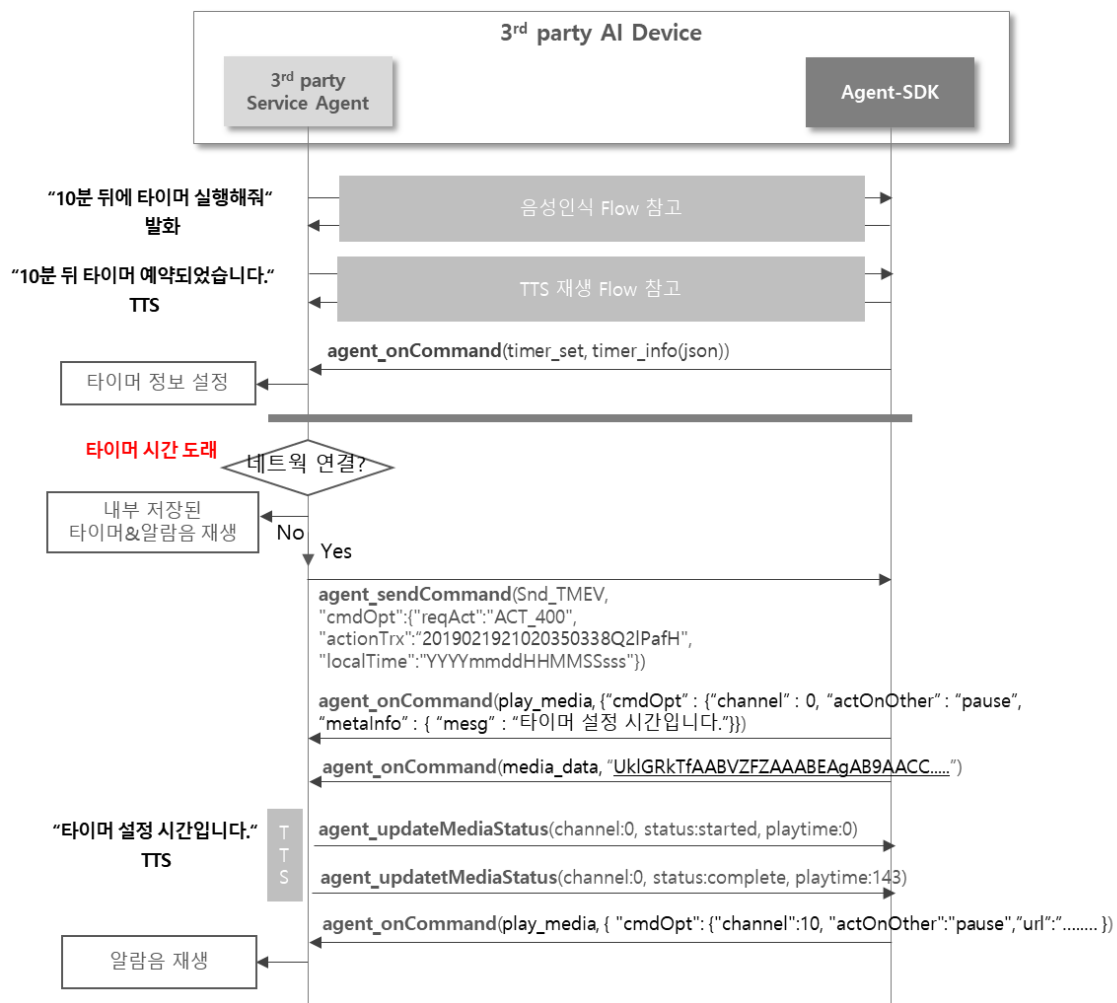
6.1 지니뮤직 서비스



- 지니뮤직 서비스의 경우 해당 서비스에서 발급받은 계정으로 로그인(인증)이 필요하다.
- 이 경우 서비스 실행 요청 시 인증을 위한 웹 페이지 URL 이 전달되며 Service Agent 는 해당 웹 페이지를 사용자에게 제공하여 인증(로그인)이 가능하도록 해야 한다. (참고 : AI Agent Server 는 해당 서비스의 OAuth 인증 방식을 지원하며 회원가입은 지니뮤직 Web/App 에서 진행한다.)
- 서비스 인증이 완료된 경우 서비스 시작을 알리는 TTS 가 전달되고 Service Agent 는 TTS 를 재생한다. (5.3.2 Wav 파일 재생 Flow 참고)
- TTS 재생완료 정보가 AI Agent Server 로 전달되면 음악재생을 위한 미디어 URL 이 전달된다. (9.5.3 agent_onCommand 의 actionType : play_media 참고)

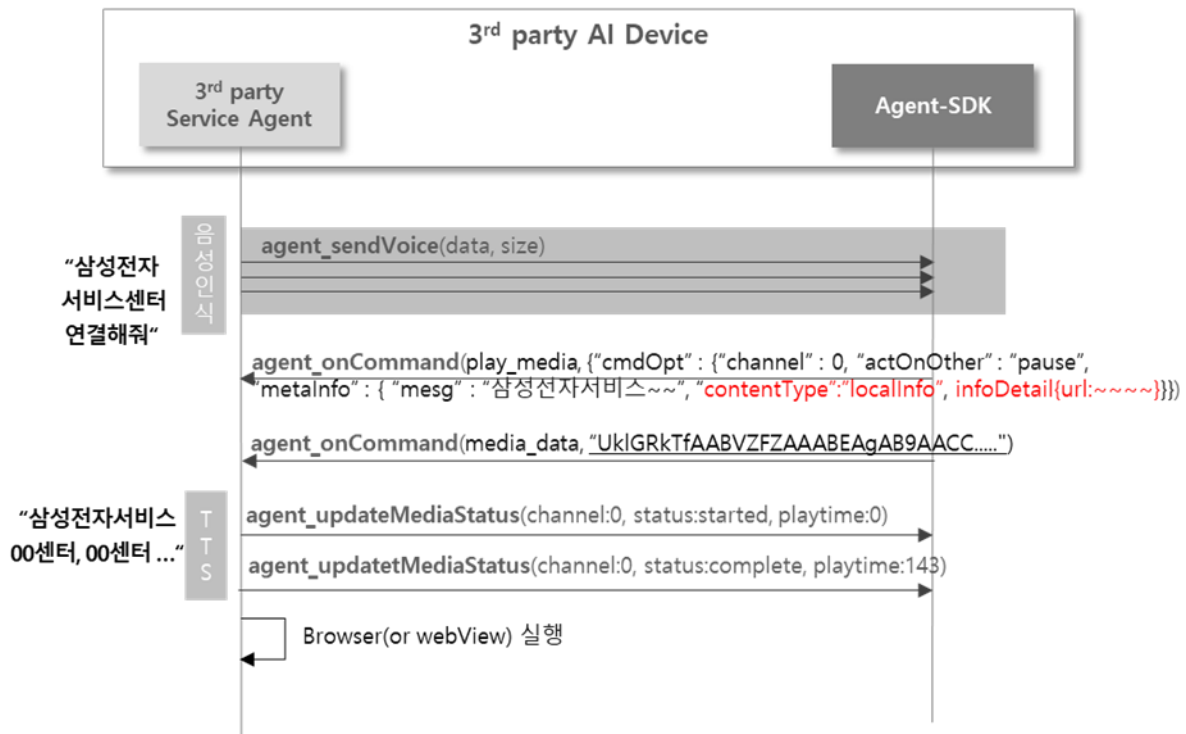
- Service Agent 는 전달된 URL 로 미디어를 재생하며 playNotiTime 으로 전달된 주기에 따라 미디어 재생 상태정보를 반드시 전달해야 한다.
- agent_updateMediaStatus(9.5.7) 규격을 참고한다

6.2 타이머 서비스



- 타이머의 경우 설정 시간(setTime)이 절대시간이 아닌 초단위의 상대시간으로 전달된다. (9.5.8 agent_onCommand 의 actionType : set_timer 참고)
- 전달받은 setTime 정보로 Service Agent 에서 디바이스 시간 기준으로 계산하여 저장/관리한다.
- 설정된 시간이 경과한 후 reqAct, actionTrx, localTime(디바이스 시간) 정보를 포함한 타이머 이벤트를 정보를 전달 한다.(9.13 agent_sendCommand 참고)
- 서버에서 타이머 설정시간 안내 메시지가 전달되면 Service Agent 에서는 안내 TTS 재생 후 내부 저장된 알람음을 재생한다.

6.3 콕콕 114 / 길찾기



- 콕콕 114/길안내 서비스의 경우 요청된 정보가 `agent_onCommand`의 `actionType:play_media`의 payload로 전달된다.
- 콕콕 114의 `contentType`은 `localInfo`, 길찾기의 `contentType`은 `directions`으로 전달되며 상세 정보는 `infoDetail { url : ~~ }`로 전달된다.
- 화면이 있고 UI를 제공하는 단말의 경우 전달된 URL을 `webView`로 보여준다.
- `agent_onCommand(9.5.3, 9.5.4)` 규격, [첨부 2] 서비스 별 `infoDetail` 정보를 참고한다.

6.4 단말 특화 서비스

[사전 작업]

- 단말 특화 서비스는 3rd Party 단말에서 제공하는 서비스가 동작하기 위한 것으로 해당 서비스를 동작시키기 위한 발화문구/정보/답변문구는 개발자 센터(API Link)의 `dialogkit`을 통해 미리 등록해야 한다.

dialogkit 이용 가이드: <https://github.com/gigagenieDmt/DialogKit-deploymentGuide/wiki>



- dialogkit 에 등록된 정보가 agent_onCommand 의 actionType:exec_dialogkit 으로 전달되면 Service Agent 는 intent, appInfo 정보를 기준으로 단말 특화 서비스를 실행시킨다. __systemMesg 로 내려오는 답변 정보로 TTS 재생 또는 (화면이 있는 단말의 경우) UI 로 표시 가능하다.
- agent_onCommand 의 actionType : exec_dialogkit (9.5.3) 규격을 참고한다.

7. 단말 하드웨어 이벤트 Flow

7.1 볼륨



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

- 단말에서 볼륨이 변경되면 AI Agent Server 로 변경된 볼륨 값을 전송한다.
`agent_sendCommand(9.13)` 규격을 참고한다.
- 사용자가 음성발화를 통해 볼륨을 조절하면 서버는 `agent_onCommand` 로 볼륨에 대한 이벤트를 전달한다.
- `agent_onCommand` 의 `actionType : control_hardware(9.5.7)` 규격을 참고한다.

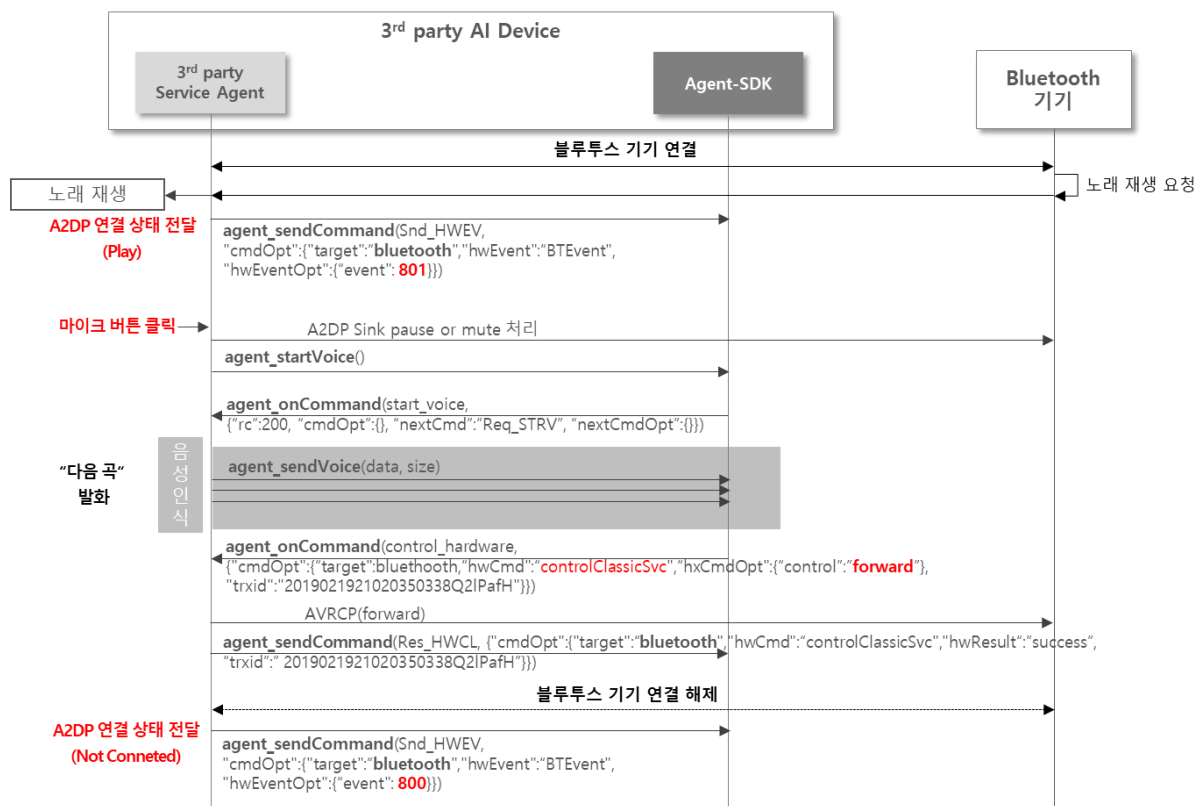
7.2 버튼



사용자가 물리적 버튼을 통해 미디어 제어(play/pause/forward/backward) 하는 경우 아래와 같이 처리한다.

- 단말에서 미디어 제어 HW 이벤트가 발생하면 AI Agent Server 로 HW 이벤트 정보를 전송한다. `agent_sendCommand`(9.13) 규격을 참고한다.
- AI Agent 서버에서는 현재 재생 중인 미디어에 따라 제어 명령을 하거나 이전곡/다음곡에 해당하는 음원 URL 을 전달 할 수 있다. `agent_onCommand` `actionType: control_media` (9.5.6) 규격을 참고한다.

7.3 블루투스



단말에서 블루투스 연결/해제 이벤트가 발생하면 아래와 같이 처리한다.

- Service Agent 는 A2DP 연결(801)/해제(800) 상태를 서버에 전달한다
agent_onCommand actionType: control_media (9.5.6) 규격을 참고
- A2DP 연결 및 음악 재생 중 음성으로 곡 제어(다음곡, 이전곡 등)를 하면 서버로 부터 control_hardware Command 가 전달된다
- agent_onCommand 의 actionType : control_hardware(9.5.7) 규격을 참고한다.

8. API 목록

8.1 Agent SDK API 목록

	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
void	agent_setCommandEventCallback (agent_on Command aoc, agent_onEvent aoe)	Agent SDK 콜백 함수 등록 (Linux, Window)
void	agent_onCommand (StringactionType, string payload)	서버로부터 전달받은 음성 인식 해석 결과 콜백
void	agent_onEvent (int evt, String opt)	서버로부터 전달받은 이벤트
void	agent_updateMediaStatus (int ch, String state, int playtime)	재생중인 미디어 상태 전송
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
String	agent_getTTS (String msg)	입력된 텍스트 TTS 변환 요청에 대한 결과 값과 TTS 스트림 전달
void	agent_sendCommand (String cmd)	서버로 전달할 요청/이벤트
String	agent_serviceLogin (String service_type, String return_url)	OAuth 인증을 통해 서비스 로그인을 수행할 수 있는 페이지정보를 요청

	API	설명
String	agent_serviceLoginStatus (String service_type)	서비스 로그인 상태 정보 요청
String	agent_serviceLogout (String service_type)	서비스 로그아웃 요청
void	agent_debugmode (bool enable)	디버그 모드 (로그 보기)
void	agent_setServerInfo (String serverIP, String grpcPort, String restPort)	연동서버 정보를 설정
String	agent_getVersion ()	Agent SDK 버전 정보 조회
void	agent_setLocation (String Longitude, String Latitude, String Address)	단말 위치 정보 입력
String	agent_unregister ()	UUID 삭제 요청
void	agent_reset ()	클라이언트 초기화

8.2 KWS SDK API 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 kwslid)	호출어 변경
int	kws_setModelPath (String path)	호출어 모델 경로 설정

9. Agent API 상세

9.1 Inside

Service Agent 가 Agent Server 와 연동하기 위해서 객체를 생성한다. Agent SDK 를 사용하기 전에 반드시 호출되어야 하며 **Android** 에서만 제공된다.

■ API

Inside(Context context, InsideListener listener)

■ Parameter

이름	유형	설명
context	Context	-
listener	InsideListener	Service Agent가 SDK 로부터 Notification이나 이벤트 등을 수신하기 위한 callback

- Service Agent 는 Notification 이나 이벤트 등을 수신하기 위해 callback 을 아래와 같이 등록한다.
- agent_onCommand(string actionType, string payload): 9.5 agent_onCommand 참고
- agent_onEvent(int evt, string opt): 9.6 agent_onEvent 참고

■ 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_register

GiGA Genie Inside 를 사용하기 위해서 단말을 등록하고 UUID 를 발급받는다. 이때 단말(사용자)을 식별하기 위한 정보로 user_id 를 입력하게 되며 단말의 MAC 주소 등 해당 단말에서 변경되지 않는 정보를 사용해야 한다.

발급받은 UUID 는 Service Agent 에 내부적으로 저장하고 있다가 agent_init() 으로 서버와 연결시에 사용한다. 만약 저장하고 있던 UUID 가 삭제된 경우에는 다시 UUID 발급해야 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 - 200: 성공 - 400: 파라미터 오류 - 401: 권한 없음(client_id, client_key불일치) - 406: 일치하지 않는 정보 - 416: 등록 허용 단말 개수 초과 - 500 ~ 502 : Agent Server 오류

이름	유형	설명
		- -1 : 알수없는 시스템에러 (SDK Exception) - -2 : (android SDK) InsideListener, Context 없음. Inside() API로 초기화 필요
rcmsg	String	결과 메시지
uuid	String	단말(사용자별) Unique한 ID

■ 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 = ""; // USERID 는 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) { // rc 가 200 인 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 = ""; // USERID 는 클라이언트 마다 다른 값이 설정되어야 함(예:
단말의 MAC 주소로 설정)

REGESITER_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;

```

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

REGESITER_CODE rc;
rc = 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_init

SDK 를 사용하기 위해서 Client Key 정보와 UUID 정보로 Agent Server 와 연결한다. 사용자가 입력한 UUID 는 서버에서 정상적으로 발급된 UUID 인지 확인하는 과정을 거치게 되는데 이때 만약 rc 가 404 로 리턴된 경우에는 agent_register()에서 설명한 UUID 발급 과정을 다시 거쳐야 한다.

agent_init 은 유효한 UUID 를 발급 받은 후 최초 1 회만 호출되어야 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	결과 코드 - 200: 성공 - 404: Client 정보(UUID를) 찾을 수 없음 - -1 : 알수없는 시스템에러 (SDK Exception)

이름	유형	설명
		- -2 : (android SDK) InsideListener, Context 없음. Inside() API로 초기화 필요
rcmsg	String	결과 메시지

■ 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_setCommandEventCallback

서버가 전송하거나 Agent SDK 로 부터 API 호출에 대한 이벤트나 커맨드를 수신하기 위해 콜백 함수를 등록한다. 커맨드와 이벤트로 전달되는 데이터는 9.1 절을 참고한다. 해당 API 는 **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

```

REGESITER_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);
}

void CginsidemfctestDlg::onCommand(std::string actiontype, std::string cmd) {
    ....
}

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

9.5 agent_onCommand

Service Agent 에게 AI Agent Server 로부터 전달받은 응답 이벤트, TTS, 미디어 정보를 전달한다. Service Agent 는 해당 command 를 받기 위한 리스너를 구현해야 한다.

■ API

agent_onCommand(string actionType, string payload)

■ Parameter

이름	유형	설명
actionType	string	단말에서 수행해야 할 action Type

이름	유형	설명
payload	string	Agent Server로부터 전달 받은 payload(JSON 포맷)

서버는 아래와 같이 **9 가지 actionType** 단말로 전달할 수 있으며 각 actionType 에 따라 payload 로 전달되는 내용은 다음과 같다.

(V1.0.3) actionType	(V1.1.0) actionType	단말 수행 동작
start_voice	start_voice	Mic ON 후 음성인식 시작
-	stop_voice	Mic OFF 후 음성인식 결과 확인
media_stream / media_url	play_media	Payload의 'channel'정보를 통해 재생할 Media 정보를 확인. <u>Payload에 URL 정보가 전달된 경우 재생.</u> Payload에 TTS 정보가 전달된 경우 actionType: media_data 정보 전달 기다림.
media_stream(contentType:wav)	media_data	Payload에 전달되는 TTS mediastream을 base64 decoding 후 재생.
dialog_response	exec_dialogkit	Service Agent에서 자체적으로 제공하는 서비스를 위해 개발자센터(API Link)사이트의 Dialog Kit에 등록된 정보가 전달됨
media_url(comandType: Req_UPMD)	control_media	Payload에 전달되는 'channel'에 해당하는 미디어를 제어
hardware_control	control_hardware	블루투스/볼륨 제어명령을 수행
timer_set	set_timer	타이머 설정기능 수행
webview_url	webview_url	전달된 웹뷰URL을 단말 브라우저로 실행

9.5.1 actionType : start_voice

서버에서 음성인식을 시작하라는 명령어가 전달된다.

Payload JSON Object			Type	Description
cmdOpt			JSON	

actionType: start_voice의 payload 전달 예시

```
{ "cmdOpt": { } }
```

단말에서는 Mic On 후 음성인식 시작하면 된다. (연속 발화 서비스도 동일)

예) "일정 등록해줘" -> "일정을 등록하실 날짜를 말씀해주세요" TTS 재생 완료 후 -> acent_onCommand로 actionType "start_voice" 전달된다.

9.5.2 actionType : stop_voice

서버에서 음성인식 처리 결과가 전달된다.

Payload JSON Object			Type	Description
cmdOpt			JSON	
	resultCd		number	처리 결과 코드 - 200: 성공 - 412: Speech Timeout - 416: Audio Timeout - 500: Agent Server 오류
	uword		number	음성인식 결과 Text
	nextCmd		String	다음으로 처리할 명령어
	nextCmdOpt		JSON	다음으로 처리할 명령어 옵션
		channel	number	채널 정보
		actOnOther	pause	채널 정보 액션(9.5.3 참고)

-

actionType: stop_voice의 payload 전달 예시

```
{ "cmdOpt": {"uword": "안녕하세요", "resultCd": 200} }
```

화면이 있는 경우 uword를 노출하고 nextCmd Req_PLMD에 해당하는 play_media 정보를 기다린다. nextCmd가 내려올 경우 기존 다른 Media 재생중이면 pause 처리한다.

9.5.3 actionType : play_media

서버에서 전달할 미디어 출력정보(TTS Text 또는 Media URL) 및 옵션 정보를 전달한다.

TTS 재생(type:wav)의 경우 TTS 메시지 정보와 다른 출력 미디어에 대한 action 정보가 내려오며 실제 TTS mediastream 정보는 actionType : media_data(9.5.4) 로 전달된다.

곡이 내려오는 경우 재생할 곡의 URL 이 전달된다. 다른 출력 미디어에 대한 action 처리 후 미디어 URL 을 재생한다. media_url 이 전달될 때 playNotiTime 에 설정된 값이 있으면, 해당 값의 주기에 맞추어 agent_updateMediaStatus API(9.7) 로 서버에 미디어 재생 상태 정보를 보내야 한다.

[Case#1. TTS Text 정보 전달]

Payload JSON Object			Type	Description
cmdOpt			JSON	
	channel		number	채널 정보
	metaInfo		JSON	
		mesg	String	TTS로 재생할 Text 정보
	actOnOther		String	다른 출력 미디어에 대한 action (미디어 Action Table 참고)

[미디어 Action Table]

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 pause상태 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 후 volume-down 상태유지
VolDownR	다른 미디어가 재생중이면 volume-Down, TTS 출력 완료 후 기존 볼륨 복구

actionType : play_media(TTS Text 정보 전달) payload 예시

```
{ "cmdOpt": { "channel":0, "actOnOther":"stop", "metaInfo":{"mesg":"지니뮤직을 실행합니다"}} }
```

[Case#2. Media URL 정보 전달]

Payload JSON Object			Type	Description
cmdOpt			JSON	
	channel		number	채널 정보
	actOnOther		String	다른 출력 미디어에 대한 action (미디어 Action Table 참고)
	url		String	재생할 URL 정보
	playNotiTime		number []	재생시간 noti 주기 (Second)
	metaInfo		JSON	
		contentType	String	[첨부1] metaInfo 상세 설명의

Payload JSON Object			Type	Description
cmdOpt			JSON	
				contentType 표 참고
		infoType	String	컨텐츠 상세 내용 전달 Type
		infoDetail	JSON	[첨부2] 서비스별 infoDetail 참고

※ Service Agent 는 playNotiTime 정보가 설정되어 전달되는 경우
재생시간 노티 주기에 따라 반드시 agent_updateMediaStatus API 로 'noti'
이벤트를 전달해야 한다. (지니뮤직 서비스의 경우 반드시 구현 필요)

※ 컨텐츠 메타 정보는 metaInfo JSON 으로 전달되며 contentType 에 따라
infoDetail JSON 정보가 달라진다. (첨부 1, 첨부 2 참고)

actionType : play_media의 media_url 전달 payload 예시
<pre>{ "cmdOpt": {"channel":101, "actOnOther":"pause","url":"http://app-genie.~~", "playNotiTime" : [5, 60], metaInfo":{"contentType" : "music", "infoType" : "text", infoDetail" : {"title" : "곡명""artist" : "가수명", "imageurl": "http://image.genie.~~", duration : 202}} }</pre>
UI가 필요한 경우 metaInfo로 전달 받은 미디어 정보를 화면 표시 후, 전달 받은 url을 재생

9.5.4 actionType :media_data

실제 TTS mediastream 은 아래와 같이 base64 encoding 되어 전달되며 Service Agent 에서 base64 decoding 을 해서 사용한다.

Payload JSON Object			Type	Description
contentType			String	컨텐츠 타입(wav)
mediastream			byte []	미디어 데이터(base64 encoded)

actionType : media_data 전달 예시 (TTS mediastream 전달)

```
{"contentType":"wav","mediastream":" UkIGRkTfAABVZFZAAABEAgAB9AACC....."}
```

UI 작업이 필요한 경우 이전에 actionType:play_media로 전달 받은 TTS text 정보를 화면에 표시 후, actionType:media_data로 전달 받은 TTS mediastream을 base64 decoding 후 재생한다

9.5.5 actionType : exec_dialogkit

서버에서 개발자센터 사이트의 Dialog Kit 에 등록된 정보를 전달한다.

Payload JSON Object				Type	Description
cmdOpt				JSON	
	execType			String	dialogHandler
	execOpt			JSON	
		action		String	kt.action.dialog.custom
		dssMsg		JSON	DialogKit 등록 정보
			Intent	String	Dialog Kit에 등록한 Intent
			appInfo	JSON	Dialog Kit에 등록한 NE 정보(Key-Value) 값이 전달됨
		__systemMesg		String	Dialog Kit 인텐트 답변이 전달됨 (systemMesg 앞 언더바가 두개임)

actionType: ex_dialogki의 payload 전달 예시

```
"cmdOpt" : { "execType" : "dialogHandler", "execOpt" : { "action" : "kt.action.dialog.custom",
"dssMsg" : { "Intent" : "lightTurnOff", "appInfo" : {"NE-LIGHT" : "전등"}, "__systemMesg" :
"전등을 켜습니다" } } }
```

payload내 json 형식의 대화 답변을 파싱하여 Service Agent의 단말 특화 대화 처리

9.5.6 actionType : control_media

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

Payload JSON Object			Type	Description
cmdOpt			JSON	
	channel		number	채널 정보
	playTime		number	플레이 시간 (단위: millisecond)
	act		String	미디어 변경 action

전달되는 action 은 다음과 같다.

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)

actionType : play_media의 media 제어 payload 예시
{ "cmdOpt": {"channel":101, "act":"pause", "playTime": -1} }
act로 전달받은 액션을 수행 (재생중인 음악 재생 중지)

9.5.7 actionType : control_hardware

H/W(volume, Bluetooth) 제어 명령은 다음과 같이 내려온다.

Payload JSON Object			Type	Description
cmdOpt			JSON	
	target		String	대상 H/W (예: volume, bluetooth)
	hwCmd		number	Command(예: setVolume)
	hwCmdOpt		JSON	
		control	String	요청 Command Option
		value	String	요청 Command Option value
	trxid		String	Transaction Id

전달되는 정보는 다음과 같다.

target	hwCmd	hwCmdOpt	설명
bluetooth	controlClassicSvc	control: play(재생), stop(중지), pause(일시정지), resume(일시정지 해 제), rewind(처음으로), forward(다음곡), backward(이전곡)	Bluetooth 제어 명 령을 설정한다.
volume	setVolume	control: UP(올림), DN(내림), MT(뮤트), UMT(뮤트 해제), ST(값으로 설정) value: control이 UP/DN인 경우 GN(일 반), MO(많이), LE(조금), MX(최대), ST의 경우 숫자 값	볼륨 제어명령을 설정한다.

actionType : hardware_control 의 payload 전달 예시

```
{ "cmdOpt": {"target": "volume", "hwCmd": "setVolume", "hwCmdOpt": {"control": "UP"},  
"trxid": "2019021921020350338Q2IPafH" }
```

단말 동작: 전달받은 볼륨 제어 정보를 수행 (볼륨 업)

9.5.8 actionType : set_timer

타이머 설정 요청 명령은 다음과 같이 내려온다.

Payload JSON Object			Type	Description
cmdOpt			JSON	
	setOpt		String	설정 옵션(set: 설정, clear: 해제)
	reqAct		String	요청하는 Act 유형 - "0": 타이머 - "1"~"8": 알람
	actionTrx		String	설정 액션 Transaction
	setTime		String	설정 시간(YYYYMMDDHH24MISS)

actionType : set_timer 의 payload 전달 예시

```
{ "cmdOpt": {"setOpt": "set", "reqAct": "0", "actionTrx": "2019021921020350338Q2IPafH",
"setTime": "20190619103059189"} }
```

단말 동작: 단말에 6월 19일 10시 30분 59초에 타이머 설정

9.5.9 actionType : webview_url

서비스 로그인이 필요한 경우 다음과 같이 로그인 페이지 URL 이 내려온다.

Payload JSON Object			Type	Description
cmdOpt			JSON	
	oauth_url		String	로그인 페이지 URL

actionType : webview_url 의 payload 전달 예시

```
{ "cmdOpt": {"oauth_url": "http://www.genie.com/login" } }
```

전달받은 로그인 URL을 디바이스의 브라우저로 실행

9.6 agent_onEvent

Service Agent 에게 SDK 이벤트를 전달한다. Service Agent 는 해당 event 를 전달받으면 상황에 따른 적절한 예외처리 또는 UI 표기를 할 수 있다.

■ API

agent_onEvent(int evt, string opt)

■ Parameter

이름		유형	설명
evt		int	이벤트명의 Integer 값
opt		JSON	이벤트에 따라 전달되는 추가 정보 (JSON포맷)
	errorCode	String	SERVER_ERROR인 경우 에러코드가 전달됨
	errorMsg	String	SERVER_ERROR인 경우 에러 TEXT가 전달됨

- 이벤트명과 추가 정보는 다음과 같다.

evt	value	Description	Service Agent 구현
GRPC_INIT_SUCCESS	0	gRPC 연결 성공	-
GRPC_INIT_FAIL	1	gRPC 연결 실패	gRPC 연결 재시도
GRPC_DISCONNECTED	2	gRPC 연결 끊김	-
GO_TO_STANDBY	3	요청 정상 처리, 요청에 대한 응답 TTS 없음	음성명령 대기상태로 전환, 미디어 서비스 상태는 유지

evt	value	Description	Service Agent 구현
SERVER_ERROR	205	서버 에러	에러코드에 맞게 재시도 or 클라이언트 초기화 후 재시도

■ 사용 예시

- Android

```
@Override
public void agent_onEvent(int evt, String opt) {
    switch (evt) {
        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::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::SERVER_ERROR:
            // 에러 처리
            ...
            break;
    }
}
```

9.7 agent_updateMediaStatus

미디어 상태 정보를 전달한다. SDK로부터 전달받아 재생되는 모든 미디어의 상태 정보(시작/중지/Noti 등) 전달이 필요하다. 지니뮤직 재생의 경우 playNotiTime 으로 전달된 주기에 따라 미디어 재생 상태정보를 반드시 전달해야 한다.

■ API

agent_updateMediaStatus(int ch, string state, int playtime)

■ Parameter

이름	유형	설명
ch	int	서버에서 전달받은 채널 값

이름	유형	설명
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

■ 사용 예시

- Android

```
insideSDK.agent_updateMediaStatus(0,"started",0); //TTS 재생 시작
insideSDK.agent_updateMediaStatus(0,"noti",30); //agent_onCommand 로
"playNotiTime"이 30 으로 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
```

9.8 agent_sendText

음성 대신 Text 로 사용자 명령을 전달할 때 사용한다.

■ API

agent_sendText(string msg)

■ Input Parameter

이름	유형	설명
msg	string	명령 텍스트

■ 사용 예시

- Android

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

@Override
public void onCommand(String actionType, String payload) {
    // actionType 으로 분기 필요
    if(actionType.equals("play_media") {
        // 처리 필요
    }
}
```

- Linux

```
agent_sendText("안녕하세요");

void onCommand(std::string actionType, std::string payload ) {
    cJSON *cmdp_jsonObj = cJSON_Parse(payload.c_str());

    // actionType 으로 분기 필요
    if (strcmp(actionType.c_str(), "play_media") == 0) {
        // 처리 필요
    }
}
```

- Windows

```
agent_sendText("안녕하세요");
```

```
void CginsidetestDlg::onCommand(std::string actionType, std::string payload ) {
    cJSON *cmdp_jsonObj = cJSON_Parse(payload.c_str());

    if (cmdp_jsonObj != NULL) {
        // actionType 으로 분기 필요
        if (strcmp(actionType.c_str(), "play_media") == 0) {
            // payload 처리 필요
        }
    }
}
```

9.9 agent_startVoice

사용자 명령을 음성으로 전달하기 위해 음성인식 시작을 요청한다.

■ API

agent_startVoice()

■ 사용 예시

- Android

```
insideSDK.agent_startVoice();

@Override
public void onCommand(String actionType, String payload) {
    switch (actionType) {
        case Inside.START_VOICE:
            // 단말 마이크 On
            // agent_sendVoice 호출
            break;
        ....
    }
}
```

- Linux

```
agent_startVoice();

void onCommand(String actionType, std::string payload)
{
    if (strcmp(actionType.c_str(), "start_voice") == 0) {
        // 단말 마이크 On
        // agent_sendVoice 호출
    }
}
```

- Windows

```
agent_startVoice();

void CginsidetestDlg:: onCommand (String actionType, std::string payload) {
    cJSON *cmdp_jsonObj = cJSON_Parse(payload.c_str());

    if (cmdp_jsonObj != NULL) {
        // actionType 으로 분기 필요
        if (strcmp(actionType.c_str(), "start_voice") == 0) {
            // payload 처리 필요
        }
    }
}
```

9.10 agent_sendVoice

음성인식 시작 명령을 받으면(agent_onCommand 의 actionType:start_voice) 사용자가 발화한 음성 데이터 스트림을 서버로 전송한다. 이 때 Service Agent 는 음성인식 파형 등의 UI 작업을 수행하면 된다. 이후 서버로부터 음성인식 중지 명령(agent_onCommand 의 actionType:stop_voice)을 받으면 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 (단위 : 샘플 Size) Windows SDK를 사용하는 경우 반드시 1280 byte 이상 전송해야 한다

■ 사용 예시

• Android

```
@Override
public void onCommand(String actionType, String payload) {{
    switch (actionType) {
        case Inside.STOP_VOICE:
            // 단말 마이크 Off
            // opt 로 전달된 음성인식 결과 표출
            break;
    }
}

@Override
public void onCommand(String actionType, String payload) {{
    switch (actionType) {
        case Inside.START_VOICE:
            // 단말 마이크 On
            // agent_sendVoice 호출
            break;
    }
}
```

• Linux

```

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

void onCommand(String actionType, std::string payload)
{
    if (strcmp(actionType.c_str(), "start_voice") == 0) {
        // 단말 마이크 Off
        // opt 로 전달된 음성인식 결과 표출
        ....
    }
}

```

- Windows

```

HRESULT __cdecl RecordCallback(BYTE* pRecordedData, int iSize) {
    short pcm_val[1280];
    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;
    }

    agent_sendVoice(pcm_val, iSize / 2);

    return 0;
}

void CginsidetestDlg::onCommand (String actionType, std::string payload) {
    cJSON *cmdp_jsonObj = cJSON_Parse(payload.c_str());

    if (cmdp_jsonObj != NULL) {
        // actionType 으로 분기 필요
        if (strcmp(actionType.c_str(), "stop_voice") == 0) {

```

```
// payload 처리 필요. 음성인식 결과가 전달됨  
}  
}
```

9.11 agent_stopVoice

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();
```

9.12 agent_getTTS

입력한 텍스트를 TTS 스트림으로 변환한다.

■ API

agent_getTTS(string msg)

■ Parameter

이름	유형	설명
msg	string	TTS 로 변환할 텍스트

■ Result

이름	유형	설명
rc	int	결과 코드 - 200: 성공 - 500 ~ 502 : Agent Server 오류 - -1 : 알수없는 시스템에러 (SDK Exception)
rcmsg	byte []	미디어 데이터(base64 encoded)

■ 사용 예시

- Android

```
String ret = insideSDK.agent_getTTS("안녕");

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

if(rc == 200) {
    // 미디어 데이터 재생
}
```

- Linux

```
std::string strRet = agent_getTTS("안녕");
if (strRet.size() > 0) {
    cmdp_jsonObj = cJSON_Parse(strRet.c_str());
    if (cmdp_jsonObj != NULL) {
        cJSON *rc = cJSON_GetObjectItem(cmdp_jsonObj, "rc");
```



```

cJSON *rcmsg = cJSON_GetObjectItem(cmdp_jsonObj, "rcmsg");
if(rc != NULL && rcmsg != NULL) {
    if(rc->valueint == 200) {
        std::string media(rcmsg->valuelstring);
        std::string decoded = base64_decode(media);
        int voiceDataSize = (int)decoded.size();
        const char* voiceData = decoded.c_str(); // tts wave file
    }
}
cJSON_Delete(cmdp_jsonObj);
}

```

- Windows

```

std::string strRet = agent_getTTS("안녕");

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) {
            // media stream 재생
        }
    }
}

```

9.13 agent_sendCommand

서버로 하드웨어 및 타이머 Event 를 전달할 때 사용한다. Service Agent 가 동작하고 있는 단말에서 Volume, Bluetooth/BLE, Timer, Button 이벤트가 발생한 경우 Agent Server 로 해당 내용을 전송해야 한다.

■ API

agent_sendCommand(string cmd)

■ Parameter

[Case #1] 단말 하드웨어 이벤트를 전달 (Snd_HWEV)

cmd JSON Object				Type	Description
commandType				String	Snd_HWEV
payload				JSON	
	cmdOpt			JSON	
		target		String	대상 H/W (예: button, volume, bluetooth)
		hwEvent		String	[target : button 인 경우] Btn_PU (Play or Pause 눌러짐) Btn_NX (Next 눌러짐) Btn_PV (Previous 눌러짐) [target : volume 인 경우] setVolume [target : bluetooth 경우] BTEvent
		hwEventOpt		JSON	
			value	number	[target : volume 인 경우] 볼륨 설정 값(ex. 3)
			event	String	[target : bluetooth 경우] A2DP 연결 상태 값 801(A2DP Play) 800(A2DP Not Connect)

sendCommand - Snd_HWEV 예시	
<pre>{ "commandType":"Snd_HWEV", "payload":{ "cmdOpt": {"target":"volume", "hwEvent":"setVolume", "hwEventOpt":{"value":3}} } }</pre>	
단말의 볼륨 3으로 변경시 서버로 이벤트 전달	

[Case #2] 설정한 Timer 시간이 되었다는 이벤트를 전달 (Snd_TMEV)

설정된 Timer 시간이 도래했을 때 Event 를 전달한다.

cmd JSON Object			Type	Description
commandType			String	Snd_TMEV
payload			JSON	
	cmdOpt		JSON	
		reqAct	String	Action 유형 ("0": 타이머, "1"~"8": 알람)
		actionTrx	String	서버로부터 받은 액션 Transaction (9.5.8 actionType:set_timer 참고)
		localTime	String	단말 Event 생성 시간 (YYYYMMDDHH24MISS)

sendCommand - Snd_TMEV 예시	
<pre>{ "commandType":"Snd_TMEV", "payload":{ "cmdOpt": {"reqAct":"0", "actionTrx":"20190219210350afH", "localTime":"20190619103059189"} } }</pre>	
단말 동작: 설정한 타이머 시간이 도래했다는 이벤트를 서버로 전달	

9.14 agent_serviceLogin

사용자 로그인이 필요한 서비스의 로그인 페이지 정보를 요청한다. 해당 API 를 호출하면 service_type 에 따른 로그인 페이지 URL 이 전달 되며 사용자가 로그인을 완료하면 입력한 return_url 로 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 - 200: 성공 - 401: 허용되지 않는 호출 - 404: 지원되지 않는 service_type 임 - 500: Agent Server 오류 - -1: 알수없는 시스템에러 (SDK Exception).
rcmsg	string	결과 메시지
oauth_url	string	login_type이 OAuth 경우 전달됨

■ 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
}
}
```

9.15 agent_serviceLoginStatus

서비스의 로그인 상태 정보를 요청한다. 해당 API 를 호출하면 service_type 에 따른 로그인 상태 정보 oauth_status 와 access_token 이 전달된다.

■ API

agent_serviceLoginStatus(string service_type)

■ Parameter

이름	유형	설명
service_type	string	로그인 할 서비스 (예: geniemusic)

■ Result

이름	유형	설명
rc	number	결과 코드 - 200: 성공 - 401: 허용되지 않는 호출 - 404: 서비스 미로그인 상태이거나 지원되지 않는 service_type 임 - 500: Agent Server 오류 - -1: 알 수 없는 시스템 에러가 발생하였습니다.
rcmsg	string	결과 메시지

이름	유형	설명
oauth_status	string	"valid" : OAuth 인증 상태 "invalid" : OAuth 인증 필요
access_token	string	oauth_status가 valid인 경우 전달
user_name	string	oauth_status가 valid 이고 정보가 있는 경우 전달

■ 사용 예시

- Android

```
String ret = insideSDK.agent_serviceLoginStatus("geniemusic");
// ret = "{\"rc\":200,\"rcmsg\":\"성공\",\"oauth_status\":\"valid\",\"access_token\":\"~~~\"}"

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

if(rc == 200) {
    // oauth_status 가 valid 이면 유효한 access_token 저장 및 사용
}
```

- Linux

```
std::string strRet = agent_serviceLoginStatus("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) {
            // oauth_status 가 valid 이면 유효한 access_token 저장 및 사용
        }
    }
}
```

```
}
}
```

- Windows

```
std::string strRet = agent_serviceLoginStatus ("geniemusic", "");
// ret = "{\"rc\":200,\"rcmsg\":\"성공\",\"oauth_status\":\"valid\",\"access_token\":\"~~~\"}"

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) {
            // oauth_status 가 valid 이면 유효한 access_token 저장 및 사용
        }
    }
}
```

9.16 agent_serviceLogout

사용자가 로그인한 서비스의 로그아웃을 요청한다. 해당 API 를 호출하면 서비스 로그아웃 상태가 된다.

■ API

agent_serviceLogout(string service_type)

■ Parameter

이름	유형	설명
service_type	string	로그인 할 서비스 (예: geniemusic)

■ Result

이름	유형	설명
rc	number	결과 코드 - 200: 성공 - 401: 허용되지 않는 호출 - 404: 지원되지 않는 service_type 임 - 500: Agent Server 오류 - -1: 알수없는 시스템에러 (SDK Exception)
rcmsg	string	결과 메시지

■ 사용 예시

- Android

```
agent_serviceLogout("geniemusic");
```

- Linux

```
agent_serviceLogout ("geniemusic");
```

- Windows

```
agent_serviceLogout ("geniemusic");
```

9.17 agent_debugmode

Agent SDK 디버그 모드를 설정한다. 로그는 실행 경로에 'ginside.log' 파일로 저장된다. (Android 는 logcat 으로 확인)

■ 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);
```

9.18 agent_setServerInfo

SDK 에서 연동할 서버의 정보를 설정한다. 개발/상용 서버 정보는 2.4 연동서버 정보에서 확인 가능하다. 서버 정보 변경 후에는 반드시 agent_init() 을 통해 변경된 서버로 연결하는 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

이름	유형	설명
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("inside-dev.gigagenie.ai", "50109", "30109");
}
```

- Linux

```
agent_setServerInfo("inside-dev.gigagenie.ai", "50109", "30109");
```

- Windows

```
agent_setServerInfo("inside-dev.gigagenie.ai", "50109", "30109");
```

9.19 agent_getVersion

SDK 버전 정보를 조회한다.

■ API

agent_getVersion()

■ 사용 예시

- Android

```
string ver = insideSDK.agent_getVersion();
```

- Linux

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

- Windows

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

9.20 agent_setLocation

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
Latitue	String	위도 값
Address	String	시/구/군 주소가 정확히 입력되어야 한다. (ex. 서울시 X -> 서울특별시 O)

■ 사용 예시

- Android

```
// 위치값 설정
insideSDK.agent_setLocation("127.029000", "37.4713370", "서울특별시 서초구
```

```
태봉로");

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

- Linux

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

- Windows

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

9.21 agent_unregister

발급된 UUID 를 삭제 요청한다. 즉 등록된 단말 정보를 삭제한다.

■ API

agent_unregister()

■ Result

이름	유형	설명
rc	int	결과코드 - 200: 성공 - 404: Client 정보(UUID를) 찾을 수 없음 - 500 : 시스템 오류 - -1 : 알수없는 시스템에러 (SDK Exception)
rcmsg	String	결과 메시지

■ 사용 예시

- Android

```
String ret = insideSDK.agent_unregister();  
// ret = "{\"rc\":200,\"resMsg\":\"unRegister 에 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  
}
```

9.22 agent_reset

SDK 사용 중 이상이 있거나 Service Agent 사용자가 변경되었을 경우 SDK 를 초기화하기 위해서 사용한다. 해당 API 호출 시 Service Agent 는 인스턴스 및 미디어 플레이어 등 모든 값을 초기화 하여야 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
```


10. KWS API 상세

10.1 kws_init

호출어 기능을 준비한다. 해당 API 는 agent_init 완료 후 사용할 수 있다.

■ API

kws_init(int kwsld)

■ Parameter

이름	유형	설명
kwsld	int	Optional 값으로 세팅할 키워드 (dafault: 지니야) 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_detect

호출어 검출을 위한 음성 데이터 stream 을 전송한다.

■ 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 - -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_reset

호출어 라이브러리를 초기화한다. 호출어 인식 결과 오류가 발생하면 사용한다.

■ API

void **kws_reset**()

■ 사용 예시

- Android

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

- Linux

```
kws_reset();
```

- Windows

```
kws_reset();
```

10.4 kws_error

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

■ API

int **kws_error()**

■ Result

이름	유형	설명
error_code	int	호출어 라이브러리 에러코드

■ 사용 예시

- Android

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

- Linux

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

- Windows

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

10.5 kws_getVersion

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_getKeyword

현재 호출어 라이브러리에서 사용 중인 호출어 ID 를 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_setKeyword

사용할 호출어를 변경한다. 호출어 모델 파일 cnsf / cnsi 의 파일명은 kwsId 에 대응한다. 즉, kws_setKeyword(2)로 세팅하면 2.cnsf(2.cnsi)에 해당하는 호출어 (친구야)가 적용된다. 호출어 교체 시 SDK 변경 없이 cnsf/cnsi 파일만 교체하면 된다. 3rd Party 고유의 호출어가 필요할 경우 KT 담당자와 협의가 필요하다.

■ 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(0);
```

- Windows

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

10.8 kws_setModelPath

호출어 모델 파일의 디렉토리를 설정하는 함수로 Linux, Window SDK 에서만 사용 가능하다. 호출어 모델파일 경로를 변경 후 kws_init() 호출이 필요하다.

Android 의 경우 asset 파일로 별도로 전달되는 호출어 모델 파일을 app/src/main/assets/kwsrnet 에 위치 시켜야 한다. (g-sdk-android_1.0.2 이후 필수 적용)

■ 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

onEvent 로 전달되는 SERVER_ERROR errCode에 대해 다음과 같이 처리한다.

- 601: 음성인식 요청 후 음성인식 시작 수신을 받지 못한 경우
- 602: 음성인식 시작 후 10 초 뒤에 음성인식 종료 수신을 받지 못한 경우
- 603: 음성인식 종료 수신 후 아무런 데이터를 받지 못한 경우
- 900: 음성인식 서버 접속 실패
- 901: 음성인식 시작 후 5 초 이내에 실제 음성 감지가 발생하지 않는 경우
- 902: 음성인식 시작 후 10 초 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

errCode	처리방법
601~603	Standby 상태로 변경 (음성인식이 정상적으로 되지 않아 다시 음성 명령어 대기상태로 변경 필요).
900~903	현재 서버 접속에(or 음성인식에) 문제가 있으니 잠시 후 다시 시도 해 주세요 TTS 안내 후 Standby 상태로 변경
902	주변이 시끄러워 정확히 듣지 못했습니다. 다시 이용해 주세요 TTS 안내 후 Standby 상태로 변경

errCode	처리방법
910~911	현재 서버 접속에(or TTS 재생에) 문제가 있으니 잠시 후 다시 시도해 주세요 TTS 안내 후 Standby 상태로 변경
912	(주로 개발 중 발생) TTS 요청 텍스트 포맷 이상, 서비스앱 관리자/담당자에게 문의 안내 후 Standby 상태로 변경
920	현재 서버 접속에(or 대화 해석에) 문제가 있으니 잠시 후 다시 시도해 주세요 TTS 안내 후 Standby 상태로 변경
921	(주로 개발 중 발생) 대화 서버 연동 이상, 인사이드 담당자에게 문의 안내 후 Standby 상태로 변경
930	현재 서버 접속에 문제가 있으니 잠시 후 다시 시도해 주세요 TTS 안내 후 Standby 상태로 변경 (인사이드 담당자에게 문의)

11.2 FAQ

문의	답변
인사이드에서는 어떤 서비스를 제공하나요?	"감성채팅/알림타이머/날짜/시간/메모/날씨/환율/배달음식/지역정보/길안내/위키디피아/지니뮤직/뉴스/라디오/팟캐스트" 서비스를 제공하고 있습니다. 지속적으로 제공 서비스를 확대해 나갈 예정입니다.
서비스별 발화 명령어는 어떻게 되나요?	서비스별 발화 예시 문장은 기가지니 서비스별 안내 페이지를 (https://gigagenie.kt.com/svcinfo.do) 참고하시면 됩니다.
디바이스에서 제공할 서비스 추가/변경은 어떻게 하나요?	디바이스 등록 시 원하는 "서비스 유형"을 선택하시면 됩니다. 이후에는 My Device 메뉴에서 "디바이스 수정"을 통해 서비스 변경이 가능합니다.
인사이드 서비스 UI/디자인 가이드 적용할 부분이 있나요?	GiGA Genie INSIDE 배지 이미지를 규격에 맞게 노출해 주시면 됩니다 (Black, White 두가지 형태가 있으며 API Link 사이트에서 이미지 다운로드 받으실 수 있습니다) Mobile SDK를 통해 귀사의 어플리케이션에 적용할 경우 앱을 통해

문의	답변
	<p>이미지 확인이 되어야 합니다.</p> <p>Device SDK를 적용하여 귀사 자체 Device에 적용할 경우 제품 패키지에 적용되어야 합니다.</p>
개발서버와 상용서버 정보가 어떻게 되나요?	<p>개발서버 정보는 다음과 같습니다.</p> <p>HOST: inside-dev.gigagenie.ai, gRPC Port: 50109, REST Port: 30109</p> <p>상용서버 정보는 다음과 같습니다.</p> <p>HOST: inside.gigagenie.ai, gRPC Port: 50051, REST Port: 9080</p>
하나의 디바이스 등록만 가능한가요?	<p>여러 개의 디바이스 등록이 가능합니다.</p> <p>하지만 등록 디바이스별 Client ID/Client Key가 다르게 발급&관리되므로 디바이스 모델이 동일하다면 하나로 등록해서 사용하시면 됩니다.</p> <p>동일한 Client ID / Client Key를 이용하더라도 실제 동작하는 디바이스 별로 unique 한 ID(uuid)가 부여되므로 개발 및 사용하시는데 불편함은 없습니다.</p>
디바이스 등록 시 장치지원 여부 항목이 무엇인가요?	<p>장치지원 항목에서 화면, 카메라, 블루투스 지원/미지원 선택에 따라 제공되는 서비스 제공이 달라질 수 있습니다. 예를들어 블루투스 미지원 단말로 등록한 경우 기가지니 서비스 중 블루투스 장치 지원이 필요한 서비스 요청 시 지원하지 않는 서비스라는 답변이 제공됩니다.</p>
Device SDK와 Mobile SDK 차이가 뭔가요?	<p>1. Device SDK : Device OS Platform 별로 제공하는 디바이스 전용 SDK 입니다.</p> <p>[지원 개발환경]</p> <ul style="list-style-type: none"> - Android API level 19 or later - Linux : Ubuntu x86_64, CentOS x86_64, Raspbian armv7l, Yocto aarch64(G-INSIDE reference H/W) - Windows 64-bit <p>2. Mobile SDK : Mobile OS Platform 별로 제공하는 모바일 전용 SDK 입니다.</p> <p>[지원 개발 환경]</p> <ul style="list-style-type: none"> - Android API level 22 or later, iOS 9 or later <p>개발 시 가장 큰 차이점은 Device SDK의 경우 H/W 의존성 최소화를 위해 오디오 캡처를 (3rd Party) Service Agent 앱에서 수행하고,</p>

문의	답변
	Mobile SDK의 경우 OS기본 마이크 API를 이용하여 SDK 내부에서 오디오 캡처를 수행합니다.
Dialog Kit은 무엇인가요?	<p>디바이스 특화 대화모델을 Dialog Kit을 통해 구성할 수 있습니다.</p> <p>* 어휘사전 : 서비스 내 발화에서 필요한 어휘에 대한 사전 입니다. (NE사전) 서비스에 사용 할 명사들을 정의 사전명과 함께 Parameter로 전달 가능 ex. TV,에어컨 등. (PR 사전) 행위/동작 등의 서술어들을 정의하는 사전 ex. 커줘,주문해줘 등</p> <p>* 인텐트 : 사용자의 발화 의도를 구분하는 용어입니다. 디바이스 서비스 App에서 인텐트와 어휘사전으로 특화 서비스를 구현하면 됩니다.</p>
Dialog Kit은 언제 사용가능 하나요?	<p>Device 등록 신청이 완료(승인)되면 이후 Dialog Kit을 통한 단말 특화 서비스 생성이 가능하며 "배포"를 진행해 주셔야 생성한 대화모델이 반영됩니다.</p> <p>상용 시스템에 반영이 필요할 경우 반드시 "상용배포"를 진행해 주셔야 합니다.</p>
개발/검증/상용배포 절차가 궁금합니다.	<p>디바이스 등록 후에 개발키(Client ID/Client Key)가 발급되어 개발을 진행합니다.</p> <p>개발 완료 시 검증 신청을 요청합니다. 이때 상용키(Client ID/ClientKey)가 발급됩니다.</p> <p>발급된 상용키를 적용하여 검증 의뢰를 합니다. (단, 이때 Dialog Kit 상용배포 적용이 반드시 필요합니다)</p> <p>검증 진행(N 차수 진행) 후 최종 검증 완료 및 상용 승인 됩니다. (모바일 앱의 경우 OneStore/Play Store/App Store에 등록하시면 됩니다.)</p>
"프리징 중입니다" 라고 뜨면서 Dialog Kit 배포 요청을 할 수가 없어요	현재 검증 진행 중 상태로 "프리징" 된 것입니다. 검증 완료 후 다시 "개발" 상태가 되면 Dialog Kit 수정 후 "배포요청"이 가능합니다.
"해당 단말에서 지원하지 않는 서비스 입니다" 라고 나	연동하는 서버 IP/Port 정보가 맞는지 확인합니다. 디바이스 등록 시 제공할 서비스 유형을 체크 했는지 확인합니다.

문의	답변
와요	(My Device > GiGA Genie Inside “변경신청” > 디바이스 수정 화면에서 신청된 서비스 유형 확인이 가능합니다) 디바이스 수정화면에서 원하는 서비스 유형을 추가 후 해당 서비스 테스트가 가능합니다.
“지원하지 않는 서비스 입니다” 라고 나와요	상용 서버 연동 환경으로 전환 후 Dialog Kit 상용 배포 하지 않는 경우 해당 안내가 나옵니다. Dialog Kit에서 상용 배포 부탁드립니다.

[첨부 1] metaInfo 상세설명

agent_onCommand (9.5)의 actType : play_media 의 payload 정보는 다음과 같다.

```
"cmdOpt":{
  "channel":0,
  "actOnOther":"pause",
  "metaInfo":{
    "mesg":"안녕하세요",
    "actFeel":"Happy",
    "sentencePattern": "Describe"
  }
}
```

metaInfo 의 구조는 제공하는 `contentType`(서비스의 종류)에 따라 달라지며, 현재 metaInfo 를 제공하는 서비스에 대한 `contentType`은 아래와 같다

서비스	contentType 값
일반 채팅	(없음)
지니뮤직	music
팟캐스트	podcast
라디오	radio
캘린더	calendar
날씨	weather
지역정보	localInfo
위키피디아	Wikipedia
환율	exchangeRate
버스도착안내	busArrivalInfo

서비스	contentType 값
길안내	directions
알람/타이머	* TBD

- metalInfo 구조는 다음과 같다.

[일반 채팅]

- metalInfo

이름	유형	필수	설명
mesg	String	O	TTS 응답 메시지 텍스트
actFeel	String	O	TTS 응답 메시지에 대한 감성 - Happy, Sad, Thanks, Neutral 등
sentencePattern	String	O	문형 종류 - Describe, Yn_question 등

- Payload 예시

```
// "안녕하세요" 발화 응답
{
  "cmdOpt":{
    "channel":0,
    "actOnOther":"pause",
    "metalInfo":{
      "mesg":"안녕하세요",
      "actFeel":"Happy",
      "sentencePattern": "Describe"
    }
  }
}
```

[위키디피아]

- metalInfo

이름	유형	필수	설명
mesg	String	O	TTS 응답 메시지 텍스트
actFeel	String	O	TTS 응답 메시지에 대한 감성 - Happy, Sad, Thanks, Neutral 등
sentencePattern	String	O	문형 종류 - Describe, Yn_question 등
contentType	String	O	"wikipedia"

- Payload 예시

```
// "현빈이 누구야" 발화 응답
{
  "cmdOpt":{
    "channel": 0,
    "actOnOther": "pause",
    "metaInfo": {
      "mesg": "<ktml>위키에서 찾아봤어요.</ktml> <pause=W\"500W\">현빈은  
대한민국의 남자 배우이다.",
      "actFeel": "Neutral",
      "sentencePattern": "Describe",
      "contentType": "wikipedia"
    }
  }
}
```

[지니뮤직]

- metaInfo

이름	유형	필수	설명
contentType	String	O	"music"
infoType	String	O	"text" - 콘텐츠에 대한 상세 정보를 전달하는 방식 - 현재 text 형태만 지원
infoDetail	Object	O	콘텐츠에 대한 상세 정보

이름	유형	필수	설명
infoDetail.title	String	O	노래 제목
infoDetail.artist	String	O	가수 이름
infoDetail.imageurl	String	O	앨범 이미지 URL
infoDetail.duration	Number	O	재생 시간 (단위: 초)

- Payload 예시

```
{
  "cmdOpt":{
    "channel": 101,
    "actOnOther": "stop",
    "setDssStatus": [
      "SU:023"
    ],
    "url": "http://app-
genie.ktcdn.co.kr/ND_O/SONG/MP3/089/382/89382913_192k.mp3?token=eyJ0eXAiOiJKV1Qi
LCJhbGciOiJIUzI1NiJ9.eyJkdXJhdGlvbil6NjAsInBhdGgiOiIvTkRfTy9TT05HL01QMy8wODkvMzgy
Lzg5MzgyOTEzXzE5MmsubXAzliwiaXNzljoiZ3AiLCJleHAiOiJlOTE3NDkxMTEsInBsYXZldGFydCI
6MH0.7XUd-Xh_DHADZvHjNWX3o-xQpDYh4bHW5-GtgAVYOSY",
    "playNotiTime": [
      5,
      60
    ],
    "metaInfo": {
      "contentType": "music",
      "infoType": "text",
      "infoDetail": {
        "title": "고래",
        "artist": "AKMU (악동뮤지션)",
        "imageurl":
"http://image.genie.co.kr/Y/IMAGE/IMG_ALBUM/081/271/853/81271853_1569396403906_1_6
00x600.JPG",
        "duration": 199
      }
    }
  }
}
```

```
}
}
```

[라디오]

- metaInfo

이름	유형	필수	설명
contentType	String	O	"radio"
infoType	String	O	"text" - 콘텐츠에 대한 상세 정보를 전달하는 방식 - 현재 text 형태만 지원
infoDetail	Object	O	콘텐츠에 대한 상세 정보
infoDetail.title	String	O	라디오 채널 이름

- Payload 예시

```
{
  "cmdOpt":{
    "channel": 101,
    "actOnOther": "stop",
    "setDssStatus": [
      "SU:026"
    ],
    "url": "http://tbs.hscdn.com/tbsradio/fm/playlist.m3u8",
    "metaInfo": {
      "contentType": "radio",
      "infoType": "text",
      "infoDetail": {
        "title": "TBS FM"
      }
    }
  }
}
```

[팟캐스트]

- metaInfo

이름	유형	필수	설명
contentType	String	O	"podcast"
infoType	String	O	"text" - 콘텐츠에 대한 상세 정보를 전달하는 방식 - 현재 text 형태만 지원
infoDetail	Object	O	콘텐츠에 대한 상세 정보
infoDetail.title	String	O	팟캐스트 채널 이름
infoDetail.artist	String	O	팟캐스트 채널 제공 주체

- Payload 예시

```
{
  "cmdOpt":{
    "channel": 101,
    "actOnOther": "stop",
    "url": "https://api-ex.podbbang.com/kt/file/12548/23559616",
    "metaInfo": {
      "contentType": "podcast",
      "infoType": "text",
      "infoDetail": {
        "title": "0609 뉴스공장 1-2 부 (김홍걸, 윤형중, 신장식, 장용진, 신유진)",
        "artist": "TBS 김어준의 뉴스공장"
      }
    }
  }
}
```

[지역정보]

- metaInfo

이름	유형	필수	설명
contentType	String	O	"localInfo"

이름	유형	필수	설명
infoType	String	O	"text" - 콘텐츠에 대한 상세 정보를 전달하는 방식 - 현재 text 형태만 지원
infoDetail	Object	O	콘텐츠에 대한 상세 정보
infoDetail.url	String	O	지역정보를 화면으로 보여줄 수 있는 페이지 URL

- Payload 예시

```
{
  "cmdOpt":{
    "channel": 0,
    "actOnOther": "pause",
    "metaInfo": {
      "mesg": "근처 치과 병원을 찾고 계신가요? 이봉호치과의원. 미치과의원.
조은이치과의원. 허우교치과. 미네르반치과의원. 에버그린치과의원. 관련 상세 정보를
연결하신 기가지니 앱으로 전송 했어요.",
      "actFeel": "Neutral",
      "sentencePattern": "Describe",
      "contentType": "localInfo",
      "infoType": "text",
      "infoDetail": {
        "url":
"http://114sms.co.kr:8080/STMS/jsp/genie_lte/genie_lte.jsp?SRV_TYPE=KTH&API_KEY=04a6c
e41955a5d1&CONT_TYPE=1&BRAND=44&GPS_X=126.9079439017859&GPS_Y=37.5619769
324024"
      }
    }
  }
}
```

[길안내]

- metaInfo

이름	유형	필수	설명
contentType	String	O	"directions"
infoType	String	O	"text" - 콘텐츠에 대한 상세 정보를 전달하는 방식 - 현재 text 형태만 지원
infoDetail	Object	O	콘텐츠에 대한 상세 정보
infoDetail.url	String	O	길안내 정보를 화면으로 보여줄 수 있는 페이지 URL

- Payload 예시

```
{
  "cmdOpt":{
    "channel": 0,
    "actOnOther": "pause",
    "metaInfo": {
      "msg": "서울특별시 서초구 태봉로 151 에서 광화문까지 7.6km 이며,
자동차로 현재 24 분 소요 됩니다. 연결된 단말로 네비를 연결합니다.",
      "actFeel": "Neutral",
      "sentencePattern": "Describe",
      "contentType": "directions",
      "infoType": "text",
      "infoDetail": {
        "url":
"https://gdialog.ktipmedia.co.kr/kdialog/ollehmap.jsp?nx=947711.2&ny=1951567.0&dx=953
805.0&dy=1953083.0&stadr=%BC%AD%BF%EF%C6%AF%BA%B0%BD%C3+%B8%B6%C6%F
7%B1%B8+%BF%F9%B5%E5%C4%C5%B7%CE30%B4%D9%B1%E6+22&desadr=%B1%A4%C
8%AD%B9%AE"
      }
    }
  }
}
```

[알람/타이머]

- 추후 제공 예정

[날씨]

- metaInfo

이름	유형	필수	설명
mesg	String	O	TTS 응답 메시지 텍스트
actFeel	String	O	TTS 응답 메시지에 대한 감성 - Happy, Sad, Thanks, Neutral 등
sentencePattern	String	O	문형 종류 - Describe, Yn_question 등
contentType	String	O	"weather"
infoType	String	O	"text" - 콘텐츠에 대한 상세 정보를 전달하는 방식 - 현재 text 형태만 지원
infoDetail	Object	O	콘텐츠에 대한 상세 정보

- Payload 예시

```
{
  "cmdOpt":{
    "channel": 0,
    "actOnOther": "pause",
    "metaInfo": {
      "mesg": "<html>오늘 <pause=WWW'0WWW">t 서초구
태봉로<pause=WWW'0WWW"> 날씨입니다. 무더운 날씨가 될 것 같아요. 그래도 시원하게
하루를 보냈으면 좋겠어요. 기온은 <pause=WWW'0WWW">26 도<pause=WWW'0WWW">
이며, 오늘 최고기온은 <pause=WWW'0WWW">31 도<pause=WWW'0WWW"> 최저기온은
<pause=WWW'0WWW">22 도<pause=WWW'0WWW"> 입니다. 미세먼지 농도가 좋음이고,
초미세먼지 농도가 보통입니다.</html>",
      "actFeel": "Neutral",
      "sentencePattern": "What_question",
      "contentType": "weather",
      "infoType": "text",
      "infoDetail": {
        "type": "today",
        ...
      }
    }
  }
}
```

```
}
```

- 날씨 infoDtail 정보는 [첨부 2] 날씨 infoDetail 구조를 참고한다.

[첨부 2] 날씨 infoDetail 구조

날씨 정보 metaInfo 내 전달되는 infoDetail 에는 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)

- weatherList[]는 총 8 개의 배열로 구성된다.

값	설명
type	"today", 날씨 정보 종류
description	날짜, 일자, 지역 메시지
area	지역
forecast	날씨 예보(맑음/흐림/구름조금 등)
forecastDetail	날씨 예보 내용
icon	날씨 아이콘 코드 (1~22)
temp	현재온도 (°C)
tempHigh	최고온도 (°C)
tempLow	최저온도 (°C)
precipitationChance	강수확률 (%)
precipitation	강수량 (mm)
windSpeed	풍속 (m/s)
windDirection	풍향 코드

값	설명
humidity	습도 (%)
pm10	미세먼지 수치 ($\mu\text{g}/\text{m}^3$)
pm25	초미세먼지 수치 ($\mu\text{g}/\text{m}^3$)
ozone	오존 수치 (ppm)
uv	자외선 수치
weatherList[]	요청 시간 이후의 시간대별(3시간 단위) 날씨 정보 배열
weatherList[n].time	시간대
weatherList[n].tempHigh	예상 최고 온도 ($^{\circ}\text{C}$)
weatherList[n].icon	날씨 아이콘 코드 (1~22)
weatherList[n].forecast	날씨 예보 코드
weatherList[n].precipitationChance	예상 강수확률 (%)
updated.time	오늘 시간대별 날씨 업데이트 시간 (MM/DDHH:mm)
updated.day	오늘 상세날씨 업데이트 시간
updated.pm	미세먼지/초미세먼지 업데이트 시간
updated.dust	황사 업데이트 시간
updated.ozone	오존 업데이트 시간
updated.uv	자외선 업데이트 시간

- 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": "21 시",  
  "tempHigh": 18,  
  "icon": 2,  
  "forecast": "구름조금",  
  "precipitationChance": 0  
},{  
  "time": "내일 0 시",  
  "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)

- 기본적으로 오전 날씨를 보여준다. (오전 날씨 데이터는 9 시 기준)
- 오후 날씨 정보는 weatherPm{}으로 구성된다. (오후 날씨 데이터는 15 시 기준)
- weatherList[]는 총 8 개의 배열로 구성된다.

값	설명
type	"tomorrow", 날씨 정보 종류
description	날짜, 일자, 지역 메시지
area	지역
forecast	날씨 예보(맑음/흐림/구름조금 등)
forecastDetail	날씨 예보 내용
icon	날씨 아이콘 코드 (1~22)
temp	내일 오전 온도 (°C)
precipitationChance	강수확률 (%)
windSpeed	풍속 (m/s)
windDirection	풍향 코드
humidity	습도 (%)
pm10	미세먼지 수치 (μg/m³)
pm25	초미세먼지 수치 (μg/m³)
ozone	오존 수치 (ppm)
uv	자외선 수치
weatherPm {}	오후(15시 기준) 날씨 정보 객체

값	설명
weatherPm.forecast	오후 날씨 예보(맑음/흐림/구름조금 등)
weatherPm.icon	오후 날씨 아이콘 코드 (1~22)
weatherPm.temp	오후 온도 (°C)
weatherPm.precipitationChance	오후 강수확률 (%)
weatherPm.windSpeed	오후 풍속 (m/s)
weatherPm.windDirection	오후 풍향 코드
weatherPm.humidity	오후 습도 (%)
weatherPm.pm10	오후 미세먼지 수치 ($\mu\text{g}/\text{m}^3$)
weatherPm.pm25	오후 초미세먼지 수치 ($\mu\text{g}/\text{m}^3$)
weatherPm.ozone	오후 오존 수치 (ppm)
weatherPm.uv	오후 자외선 수치
weatherList[]	시간대별(3시간 단위) 날씨 정보 배열
weatherList[n].time	시간대
weatherList[n].tempHigh	예상 최고 온도 (°C)
weatherList[n].icon	날씨 아이콘 코드 (1~22)
weatherList[n].forecast	날씨 예보
weatherList[n].precipitationChance	예상 강수확률 (%)
updated.time	내일 시간대별 날씨 업데이트 시간 (MM/DDHH:mm)
updated.day	내일 상세날씨 업데이트 시간
updated.pm	미세먼지/초미세먼지 업데이트 시간
updated.dust	황사 업데이트 시간
updated.ozone	오존 업데이트 시간

값	설명
updated.uv	자외선 업데이트 시간

- 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": "0 시",  
  "tempHigh": 15,  
  "icon": 1,  
  "forecast": "맑음",  
  "precipitationChance": 0  
},{  
  "time": "3 시",  
  "tempHigh": 13,  
  "icon": 1,  
  "forecast": "맑음",  
  "precipitationChance": 0  
},{  
  "time": "6 시",  
  "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)

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

값	설명
type	"day", 날씨 정보 종류
description	날짜, 일자, 지역 메시지
area	지역
forecast	날씨 예보(맑음/흐림/구름조금 등)
forecastDetail	날씨 예보 내용
icon	날씨 아이콘 코드 (1~22)
temp	내일 오전 온도 (°C)
precipitationChance	강수확률 (%)
windSpeed	풍속 (m/s)
windDirection	풍향 코드
humidity	습도 (%)
pm10	미세먼지 수치 ($\mu\text{g}/\text{m}^3$)
pm25	초미세먼지 수치 ($\mu\text{g}/\text{m}^3$)
ozone	오존 수치 (ppm)
uv	자외선 수치
weatherPm {}	오후(15시 기준) 날씨 정보 객체
weatherPm.forecast	오후 날씨 예보(맑음/흐림/구름조금 등)
weatherPm.icon	오후 날씨 아이콘 코드 (1~22)
weatherPm.temp	오후 온도 (°C)
weatherPm.precipitationChance	오후 강수확률 (%)
weatherPm.windSpeed	오후 풍속 (m/s)
weatherPm.windDirection	오후 풍향 코드

값	설명
weatherPm.humidity	오후 습도 (%)
weatherPm.pm10	오후 미세먼지 수치 ($\mu\text{g}/\text{m}^3$)
weatherPm.pm25	오후 초미세먼지 수치 ($\mu\text{g}/\text{m}^3$)
weatherPm.ozone	오후 오존 수치 (ppm)
weatherPm.uv	오후 자외선 수치

- 예시

```
{
  "type": "day",
  "description": "05 월 28 일 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)

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

값	설명
type	"day", 날씨 정보 종류
description	날짜, 일자, 지역 메시지
area	지역
forecast	날씨 예보(맑음/흐림/구름조금 등)
forecastDetail	날씨 예보 내용
icon	날씨 아이콘 코드 (1~22)
tempHigh	토요일 최고온도 (°C)
tempLow	토요일 최저온도 (°C)
precipitationChance	토요일 강수확률 (%)
weatherSun {}	주말[토요일, 일요일] 날씨 정보 객체
weatherSun.forecast	일요일 날씨 예보(맑음/흐림/구름조금 등)
weatherSun.icon	일요일 날씨 아이콘 코드 (1~22)

값	설명
weatherSun.tempHigh	일요일 최고온도 (°C)
weatherSun.tempLow	일요일 최저온도 (°C)
weatherSun.precipitationChance	일요일 강수확률 (%)
weatherSun.windSpeed	일요일 예상 풍속 (m/s)

- 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)

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

값	설명
type	"day", 날씨 정보 종류
description	날짜, 일자, 지역 메시지
area	지역
weatherList[]	일별 날씨 정보 배열
weatherList[n].time	일자
weatherList[n].tempHigh	예상 최고온도 (°C)
weatherList[n].tempLow	예상 최저온도 (°C)
weatherList[n].icon	날씨 아이콘 코드 (1~22)
weatherList[n].forecast	날씨 예보 코드
weatherList[n].precipitationChance	예상 강수확률 (%)

- 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"
  }
}

```

[날씨 아이콘 코드]

코드	설명
1	맑음
2	구름 조금
3	흐림
4	비
5	눈

코드	설명
6	눈비
7	소나기
9	안개
10	뇌우
11	차차 흐려짐
13	흐려져 비
14	흐려져 눈
15	흐려져 눈비
16	흐린 후 갸
18	비온 후 갸
19	눈온 후 갸
20	눈비 후 갸
21	구름 많음

[풍향 코드]

코드	설명
0	바람 없음
1	북풍
2	북동풍
3	동풍
4	남동풍
5	남풍
6	남서풍

코드	설명
7	서풍
8	북서풍

끝.