

GiGA Genie Mobile Plugin 2.0

Developer Guide

for Android

융합기술원 Service연구소



목차

1. 개요	4
1.1. 문서 연혁	4
1.2. 목적	5
1.3. 적용범위	5
1.4. 약어 정리	5
2. 일반 사항	6
2.1. AI 비서 라이브러리 프로젝트 적용 방법	6
2.2. 주의 사항	9
2.2.1. 난독화 (Proguard)	9
2.3. 지원 버전	9
3. API description	10
3.1. API 목록	10
3.2. G-Plugin Common API	13
3.2.1. G-Plugin 라이브러리 생성	13
3.2.2. G-Plugin 서비스 시작	14
3.2.3. G-Plugin 서비스 종료	14
3.2.4. G-Plugin 서비스 Callback 등록	15
3.2.5. G-Plugin 버전정보 조회	18
3.2.6. 위치정보 업데이트	18
3.2.7. 라이브러리 내부 로그 보기 설정	19
3.3. G-Plugin Main Service API	19

3.3.1.	G-Plugin Main Service Flow.....	19
3.3.2.	G-Plugin 호출어 API.....	23
3.3.3.	Command 요청.....	25
3.3.4.	Command 취소.....	26
3.3.5.	Callback 및 미디어 재생 상태 업데이트.....	27
3.4.	G-Plugin 단위서비스 API.....	32
3.4.1.	TTS API.....	32
4.	개발 참고 사항 및 이슈 사항.....	33
4.1.	예외처리(TBD).....	33
4.2.	FAQ.....	33
4.2.1.	“CFK_FAILED_CONN” 에러코드 발생하는 경우.....	33
[첨부 1]	G-Plugin EventCode.....	34
[첨부 2]	G-Plugin ErrorCause.....	35
[첨부 3]	KWS ErrorCause.....	35
[첨부 4]	음성 및 대화 Life-cycle.....	36

1. 개요

1.1. 문서 연혁

변경일	변경사항	작성자
2018.12.05	- 최초	KT
2018.12.14	- API 재분류	KT
2019.01.03	- GPlugin init API 수정 (3.2.1) - 위치정보 업데이트 API 설명 및 예시 추가 (3.2.10) - TTS Stream 요청 API 수정 (3.4.3.1) - "미디어 콘텐츠 UI 구성 방법" 항목 추가 (첨부5)	KT
2019.02.12	- Callback 등록 : webview_url 항목 및 설명 추가 (3.3.4) - Callback 등록 : dss_xml 항목을 dialog_response로 변경 (3.3.4) - 2.0 미지원 API 삭제	KT
2019.05.03	- EventCode 현행화 (첨부1)	KT
2019.05.17	- EventCause 추가 (첨부2)	KT
2019.06.17	- getTTS 데이터 포맷 설명 추가	KT
2019.06.24	- 2.0 기준 API 재분류	KT
2019.06.27	- KWS 에러 설명 추가	KT
2019.07.11	- 외부 라이브러리 적용 방법 설명 추가 (build.gradle) - setServerInfo API 추가	KT
2019.08.19	- Instant Run 옵션 관련 내용 추가 - 호출어 사용시 Main Service Flow 변경 (3rd party가 음성인식 시작 API를 호출해야 함)	KT
2019.10.21	- 빌드 버전 수정 (26 → 28) - dialogResponse의 JSON 포맷 설명 수정	KT
2019.11.04	- Callback 등록 : media_stream_meta 항목 및 설명 추가 (3.3.5) - stopVoiceCmd → cancelVoiceCmd로 API명 변경 (3.3.4)	KT
2019.11.22	- KWS 관련 EventCode 추가 (첨부1)	KT

1.2. 목적

본 문서는 (주)케이티(이하 KT)의 GiGA Genie AI 서비스를 모바일 앱에서 이용할 수 있도록 제공하는 GiGA Genie Mobile Plugin(이하, G-Plugin) 라이브러리를 3rd Party Application 에 연동 방법에 대해 기술한다.

1.3. 적용범위

- 3rd Party Application 에서 AI 비서 라이브러리를 연동하는 방법을 설명한다.
- 3rd Party Application 의 빌드 버전에 따라 일부 기능에 제약이 있을 수 있으며, 경우에 따라 별도 빌드된 G-Plugin 라이브러리를 제공 가능하다.
- 지원 버전은 2.3 항목에 작성하였다.
- 내용은 KT 서비스 제공 계획의 변경 및 추가에 따라 추후 보완 및 변경 될 수 있다.

1.4. 약어 정리

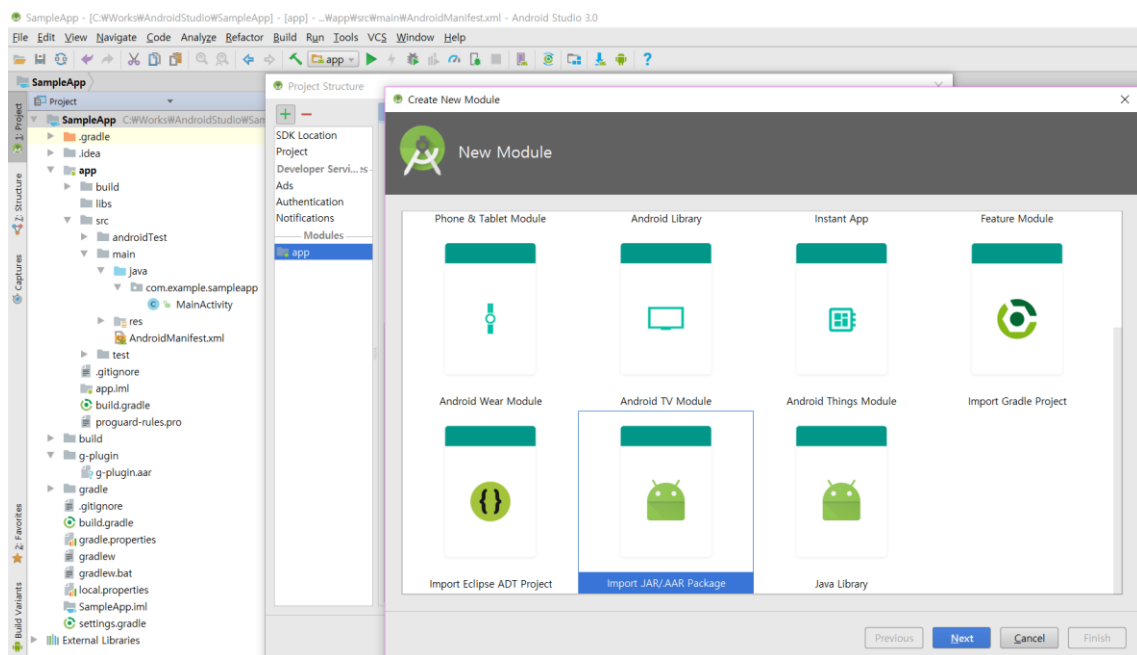
- G-Plugin – GiGA Genie Mobile Plugin 의 약어
- KWS – Keyword Spotting 의 약어

2. 일반 사항

3rd Party Application 이 G-Plugin 과 연동하기 위한 API 사용 방법을 설명한다.

2.1. AI 비서 라이브러리 프로젝트 적용 방법

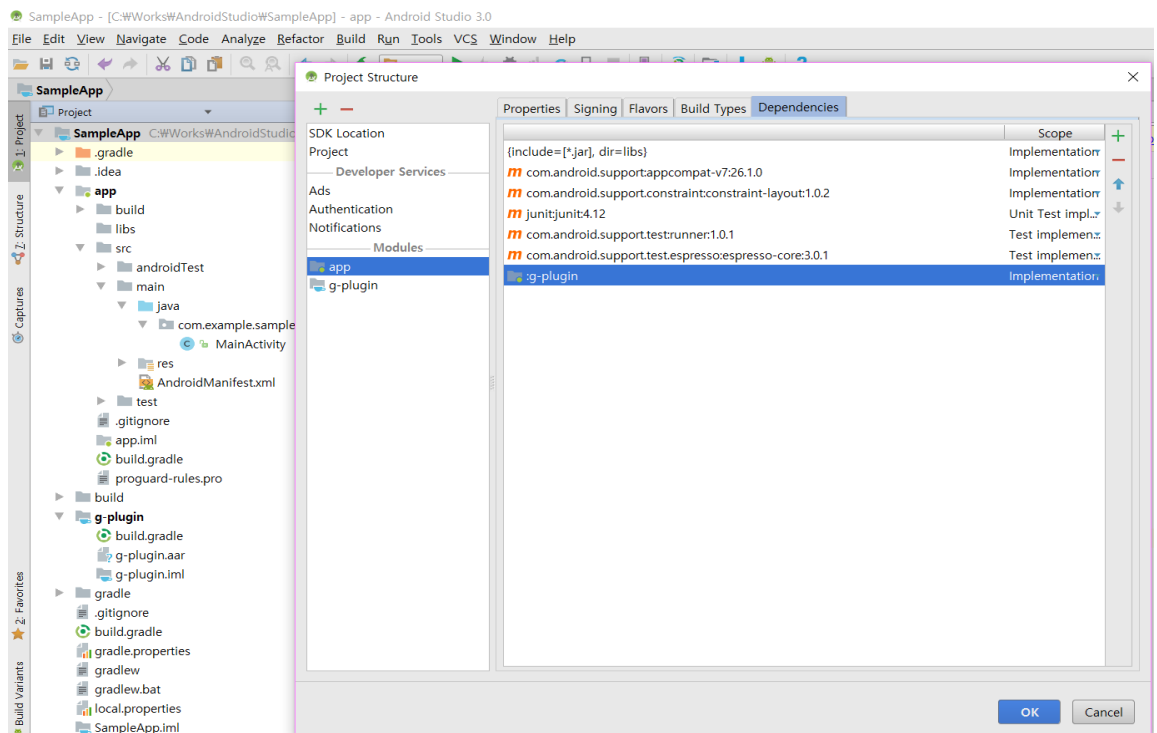
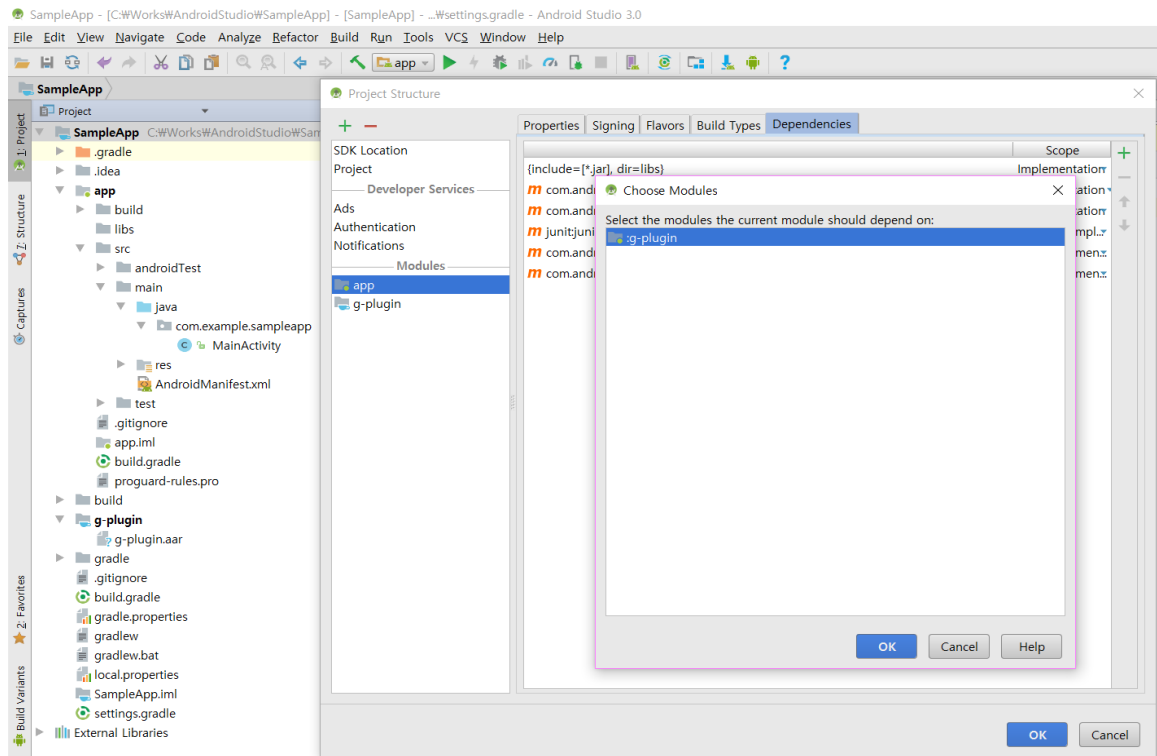
- 1) File > Project Structure > Create New Module > Import .JAR/.AAR Package 를 통해 배포된 g-plugin.aar 라이브러리를 import 한다. (import 후 g-plugin 라이브러리 사이즈가 0 인 경우 g-plugin.aar 라이브러리를 수동으로 복사하여 덮어쓰기 한다.)



- 2) settings.gradle 을 다음과 같이 수정하고 Refresh all gradle projects 를 수행한다.

```
include ':app', ':g-plugin'
```

- 3) G-plugin.aar 라이브러리가 import 된 폴더(g-plugin₩)에 build.gradle 파일을 복사해 넣는다.
- 4) File > Project Structure > Modules 의 app 선택 > Dependencies 탭 선택 후 + 클릭 > 3. Module dependency 선택 > g-plugin 선택하여 g-plugin 모듈을 추가한다.



5) 외부 라이브러리 사용을 위해 app\build.gradle 파일을 다음과 같이 수정한다.

```
android{
```

```

compileOptions{
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}
}

dependencies {
    ...
    // JSON
    api 'com.google.code.gson:gson:2.8.2'

    implementation 'io.reactivex.rxjava2:rxjava:2.1.14'
    implementation 'io.reactivex.rxjava2:rxandroid:2.0.2'

    // REST
    implementation 'com.squareup.retrofit2:retrofit:2.3.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.3.0'
    implementation('com.squareup.retrofit2:converter-simplexml:2.3.0') {
        exclude group: 'xpp3', module: 'xpp3'
        exclude group: 'stax', module: 'stax-api'
        exclude group: 'stax', module: 'stax'
    }

    // gRPC
    implementation 'io.grpc:grpc-okhttp:1.11.0'
    implementation 'io.grpc:grpc-protobuf-lite:1.11.0'
    implementation 'io.grpc:grpc-stub:1.11.0'

    implementation 'javax.annotation:javax.annotation-api:1.2'
}

```

- 6) 서비스 제공자가 라이브러리를 사용하기 위해서 AndroidManifest.xml 파일에 다음과 같은 Permission 을 추가한다.

```

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

```



```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

2.2. 주의 사항

2.2.1. 난독화 (Proguard)

G-Plugin 라이브러리는 안드로이드 Proguard 를 이용하여 난독화되어 배포되므로 3rd party 앱 난독화 적용 시 추가로 난독화 되지 않도록 한다.

2.2.2. Instant Run

G-Plugin 라이브러리는 Android Studio 의 Instant Run off 모드를 지원한다. (Android Studio default 값은 off 이다.)

2.3. 지원 버전

G-Plugin 라이브러리는 아래 버전으로 빌드 되었다.

- compileSdkVersion : 28
- targetSdkVersion : 28

3. API description

3.1. API 목록

	APIs	설명
GPlugin	init (String clientId , String clientKey , String clientSecret , String userId)	G-Plugin 라이브러리 생성 - G-Plugin 라이브러리를 사용하기 전에 반드시 호출되어야 한다. - ClientId, ClientKey, ClientSecret 은 기가지니 포탈에서 발급 - userId 는 3rd party 가 생성
void	startGPlugin()	G-Plugin 서비스 시작 - 음성인식 + AI 서비스 이용 가능
void	stopGPlugin()	G-Plugin 서비스 종료
void	setGPluginListener(GPluginListener listener) GPluginListener - public void onError(int error) - public void onEvent(Bundle event) - public void onCommand (JSONObject result)	G-Plugin 으로부터 음성명령 결과나 이벤트 등을 수신하기 위한 callback 등록 - onCommand : 음성 또는 텍스트 인식 해석 결과를(json object)를 수신하기 위한 Callback 등록
String	getGPluginVersion()	G-Plugin 라이브러리 버전정보 조회
void	updateLocInfo(String cAddress, String dAddress, String aAddress)	현재 위치, 도착지(목적지) 및 경유지 주소를 업데이트 한다. - startGPlugin() 호출 후 최초 한번 위치정보를 업데이트 하고 그 이후 변경 시마다 또는 주기적으로 업데이트 해야 한다. (위치 정보 서비스 확장을 위하여 사용, 현재는 날씨 정보에 사용 중)

		(예시) cAddress : "서울 서초구 우면동" 사용하지 않는 인자 : null
void	setLogVisible(boolean flag)	라이브러리 내부 로그 보기 설정
int	getKwsId	현재 G-Plugin 음성인식 호출어 조회 - 0(기가지니), 1(지니야)
void	setKwsId(int id)	G-Plugin 음성인식 호출어 변경 - 0(기가지니), 1(지니야)
void	enableKws(boolean on)	G-Plugin 호출어 인식 사용여부 설정 - true : 호출어 인식 활성화 - false : 호출어 인식 비활성화
boolean	checkKwsAvailable()	G-Plugin 호출어 인식 사용여부 조회 - true : 호출어 인식 사용 - false : 호출어 인식 미사용
void	startKws()	G-Plugin 호출어 인식 시작
void	stopKws()	G-Plugin 호출어 인식 종료
void	voiceCommand()	음성 명령으로 Service 요청
void	textCommand(String msg, String type)	텍스트 명령으로 Service 요청
void	setMediaStatus(Int channel, String status, Int playtime)	단말에서 미디어를 제어(재생시작, 일시정지,종료)하는 경우 해당 동작을 서버로 알림
void	serviceLogin(String appld, String type)	서비스 로그인 페이지 요청 (예, 지니뮤직 등)

void	cancelVoiceCmd()	G-Plugin 음성인식 종료
void	getTTS(string msg)	G-Plugin TTS Stream 요청

3.2. G-Plugin Common API

3.2.1. G-Plugin 라이브러리 생성

3.2.1.1. `init(Context context, String clientId, String clientKey, String clientSecret, String userId)`

3rd Party Application 이 AI 비서와 연동하기 위해서 G-Plugin 을 생성한다. 3rd Party Application 이 G-Plugin 라이브러리를 사용하지 전에 반드시 호출하여야 한다.

Parameter 정의

Column Name	Type	설명
context	Android.content.Context	Context
clientId	String	기가지니 포탈 또는 KT 담당자를 통해 발급
clientKey	String	기가지니 포탈 또는 KT 담당자를 통해 발급
clientSecret	String	기가지니 포탈 또는 KT 담당자를 통해 발급
userId	String	사용자 식별을 위한 정보로 3 rd party Application 에서 생성하여 입력한다. VOC 대응을 위한 식별자이며, 만약 이 정보가 없는 경우 단말의 MAC 주소를 사용자 식별 정보로 사용하게 된다. (44 자리 이내로 권장)

3.2.1.2. init 호출 예시

```
import com.kt.gigagenie.gplugin;

String client_id = "YOUR_CLIENT_ID";
String client_key = "YOUR_CLIENT_KEY";
String client_secret = "YOUR_CLIENT_SECRET";
mGplugin = GPlugin.init(mContext, client_id, client_key, client_secret, "T-Car");
```

3.2.2. G-Plugin 서버 정보 변경

3.2.2.1. setServerInfo(String serverIP, String grpcPort, String restPort)

3rd Party Application 이 G-Plugin 라이브러리에서 통신할 서버의 정보를 설정한다. 해당 API 를 호출하지 않으면 Default 는 상용서버로 연동하게 된다. 연동 서버 정보를 변경하기 위해서는 Init 하기 전에 해당 API 를 먼저 호출하여야 한다.

Parameter 정의

Column Name	Type	설명
serverIP	String	연동 서버 IP
grpcPort	String	연동 서버 gRPC Port
restPort	String	연동 서버 rest Port

3.2.2.2. setServerInfo 사용 예시

```
GPlugin.setServerInfo("inside.gigagenie.ai", "50051", "9080");
```

3.2.3. G-Plugin 서비스 시작

3.2.3.1. startGPlugin()

3rd Party Application 이 G-Plugin 라이브러리를 사용하기 위해 G-Plugin 서비스를 시작한다.

Parameter 정의

Column Name	Type	설명

3.2.3.2. startGPlugin 사용 예시

```
mGplugin.startGPlugin();
```

최초 startGPlugin() 을 호출하게 되면 호출어를 사용 하는 경우에는 startkws 를 라이브러리 내부에서 실행을 한다.

3.2.4. G-Plugin 서비스 종료

3.2.4.1. stopGPlugin()

3rd Party Application 이 G-Plugin 라이브러리를 사용을 종료하기 위해 G-Plugin 서비스 종료한다.

Parameter 정의

Column Name	Type	설명

3.2.4.2. stopGPlugin 사용 예시

```
mGplugin.stopGPlugin();
```

3.2.5. G-Plugin 서비스 Callback 등록

3.2.5.1. setGPluginListener(GPluginListener listener)

3rd Party Application 이 G-Plugin 라이브러리로부터 Notification 이나 이벤트 등을 수신하기 위한 callback 을 등록한다.

1) onError(int error)

- G-Plugin 에러 전달. UI 작업을 쉽게 하기 위해서 Error 도 이벤트로 전달한다.

2) onCommand(String result)

- 음성 또는 텍스트 인식 해석 결과 전달
- 전달받은 actionType 종류에 따라 아래와 같은 시나리오로 처리하면 된다.
 - dialog_response : 대화서버 응답을 파싱하여 서비스 앱의 flow 에 맞도록 처리
 - media_stream : 전달받은 음성 stream 을 재생
 - media_url : 전달받은 url 을 통해 미디어 재생.
 - webview_url : 전달받은 url 을 웹뷰에서 실행.
 - start_voice : 음성인식(voiceCommand) 시작 요청.

3) onEvent(Bundle event) : [첨부 1] G-Plugin EventCode 참고

- 음성인식(음성명령 시작/종료/에러), TTS(재생완료/취소) 이벤트 전달

- 전달받은 이벤트 코드를 수신하여, UI 및 음성인식 재시작 혹은 음성 명령어 모드를 동작시킨다.

4) Parameter 정의

Column Name	Type	설명
Listener	GPluginListener	G-Plugin callback 수신

3.2.5.2. setGPluginListener 등록 예시

```
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.kt.gigagenie.gplugin;
import com.kt.gigagenie.gpluginlistener;
import com.kt.gigagenie.DSSRespMessage;

mGplugin.setGPluginListener(new GPluginListener() {

    @Override
    public void onError(int error) {
        // TODO Auto-generated method stub
        ...
    }

    @Override
    public void onCommand(JSONObject result) {
        // TODO: need to format of result

        String actionType = null;
        String value = null;

        JSONObject jobject = null;
        try {
            jobject = result;
            actionType = jobject.getString("actionType");
            value = jobject.getString("value");
        } catch (JSONException e) {
            e.printStackTrace();
        }

        if ("dialog_response".equals(actionType)) {
            String messag = value;
            tvCmd.setText(messag);
        } else if ("media_stream".equals(actionType)) {
            String stringRaw = value;
            byte[] decodedString = Base64.decode(stringRaw, Base64.DEFAULT);
            bytToWavPlay(decodedString);
            .....
        } else if ("media_url".equals(actionType)) {
            String media_url = value;
        }
    }
});
```



```

        PlayMedia(media_url);
        .....
    }
    ...
}

@Override
public void onEvent(Bundle results) {
    String event = null;
    String data = null;
    if(results != null) {
        event = results.getString("eventCode");
        data = results.getString("data");
    }

    if(event != null && !event.isEmpty()) {
        if(event.equals("VOICE_STARTED")) {
            // 음성인식 시작
            ...
        }
    }
}

```

```

public class DSSRespMessage
{
    private String userWord; // 사용자 발화내용(음성인식 결과 Text)
    private String sysAct;   // "0"(Specify) : 추가적인 정보 요청
                           // "1"(Confirm) : 사용자 의도 확인
                           // "100"(All) : 가능한 액션 중 사용자의 선택을 유도
    private String sPattern; //
    private String msg;      // 대화서버 응답 메시지
    private String aFeel;
    private String actGroup; // 서비스 도메인(비서, 기본, 홈IoT, 지니뮤직, Navi 등)
    private String actType;  // 응답 결과 (목적지 검색, 경유지 추가, ...)
    private String srchWord; // 검색 질의어
    private String serviceId; // 서비스 연결 ID
    private String isMultiAction;

    ...
}

```

3.2.6. G-Plugin 버전정보 조회

3.2.6.1. getGPluginVersion()

3rd Party Application 이 현재 사용중인 G-Plugin 라이브러리 버전정보를 조회한다.

3.2.6.2. getGPluginVersion 사용 예시

```
String gpluginVer = mGplugin.getGPluginVersion();
```

3.2.7. 위치정보 업데이트

3.2.7.1. updateLocInfo(String cAddress, String dAddress, String aAddress)

3rd Party Application 이 현재 위치 정보 및 도착지(목적지) 주소 정보를 업데이트 한다. 이 정보는 날씨 서비스, 내비게이션 서비스 등 위치 기반의 서비스를 하는 경우 활용된다.

G-Plugin 서비스 시작 시(startGPlugin() 호출 후) 최초 한번 위치 정보를 업데이트하고 그 이후에는 위치정보/목적지가 변경될 때마다 또는 주기적으로(App 에서 필요시) 위치정보를 업데이트 해야 한다.

※ 현재 원내비에서만 사용 가능하도록 라이선스 계약됨.

Parameter 정의

Column Name	Type	설명
cAddress	String	현재 위치 주소 - '서울특별시 서초구 우면동' 과 같은 양식으로 입력
dAddress	String	도착지 주소
aAddress	String	경유지 주소

3.2.7.2. updateLocInfo 사용 예시

```
mGplugin.updateLocInfo("서울 서초구 우면동", "경기 성남시 분당구", null);
```

※ 도착지(목적지) 주소가 없는 경우 null 로 설정하면 된다.

3.2.8. 라이브러리 내부 로그 보기 설정

3.2.8.1. setLogVisible(boolean flag)

라이브러리 내부 API 로그를 보기 위한 설정

Parameter 정의

Column Name	Type	설명
flag	boolean	내부 로그 보기 설정 true : visible false : invisible

3.2.8.2. setLogVisible 호출 예시

```
mGplugin.setLogVisible(true); // 내부 로그 보기 설정
```

3.3. G-Plugin Main Service API

3.3.1. G-Plugin Main Service Flow

3rd Party Application 이 G-Plugin 서비스를 사용하기 위해서는 먼저 인증 및 초기화(init API 호출)작업을 통해 Gplugin 객체를 생성하고 startGplugin 을 호출해야 한다. GiGA Genie AI 서비스를 요청하는 방법은 "3.3.1.1 G-Plugin Command 요청 Flow" 에서 설명한다. G-Plugin 은 GiGA Genie AI 서비스에서 해석된 내용을 콜백으로 리턴해주며, 3rd party 에서는 onCommand 콜백으로 전달받은 액션을 수행해야 한다. 해당 시나리오에 대한 Flow 는 "3.3.1.2 G-Plugin Service 제어 Flow"에서 설명한다.

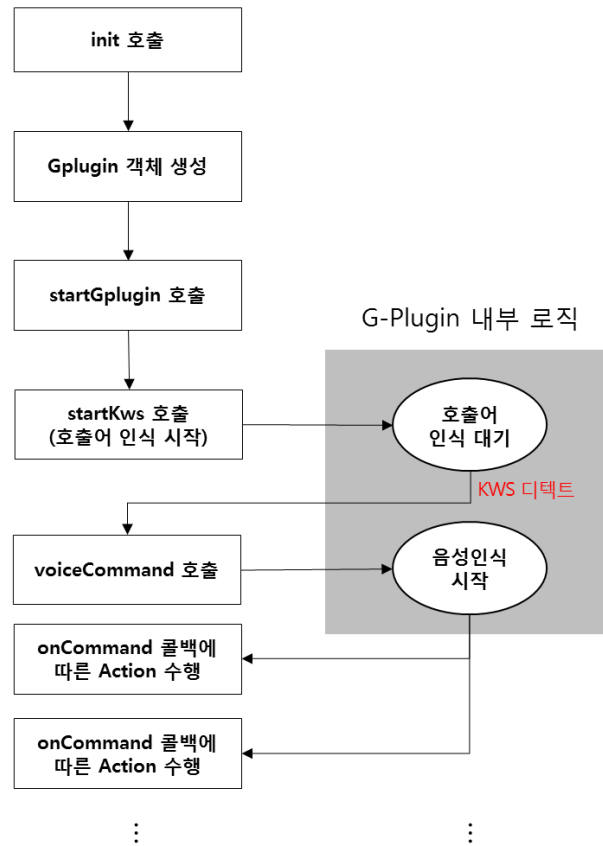
3.3.1.1. G-Plugin Command 요청 Flow

3rd Party Application이 G-Plugin을 통해 GiGA Genie AI 서비스를 요청하기 위해서는 아래 2가지 방법을 사용할 수 있다. 호출어 인식 및 음성인식을 위한 마이크 제어 권한은 G-Plugin이 가지고 있다.

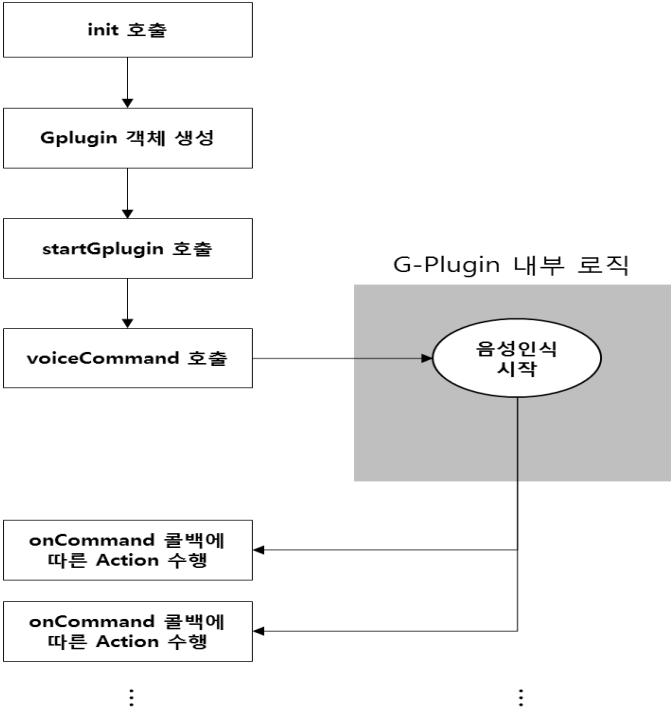
1) startKws API를 호출하여 호출어("기가지니" 또는 "지니야")인식 대기 상태를 요청하고, 이 상태에서 호출어가 인식된 경우(호출어 인식 이벤트 수신), voiceCommand API를 호출하면 음성인식을 시작하게 된다.

2) voiceCommand API를 호출한 경우 G-Plugin에서 바로 음성인식을 시작하게 된다.

KWS(호출어) 디렉트를 통해 음성인식을 시작하는 Flow

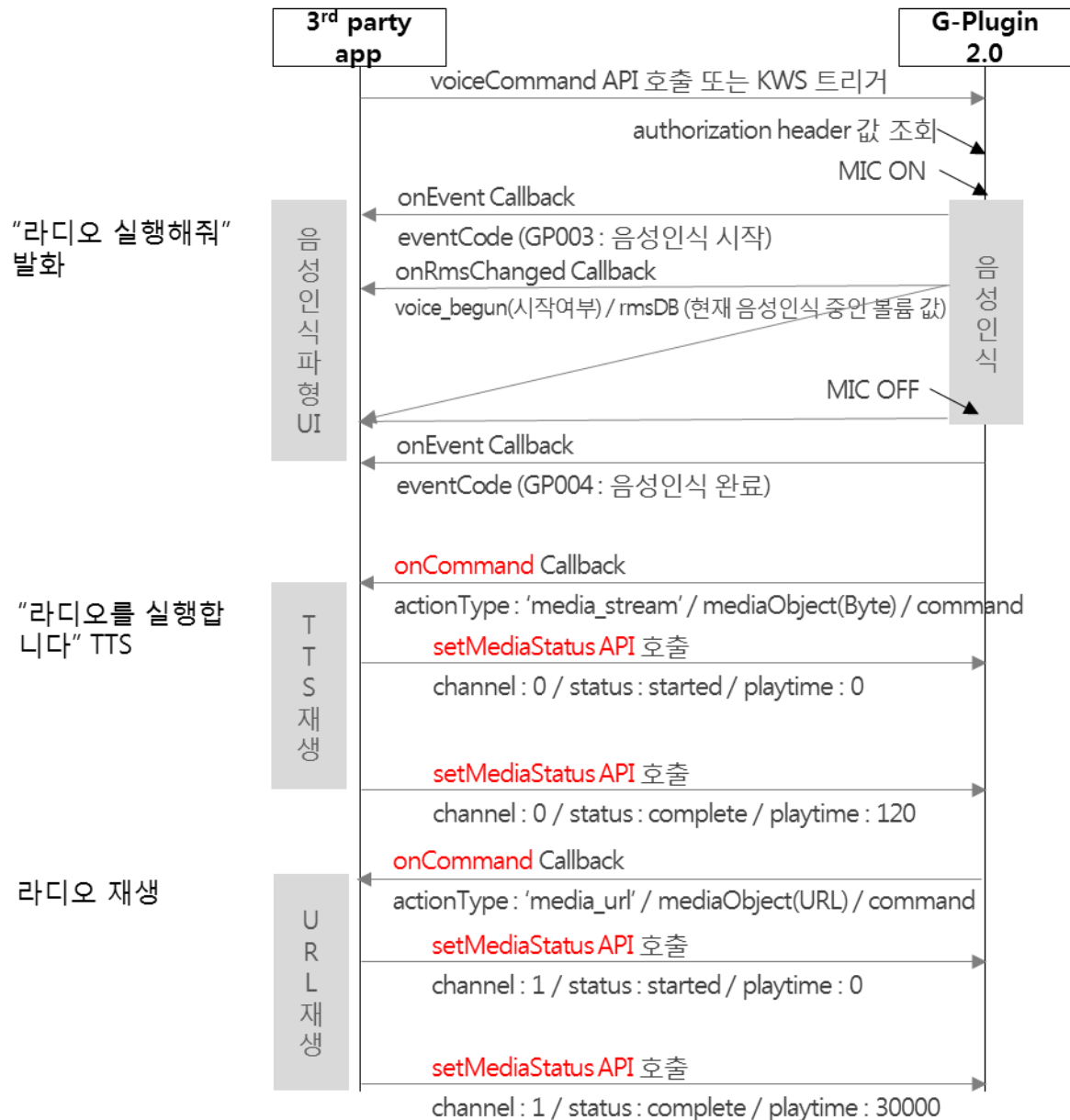


3rd party app이 음성인식을
직접 요청하여 시작하는 Flow



3.3.1.2. G-Plugin Service 제어 Flow

GiGA Genie AI 서비스를 요청한 이후 onCommand 콜백으로 전달받은 액션을 수행해야 한다. 또한 3rd party application이 미디어를 제어하는 경우(재생, 일시정지, 중단, 완료 등) G-Plugin으로 현재 미디어 재생 상태를 G-Plugin으로 전달해주어야 다음 Command를 정상적으로 콜백받을 수 있다. Callback 종류에 따른 시나리오 및 각 옵션에 대한 설명은 "3.3.4. Callback 및 미디어 재생 상태 업데이트"에서 설명한다.



3.3.2. G-Plugin 호출어 API

3.3.2.1. 현재 호출어 조회

3.3.2.1.1. getKwsId()

3rd Party Application 이 G-Plugin 을 통해 현재 호출어를 조회한다.

호출어 ID : 0(기가지니), 1(지니야)

- default 호출어 : 지니야(1)

Column Name	Type	설명

3.3.2.1.2. getKwsId 사용 예시

```
int kwsId = mGplugin.getKwsId();
```

3.3.2.2. 호출어 변경

3.3.2.2.1. setKwsId(int id)

3rd Party Application 이 G-Plugin 을 통해 호출어를 변경한다.

Column Name	Type	설명
id	int	호출어 변경 0 : 기가지니 1 : 지니야

3.3.2.2.2. setKwsId 사용 예시

```
mGplugin.setKwsId(1); // 호출어가 지니야로 설정됨
```

3.3.2.3. 호출어 인식 사용여부 설정

3.3.2.3.1. enableKws(boolean on)

3rd Party Application 이 G-Plugin 을 통해 호출어 인식 사용여부를 설정한다.

Column Name	Type	설명
on	boolean	호출어 인식 사용 여부 설정 - true : 호출어 인식 활성화 - false : 호출어 인식 비활성화

3.3.2.3.2. enableKws 사용 예시

```
mGplugin.enableKws(true);
```

3.3.2.4. 호출어 인식 사용여부 조회

3.3.2.4.1. checkKwsAvailable()

3rd Party Application 이 G-Plugin 을 통해 호출어 사용여부를 조회한다.

(enableKws 에 의해 설정 값을 조회하며 true(사용), false(미사용)으로 리턴된다.)

Column Name	Type	설명

3.3.2.4.2. checkKwsAvailable 사용 예시

```
boolean kwsUse = mGplugin.checkKwsAvailable();
```

3.3.2.5. 호출어 인식 시작

3.3.2.5.1. startKws()

3rd Party Application 이 G-Plugin 을 통해 호출어 인식을 시작한다.

Column Name	Type	설명

3.3.2.5.2. startKws 사용 예시

```
mGplugin.startKws ();
```

3.3.2.6. 호출어 인식 종료

3.3.2.6.1. stopKws()

3rd Party Application 이 G-Plugin 을 통해 호출어 인식을 종료한다.

Column Name	Type	설명

3.3.2.6.2. startKws 사용 예시

```
mGplugin.stopKws ();
```

3.3.3. Command 요청

G-Plugin 으로 Command 를 요청하기 위해서 아래 3 가지 방법을 사용할 수 있다.

1. KWS(호출어) 디텍트되어 G-Plugin 내부에서 voiceCommand 함수 호출하여 음성인식 시작
2. 3rd party application 이 직접 voiceCommand 함수 호출하여 음성인식 시작
3. 3rd party application 이 직접 텍스트 명령어를 textCommand 함수로 호출

3.3.3.1. voiceCommand()

3rd Party Application 이 G-Plugin 을 통해 음성인식을 시작한다.

해당 API 사용 시 호출어가 인식된 이후 동작과 동일하게 동작하게 된다.

결과는 onCommand() callback 을 통해 전달된다.

Column Name	Type	설명

3.3.3.2. voiceCommand 사용 예시

```
mGplugin.voiceCommand();
```

3.3.3.3. textCommand(String msg, String type)

3rd Party Application 이 G-Plugin 을 통해 텍스트 명령을 요청한다.

결과는 onCommand() callback 을 통해 전달된다.

Column Name	Type	설명
msg	String	명령 텍스트를 입력한다.
type	String	명령 텍스트의 종류 - query : 텍스트 쿼리 전송 - event : 이벤트 전송

3.3.3.4. textCommand 사용 예시

```
mGplugin.textCommand("오늘 날씨 알려줘", "query"); // 텍스트 쿼리 전송  
mGplugin.textCommand("ED:D101E200", "event"); // 앱 구동 이벤트 전송
```

3.3.4. Command 취소

3.3.4.1. 음성인식 종료

3.3.4.1.1. cancelVoiceCmd()

3rd Party Application 이 voiceCommand 요청으로 진행중인 음성인식을 종료한다.

Column Name	Type	설명

3.3.4.1.2. cancelVoiceCmd 사용 예시

```
mGplugin.cancelVoiceCmd();
```

3.3.5. Callback 및 미디어 재생 상태 업데이트

3.3.5.1. G-plugin Callback

G-plugin 은 3rd Party Application 으로 음성 또는 텍스트 인식 해석 결과를 onCommand Callback 으로 전달한다. 3rd Party Application 은 onCommand Callback 의 actionType 에 따라 미디어 재생 등의 action 을 수행해야 한다. **[3.2.5] G-Plugin 서비스 Callback 등록 참고**

3.3.5.1.1. onCommand - dialog_response

onCommand Callback 의 actionType 이 dialog_response 인 경우, 대화서버 응답을 파싱하여 서비스 앱의 flow 에 맞도록 처리해야 한다. 기가지니 포탈의 Dialog Kit 으로 서비스 시나리오를 구성하면 대화서버의 응답을 custom 할 수 있다.

예제)

```
"actionType": "dialog_response",  
"dialogResponse": JSON, //JSON포맷의 대화서버 응답 결과로 3rd party에서 파싱하여 처리
```

① dialogResponse 의 JSON 포맷

key	Description
Intent	DialogKit에서 정의한 Intent
applInfo	DialogKit에서 정의한 NE정보

3.3.5.1.2. onCommand - media_stream

onCommand Callback 의 actionType 이 media_stream 인 경우, Callback 의 mediastream 을 재생해야 한다. 특히 TTS 재생이 요청될 때 사용된다. mediastream 정보는 String 형태로 전달되며, 이를 다음과 같이 byte array 로 변환하여 사용할 수 있다.

- byte[] decodedString = Base64.decode(mediastream, Base64.DEFAULT);

예제)

```
"actionType": "media_stream",  
"mediastream": AUDIO_STREAM_STRING, //음성 ByteArray (바로 재생 또는 audio file 생성 가능)
```

3.3.5.1.3. onCommand - media_stream_meta

media_stream Callback 이 전달될때, 해당 stream 의 meta 정보는 media_stream_meta Callback 으로 연속적으로 전달된다. 이는 TTS 재생 메시지를 화면에 표시하거나, 응답의 감성 정보를 표현할 때 사용할 수 있는 정보이다.

예제)

```
"actionType":"media_stream_meta",
"commandType":"Req_PLMD",
"payload":
{
  "cmdOpt":{
    "actOnOther":"pause",
    "channel":"0",
    "errorCode":0,
    "metaInfo":
    {
      "actFeel":"Neutral",
      "media_playtime":0,
      "media_size":0,
      "mesg":"안녕하세요?",
      "sentencePattern":""
    },
    "playNotiTime":0,
    "playTime":0
  }
}
```

- cmdOpt: 명령어 옵션으로 다음의 옵션 전달
 - channel: 출력할 채널 번호
 - actOnOther: 다른 출력 미디어에 대한 action(다른 출력 미디어에 대한 액션 표 참조)
 - metaInfo: media 메타 정보가 설정됨(metaInfo: TTS 재생하는 Text 정보 / actFeel: 응답의 감성 정보)

① 다른 출력 미디어에 대한 액션(actOnOther)

Action	Description
mute	현재 다른 미디어가 재생 중이면 mute 시키고, 미디어 출력 완료 후에도 mute 상태 유지
muteR	현재 다른 미디어가 재생 중이면 mute 시킴, 미디어 출력 완료 후 unmute 시킴
pause	현재 다른 미디어가 재생 중이면 pause 시키고, 미디어 출력 완료 후에도 pause상태 유지
pauseR	현재 다른 미디어가 재생 중이면 pause 시킴, 미디어 출력 완료 후 다른 미디어 resume 시킴
stop	현재 다른 미디어가 재생 중이면 stop 시킴, 다른 미디어 완전 재생 종료
volDown	현재 다른 미디어가 재생 중이면 volume-down 시킴, 미디어 출력 완료 후에도 volume-down 상태 유지
volDownR	현재 다른 미디어가 재생 중이면 volume-Down 시킴, 미디어 출력 완료 후 기존 볼륨 복구

3.3.5.1.4. onCommand - media_url

onCommand Callback 의 actionType 이 media_url 인 경우, 전달받은 url 을 통해 미디어 재생해야 한다. 미디어 재생에 대한 옵션과 metadata 는 onCommand Callback 의 payload(JSON 포맷)로 아래와 같이 전달된다.

예제)

```

"actionType": "media_url",
"payload":
{
  "cmdOpt": {
    "channel": 1,
    "actOnOther": "pause",
    "url": "http://dev-apis.ginie.co.kr/api/v1/tracks/87101805",
    "playNotiTime": 60000,
    "metaInfo": { media_type: 'music', title: '좋은날', artist: '아이유', ..... },
  }
}

```

```

    }
}

```

- cmdOpt: 명령어 옵션으로 다음의 옵션 전달
 - channel: 출력할 채널 번호
 - actOnOther: 다른 출력 미디어에 대한 action(다른 출력 미디어에 대한 액션 표 참조)
 - url: 출력 URL (channel 이 0 이면 null)
 - playNotiTime: 특정 시간 플레이 되었는지 noti해주는 시간(지니뮤직은 60 초)
 - metaInfo: media 메타 정보가 설정됨(미디어 메타 정보 표 참조)

① 다른 출력 미디어에 대한 액션(actOnOther)

Action	Description
mute	현재 다른 미디어가 재생 중이면 mute 시키고, 미디어 출력 완료 후에도 mute 상태 유지
muteR	현재 다른 미디어가 재생 중이면 mute 시킴, 미디어 출력 완료 후 unmute 시킴
pause	현재 다른 미디어가 재생 중이면 pause 시키고, 미디어 출력 완료 후에도 pause상태 유지
pauseR	현재 다른 미디어가 재생 중이면 pause 시킴, 미디어 출력 완료 후 다른 미디어 resume 시킴
stop	현재 다른 미디어가 재생 중이면 stop 시킴, 다른 미디어 완전 재생 종료
volDown	현재 다른 미디어가 재생 중이면 volume-down 시킴, 미디어 출력 완료 후에도 volume-down 상태 유지
volDownR	현재 다른 미디어가 재생 중이면 volume-Down 시킴, 미디어 출력 완료 후 기존 볼륨 복구

② 미디어 메타 정보(metaInfo)

Key		Value		
media_type	mand	music	radio	podcast
title	mand	노래 제목	라디오 채널명	팟캐스트 제목

description	opt	설명		
artist	opt	가수명	방송사	제작자
imageurl	opt	앨범 이미지	채널 이미지	팟캐스트 이미지
media_name	opt	노래제목		에피소드명
media_playtime	opt	재생 시간		
media_size	opt	Bytes		
category	opt	카테고리(가요, 팝송)		
album_name	opt	앨범 명		

3.3.5.1.5. onCommand - webview_url

onCommand Callback 의 actionType 이 webview_url 인 경우, 전달받은 url 을 웹뷰로 실행해야 한다. (예, 지니뮤직 로그인 요청이 콜백으로 온 경우)

예제)

```
"actionType": "webview_url",
"oauth_url": "https://dev-
auth.genie.co.kr/oauth/authorize?response_type=code&client_id=YjEyYWM0NDetYzBkNC00MjkzL
TkyODQt&redirect_uri=https%3A%2F%2Fagent.gigagenie.ai%3A10089%2Foauthresult%2Fgeniem
usic%2F07d87279-0cf8-5ce5-b04f-76941a6fc4f6&state=DIA0K3vIBG9"
```

3.3.5.1.6. onCommand - start_voice

onCommand Callback 의 actionType 이 start_voice 인 경우, voiceCommand API 를 호출하여 음성인식을 진행해야 한다. (예, '일정 등록' 등 연속 발화가 필요한 시나리오의 경우)

예제)

```
"actionType": "start_voice",
"payload":
{
  "cmdOpt":{
    .....
  }
}
```

3.3.5.2. 미디어 재생 상태 업데이트

3.3.5.2.1. setMediaStatus()

3rd Party Application 이 미디어를 제어하는 경우(재생, 일시정지, 중단, 완료 등) G-Plugin 으로 현재 미디어 재생 상태를 전달해야 한다. 상태 정보는 필수로 전달해야 하며, 해당 정보를 GiGA Genie AI 서버에서 있어야 상태에 따른 적절한 command 를 판단 가능하다.

Column Name	Type	설명
channel	int	출력하는 채널 정보
status	String	started(시작됨), paused(중지됨), resumed(재시작됨), complete(플레이 완료), stopped(외부 요인에 의한 종료), noti(playNotiTime 에 의한 noti)
playTime	int	플레이 시간(ms)

3.3.5.2.2. setMediaStatus 사용 예시

```
mGplugin.setMediaStatus(0, "paused", 1200);
```

3.4. G-Plugin 단위서비스 API

3.4.1. TTS API

3.4.1.1. TTS Stream 요청

3.4.1.1.1. getTTS(String msg)

3rd Party Application 이 TTS 재생을 위한 Audio Stream 을 요청한다. 리턴되는 stream 정보는 wav 포맷으로 wav header 44bytes 를 포함한다. 3rd Party Application 은 해당 값으로 wav 파일을 생성하여 재생하거나, 44bytes 이후 정보만 재생하도록 구성하면 TTS stream 이 재생된다.

Parameter 정의

Column Name	Type	설명
msg	String	TTS 재생할 메시지(WAV 포맷)

3.4.1.1.2. getTTS 호출 예시

```
byte[] audioContent = mGplugin.getTTS("테스트 중입니다");
byte[] slice = Arrays.copyOfRange(audioContent, 44, audioContent.length);
...
mAudioTrack.play();
mAudioTrack.write(slice, 0, slice.length); //44bytes 이후 정보만 재생
```

4. 개발 참고 사항 및 이슈 사항

4.1. 예외처리(TBD)

4.2. FAQ

4.2.1. "CFK_FAILED_CONN" 에러코드 발생하는 경우

음성인식 서버는 TB(TestBed)서버와 상용서버로 구분되어 있다. TB 음성인식 서버에 연동하는 경우, 단말이 WiFi 나 타사 통신망(SKT, LGU+)에 연결되어 있으면 "CFK_FAILED_CONN" 에러코드가 발생한다 (첨부 1 [음성인식 에러 코드] 참고).

TB 음성인식 서버 연동은 KT 통신망에서만 가능하며, 상용 음성인식 서버는 통신사나 WiFi 연결 상태와 관련 없이 정상 동작한다.

[첨부 1] G-Plugin EventCode

Event Code	설명	전달 데이터
"VOICE_STARTED"	음성인식 시작	event.putString("eventCode", " VOICE_STARTED");
"VOICE_STOPPED"	음성인식 종료	event.putString("eventCode", " VOICE_STOPPED");
"SERVER_ERROR"	서버 오류	event.putString("eventCode", " SERVER_ERROR"); event.putString("data", errorCause); - errorCause : 서버 오류 코드(첨부 2)
"KWS_STARTED"	KWS 시작	event.putString("eventCode", " KWS_STARTED");
"KWS_DETECTED"	KWS 감지	event.putString("eventCode", " KWS_DETECTED");
"KWS_ERROR"	KWS 오류	event.putString("eventCode", " KWS_ERROR");

[첨부 2] G-Plugin ErrorCause

Error Cause	설명
900	음성인식 서버 접속 실패
901	음성인식 시작 후 5 초 이내에 실제 음성 감지가 발생하지 않는 경우 (발화를 하지 않는 경우)
902	음성인식 시작 후 10 초 이내에 음성 끝 감지가 발생하지 않는 경우 (발화를 10 초 이상 계속 하는 경우)
910	TTS 서버 접속 실패
911	TTS 변환 요청 실패
912	TTS 요청 텍스트의 포맷이 잘못되었거나 빈 스트링인 경우
920	대화 서버 접속 실패
921	클라이언트 타입에 상응하는 대화 서버 프로파일 없음
930	Filter 서버 접속 실패
601	음성인식 시작 요청이 오지 않는 경우
602	음성인식 중단 요청이 오지 않는 경우

[첨부 3] KWS ErrorCause

Error Cause	설명	Error 처리 방식
E_INIT	KWS 라이브러리 초기화 실패	호출어 정지(stopKws)
E_MIC	KWS 마이크 연동 실패	호출어 정지(stopKws)
E_NETWORK	KWS 네트워크 연동 실패	호출어 정지(stopKws) 및 시작(startKws)
E_AUDIO	KWS 오디오 오류	호출어 정지(stopKws)
E_RECOGN_TIMEOUT	KWS 인식 결과 타임아웃	호출어 정지(stopKws) 및 시작(startKws)
E_AUDIO_TIMEOUT	KWS 오디오 데이터 입력 대기 초과	호출어 정지(stopKws) 및 시작(startKws)
E_INTERNAL	KWS 내부 오류	호출어 정지(stopKws)

E_SPEECH_TIMEOUT	KWS 음성입력 신호 타임아웃	호출어 정지(stopKws) 및 시작(startKws)
E_UNKNOWN	정의되지 않는 오류	호출어 정지(stopKws)

[첨부 4] 음성 및 대화 Life-cycle

1. startKws 로 호출어 인식 대기
2. 호출어 발화
3. 음성인식 시작
4. 사용자 발화
5. 음성인식 종료
6. 서버에서 response 수신
7. App 에서 action 수행
8. startKws 로 호출어 인식 대기