

GiGA Genie Service SDK

이용 가이드 v1.4

GiGA Genie

2017.11

KT Service 연구소

목차

1. 개요	3
2. 일반 사항.....	7
3. Namespace 별 서비스 제공 목록	18
4. API 설명	22
4.1 API 초기화 (gigagenie.init).....	22
4.2 Voice API (gigagenie.voice).....	25
4.3 통화 API (gigagenie.call)	37
4.4 데이터관리 API (gigagenie.appdata).....	38
4.5 G-Box 정보 관리 API (gigagenie.appinfo).....	44
4.6 Media API (gigagenie.media)	55
4.7 Payment API (gigagenie.payment).....	65
4.8 영상인식 API (gigagenie.vision)	67
4.9 Admin API (gigagenie.admin).....	69
5. 참고 사항.....	73
5.1 3rd Party 서비스 사용자 인증 방법	73
5.2 리모컨 Key Code 가이드	74
5.3 gigagenie.test.setAudioRecog 이용 플로우	75
5.4 WebRTC 이용 플로우 (AudioStream).....	76
6. 별첨	77

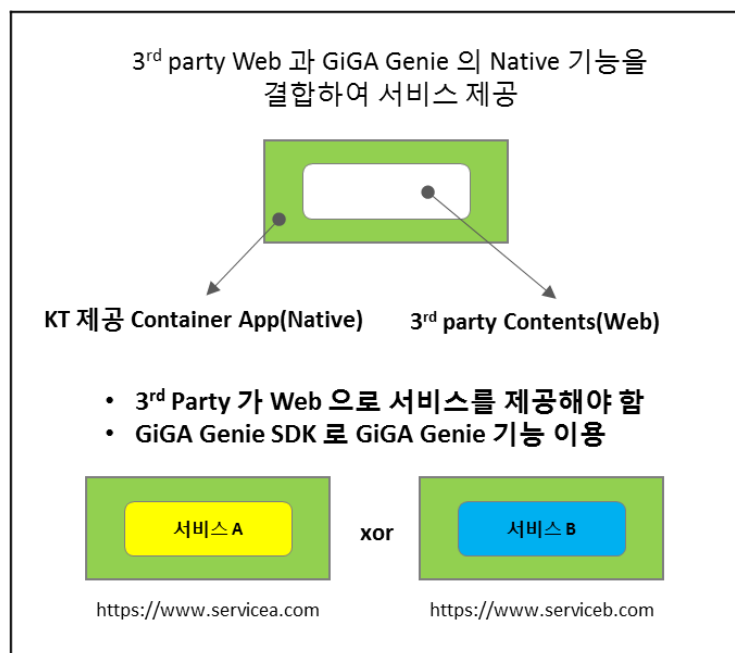
1. 개요

1.1 목적

본 문서는 3rd Party 사업자가 GiGA Genie 에서 서비스를 제공하기 위해 사용해야 하는 Service SDK 의 이용 가이드이다. Service SDK 를 이용하면, 3rd Party 사업자는 GiGA Genie 에 탑재된 음성, 통화, 데이터 관리 기능을 활용하여 GiGA Genie 에 제공될 서비스를 개발할 수 있다. 본 문서에서는 Service SDK 를 이용하기 위한 방법과 Service SDK 가 제공하는 API(Application Programming Interface)에 대한 설명, 그리고 이를 활용한 샘플 코드의 예를 제시한다.

1.2 3rd Party App 수용 구조

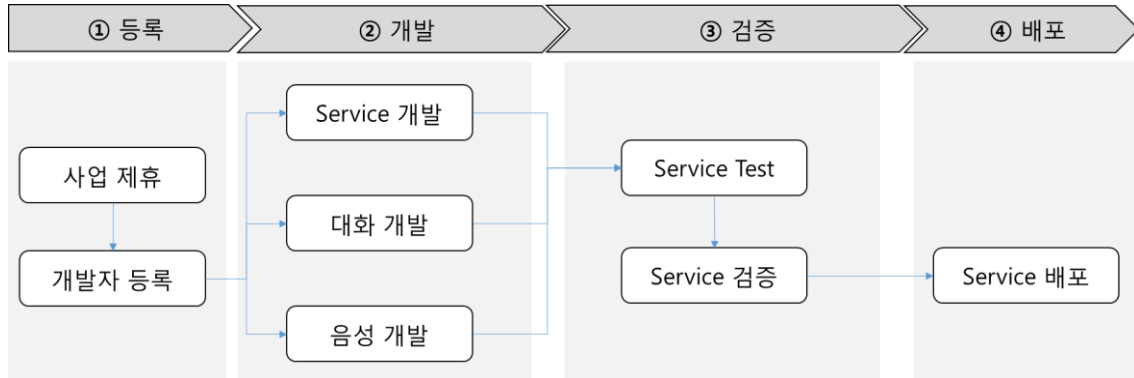
GiGA Genie 에서 3rd Party 서비스는 GiGA Genie Service SDK 를 이용해서 구현한 3rd Party 의 Web-App 을 GiGA Genie 의 Container App 에서 실행시켜 주는 형태로 제공된다.



3rd Party 는 Service SDK 가 제공하는 JavaScript API 를 이용해서 웹 어플리케이션으로 구현해야 하며, 자체 웹 서버를 통해서(또는 KT 와 제휴하여 GiGA Genie Service SDK 에서 제공하는 Hosting 웹서버를 이용하여) 서비스를 제공하여야 한다.

1.3 3rd Party 서비스 제공 절차

3rd Party 가 GiGA Genie 에서 서비스를 제공하기 위해서는 다음의 단계를 거친다.



1. 등록

○ 사업 제휴

- 서비스 제공, 홍보, 결제/과금 등의 사업적 측면의 절차이다. (향후 관련 절차 제공 예정)

○ 개발자 등록

- GiGA Genie 에서 서비스를 개발하고 제공하고자 하는 사업자는 개발자 등록을 하여 developer id (개발자 ID)를 발급 받아야 한다.
- 개발자 등록시, 제공하고자 하는 서비스(이하 App)정보를 등록하여 app id 를 발급 받아야 한다.
- App 정보 등록과 함께 App 의 인증 키 값인 AppKey 를 발급 받아야 한다. (개발자 센터는 2017 년 9 월에 웹으로 제공될 예정이며, 그 전에는 담당자에게 관련 ID/Key 를 발급 받는다.)

2. 개발

○ Service 개발

- Service 개발은 GiGA Genie Service SDK 를 이용해서 제공할 App 을 개발하는 과정으로 본 문서를 참조하여 개발해야 한다.

○ 대화 개발

- 대화 개발은 상황 인지를 위한 개발로, 3rd Party App 의 실행 조건 또는 상황을 판단하도록 한다. 관련 내용은 본 문서에서 다루지 않는다.

○ 음성 개발

- 3rd Party 서비스에서 필요한 음성 서비스를 개발하는 단계로, 관련 내용은 본 문서에서 다루지 않는다.

3. 검증

○ Service Test

- 3rd Party 가 개발한 서비스를 테스트 하는 과정이다. 테스트는 다음의 두가지 방법으로 진행한다.
 - Emulator 테스트 : Emulator 에서 테스트 하는 과정으로, 개발 즉시 확인이 가능하다.
 - G-BoX 개발자 모드 테스트 : G-Box 의 Container App 을 개발자 모드로 설정-테스트 한다. G-Box 의 개발자 모드는 특정 Device 의 특정 AppID 에만 동작한다.

○ 서비스 검증

- 서비스 검증은 해당 서비스를 배포하기 전에 동작을 확인하는 단계이다.
(향후 관련 절차 제공 예정)

4. 배포

○ Service 배포

- 서비스 검증은 해당 서비스를 배포하기 전에 동작을 확인하는 단계이다.
(향후 관련 절차 제공 예정)

1.4 용어 정리

용어	설명
GiGA Genie 단말	셋탑박스 형태로 제공되는 GiGA Genie 서비스 하드웨어를 의미함
G-Box	GiGA Genie 단말과 같은 의미
Service SDK	특정 서비스를 개발하기 위한 환경과 관련 API, 개발 툴킷을 의미
Hybrid App	HTML5, CSS, Javascript 등 웹의 표준 기술을 이용, HTML 에 단말의 네이티브 호출을 적용하여 구현한 앱
MC	Main Controller 의 준말로, GiGA Genie 의 서비스 제어 프로그램을 의미
Proxy	클라이언트가 간접적으로 다른 네트워크 서비스에 접속하도록 해주는 시스템
Container App	다른 App 을 실행해주는 환경을 Container 라 지칭하며, Container 를 실행해주는 앱을 의미함
JS	JavaScript 의 약자
Emulator	Emulator 는 특정 시스템의 동작을 흉내내는 시스템
UUID	Universally Unique Identifier 의 약자로, 소프트웨어 구축에 쓰이는 식별자 표준임
Sandbox	외부로부터 들어온 프로그램이 보호된 영역에서 동작하도록 하는 것
GATE	GiGA Genie App Transmission Environment 의 약자로 Service SDK 동작을 지원하기 위한 서버 시스템을 의미한다.

2. 일반 사항

2.1 공통 사항

모든 API 는 Callback 방식으로 결과를 전달한다. 다음과 같이 호출한다.

- gigagenie.[영역].[함수](함수파라미터, callback)으로 호출한다.
- [영역] 은 API 를 Grouping 하는 단위로 이후 namespace 로 설명한다.
- callback 함수는 다음과 같이 정의 된다.
 - function callback(result_cd,result_msg,extra)
 - ◆ result_cd: 결과 코드, 숫자
 - ◆ result_msg: 결과 메시지, 문자
 - ◆ extra : 호출 결과로 전달되는 데이터들(Javascript Object)
 - 각각의 API 에 따라서 전달되는 데이터가 다르다.
 - 예: getSelectedIndex 의 경우 extra 에 selected 라는 숫자 값이 설정되어 전달된다.

2.2 NameSpace

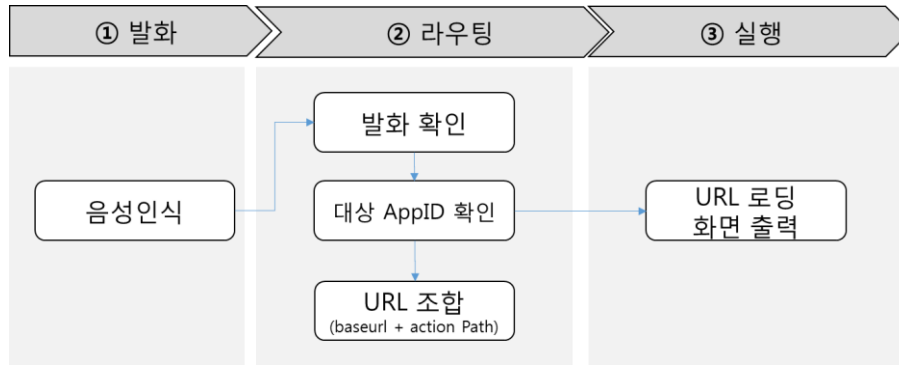
API 는 Javascript Object 로 제공된다. Javascript Object 를 가져오기 위해서는 HTML 에 다음을 Embed 해야 한다.

```
<script type="text/javascript" src="https://svcapl.gigagenie.ai/sdk/v1.0/js/gigagenie.js">
```

- HTML 에 embed 하면 gigagenie 객체를 이용 가능하다.
 - 함수 호출은 gigagenie.[영역].[함수명](파라미터) 로 실행한다.
 - 호출 예: gigagenie.voice.getVoiceText("안녕 기가지니",null,callback)

2.3 Container App 에서 3rd Party 앱 실행

GiGA Genie 에서 사용자가 3rd Party 앱을 이용하고자 할 경우 다음의 절차로 3rd Party 앱을 실행(3rd Party URL 로 Container App 에서 브라우징)한다.



사용자는 “기가지니 [발화]” 를 이용해서 3rd Party App 을 실행한다. 음성인식 이후 발화는 3rd Party 앱 실행을 위한 발화인지 확인(해석)되며 실행 대상 AppID 를 확인한다. 유효한 AppID 일 경우 해당 AppID 에 등록된 3rd Party 웹의 baseUrl + ActionPath 를 Container App 에서 로딩, 화면에 출력하는 형태로 실행 된다. 이때, 발화에서 생성한 파라미터를 GET Parameter 방식으로 전달한다.

2.4 웹 URL 구성 방법

2.4 에서 설명한 것처럼 3rd Party 웹 로딩은 baseUrl + ActionPath 를 조합한 URL 을 로딩한다. 그리고 HTML 태그 상에서 링크(href), 소스(src)는 baseUrl 기준으로 해석된다. 다음은 3rd Party 가 등록한 appid 의 예이다.

Appid	대상서비스	baseUrl	Intent(actioncode)	ActionPath
TEST0001	테스트앱	https://test.com	QueryTest1	/app/test1.jsp
			QueryTest2	/app/test2.jsp
			QueryTest3	/app/test3.jsp

의 발화 확인 시 대상 appid 가 TEST0001 이고 Action 이 QueryTest1 이면 ContainerApp 은 baseUrl+ActionPath 조합인 <https://test.com/app/test1.jsp> 를 로드하며 GET Parameter 로 발화 확인된 파라미터를 전달한다. 따라서 발화 개발시에 파라미터를 받도록 하였으면 <https://test.com/app/test1.jsp?key=value&key=value> 형태로 전달 받도록 URL 을 구성해야 한다.

2.5 링크, 소스 태그의 경로 설정 방법

Container App 에서 3rd Party 웹서버로 접근할 때, 링크(href), 소스(src) 태그의 경우 절대 경로를 사용해야 한다. Container App 에서는 3rd Party 웹서버에 접근 할 때,

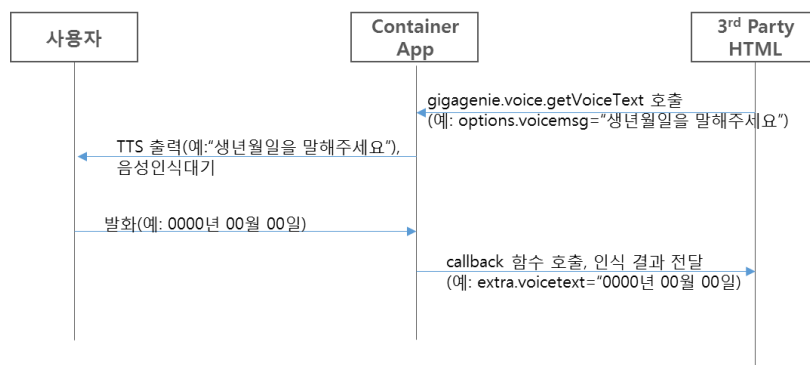
baseUrl 을 기준으로 링크, 소스를 Access 하기 때문에, 상대 경로를 사용하면 해당 리소스가 접근이 안될 수 있다.

- 잘못된 설정 예: baseUrl 이 <http://example.com> 이고 해당 action URL 이 /gigagenie/index.html 일 경우 index.html 에 다음의 코드가 들어갔을 경우 해당 URL 을 참조할 수 없게 된다.
 - < link href="../../css/style.css" rel="stylesheet">
 - <script src="../../js/jquery-min.js"> </script>
- 바람직한 설정 예 : 상기 링크, 소스 태그를 baseUrl 로 부터의 절대 경로로 지정해야 한다.
 - <link href="/css/style.css" rel="stylesheet">
 - <script src="/js/jquery-min.js"> </script>

2.6 음성 이벤트의 동작 방식

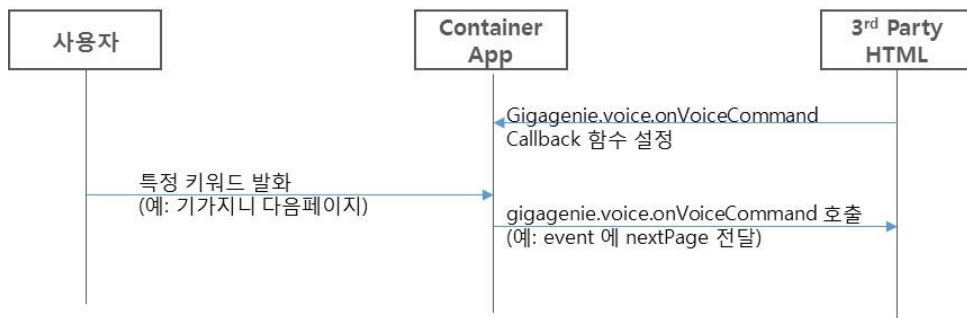
음성 이벤트 수신은 다음의 Voice API 의 Callback 으로 전달하며, API 에 따라 동작 방식과 전달 데이터가 다르다.

- 음성인식 API (4.2.1 gigagenie.voice.getVoiceText)
 - 음성 인식 API 는 gigagenie.voice.getVoiceText 호출 이후 해당 함수에 등록된 callback 에 음성인식 결과가 전달된다.
 - JavaScript API 로 명시적으로 음성 인식을 개시한다.
 - 동작 Flow :



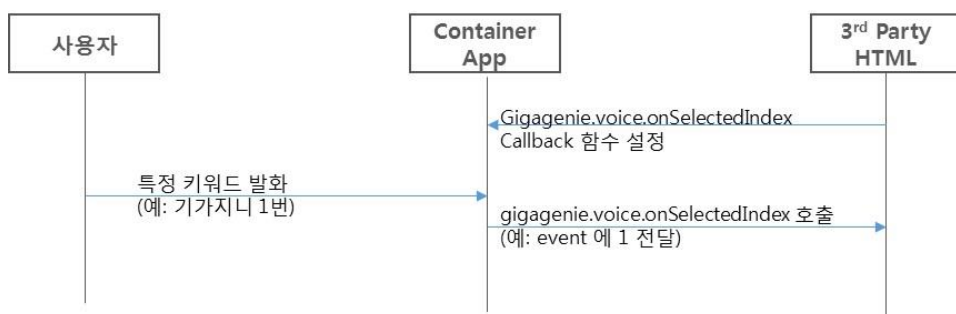
- 음성명령수신 API (4.2.3 gigagenie.voice.onVoiceComand)

- 음성 명령 수신 API 는 `gigagenie.voice.onVoiceCommand` 에 callback 함수를 등록하면 callback 에 명령 수신 결과를 전달한다.
- 사용자의 인식 특정 발화(기가지니, 지니야, 친구야, 자기야) 이후 특정 명령에 대한 발화(다음페이지, 이전페이지) 발생시 callback 에 관련 결과를 전달한다.
- 사용자 발화로 음성 인식이 개시한다.
- 동작 Flow



○ 음성선택번호수신 API (4.2.4 `gigagenie.voice.onSelectedIndex`)

- 음성선택번호 수신 API 는 `gigagenie.voice.onSelectedIndex` 에 callback 함수를 등록하면 callback 에 명령 수신 결과를 전달한다.
- 사용자의 인식 특정 발화(기가지니, 지니야, 친구야, 자기야) 이후 번호를 발화하면 callback 에 발화한 번호를 전달한다.
- 사용자 발화로 음성 인식이 개시한다.
- 동작 Flow

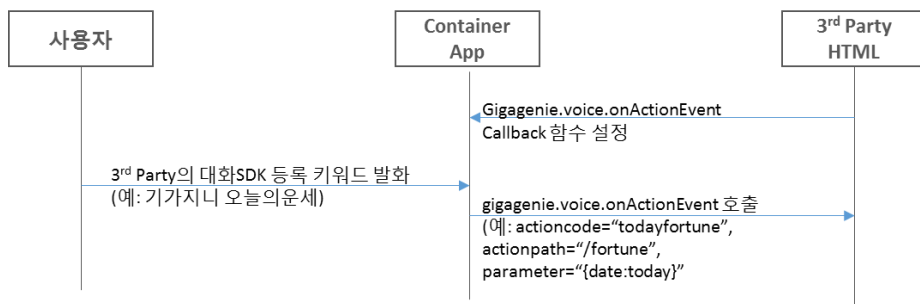


○ Action 수신 API (4.2.7 `gigagenie.voice.onActionEvent`)

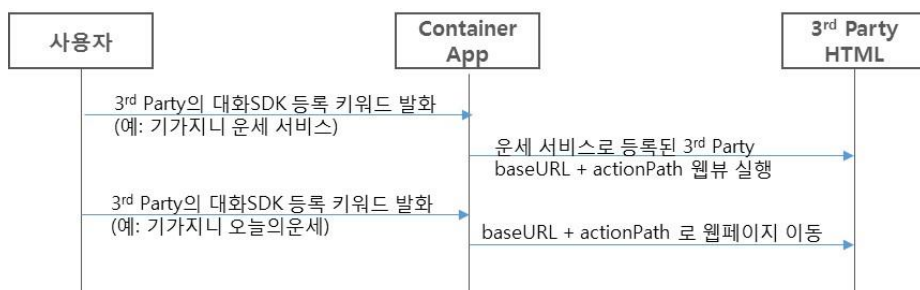
- Action 수신 API 는 `gigagenie.voice.onActionEvent` 에 callback 함수를 등록하면 callback 에 명령 수신 결과를 전달한다.
- 3rd Party 가 대화 SDK 에서 생성한 대화 모델에 포함되는 발화에 대해, 현재 해당 3rd Party App 을 실행 중 일 때 JavaScript Callback 함수를 호출한다.

- ◆ 3rd Party App 이 현재 실행 중이지 않을 경우 등록된 baseURL + Path 로 Container App 을 실행한다.
- ◆ 현재 3rd Party App 이 실행 중이고, 3rd Party App 에서 onActionEvent Callback 을 구현하지 않을 경우 해당 발화가 발생하면 등록된 baseURL + Path 로 이동한다.
- ◆ 현재 실행중인 3rd Party App 과 발화로 실행되어야 하는 3rd Party App 이 다를 경우 현재 실행중인 3rd Party App 을 중지하고 발화로 실행되어야 할 3rd Party App 을 실행한다.
- 사용자 발화로 음성 인식이 개시된다.
- 동작 Flow

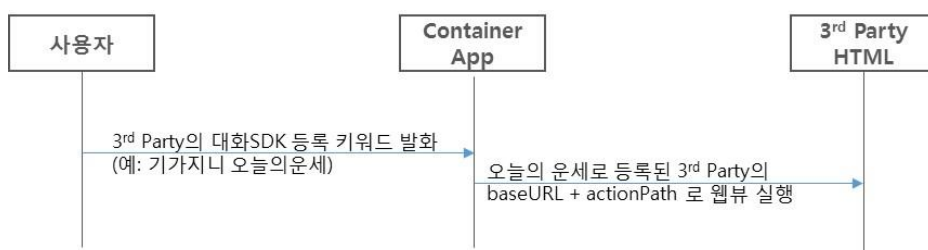
- ◆ 3rd Party App 이 현재 실행 중이고 Callback 함수를 설정한 경우



- ◆ 3rd Party App 이 현재 실행 중이고 Callback 함수를 설정하지 않은 경우



- ◆ 3rd Party App 이 실행중이지 않은 경우



2.7 화면 해상도 설정

3rd Party 화면은 1920 x 1080 으로 표시된다. HTML 에 아래 Meta Tag 를 등록해야 정상적으로 보여진다.

```
<meta name="viewport" content="height=1080, width=1920, user-scalable=no" />
```

2.8 URL 호출시 Parameter 로 전달되는 값

- 화자식별 발화 정보
 - 음성명령 호출시, 화자 인식된 정보는 인식된 사용자의 ContainerID 가 GET Parameter 로 전달된다. GET Parameter 는 voiceid 이다. 화자 식별이 등록되어 있지 않으면 "NONE"으로 화자 식별이 인식이 안된 경우 "UNKNOWN"이 전달된다.
- 사용자 발화 문구
 - 음성명령 호출 시 사용자가 발화한 문구가 GET Parameter 로 전달된다. Get Parameter 는 uword 이다. 발화로 진입한 경우가 아니면 "NONE"이 전달된다.

2.9 API 권한 확인

결제 API 등 일부 API 는 KT 와 관련 제휴(CP 등록 등)를 맺은 AppId 만 호출 가능하다. 권한이 없는 AppId 에서 해당 API 를 호출하면 result_cd 는 401, result_msg 는 "Unauthorized" 가 발생한다.

다음은 권한이 필요한 API 와 권한 Group 이다.

○ 결제 권한 그룹

권한 Group	대상 Namespace
payment	payment.*
userinfo	appinfo.getAddressDetail
	appinfo.getUserSetInfo
ktadmin	admin.*

2.10 TTS 시작-종료 관련

TTS 는 sendTTS(4.2.2) 또는 getVoiceText(4.2.1)에 의해서 시작되고 stopTTS(4.2.9) 에 의해서 종료될 수 있다. 또는 3rd Party App 과 관계 없이 사용자가 "기가지니"와

같은 KWS(Key Word Spotting)의 질의 응답, 또는 전화가 왔을 경우 등에 따라 외부 요인에 따라 종료될 수 있다.

○ TTS 시작-중지-재시작 관련

sendTTS 로 TTS 재생시 sendTTS 를 호출하게 되면 기존 sendTTS 의 callback 에는 result_cd 에 오류코드가 전달되고 기존 TTS 는 정지한다. 그리고 새로 호출한 sendTTS 의 내용에 의해 TTS 가 재생된다. 이는 getVoiceText 에서 TTS 를 재생할 때도 동일하게 적용된다. 즉 getVoiceText 의 TTS 재생시 getVoiceText 를 호출하면 기존 TTS 는 정지되고 처음 호출한 getVoiceText 의 콜백에는 result_cd 에 오류코드가 전달된다. getVoiceText 호출 이후 sendTTS 호출, 또는 sendTTS 호출 이후 getVoiceText 를 호출 하는 경우도 동일하다.

getVoiceText 시 TTS 가 재생되고 음성인식 대기 상태로 있는 경우 getVoiceText 또는 sendTTS 를 다시 호출하면 음성인식이 중지되고 getVoiceText 또는 sendTTS 가 시작된다.

○ 외부에 의한 TTS 중지

sendTTS 와 getVoiceText 로 TTS 재생시 KWS 가 발생하거나 전화가 오는 등, 외부에서 오디오를 사용해야 하는 경우 TTS 는 자동 중지된다. KWS 의 경우 sendTTS, getVoiceText 로 501 오류가 전달된다. 리모콘의 마이크 버튼을 누를 경우 또는 전화 오는 경우 sendTTS, getVoiceText 로 503 이 전달된다. 이 경우에 대한 복구는 onResume Event 콜백을 받아서 복구해야 한다.

○ 외부에 의한 TTS 중지

2.11 Mute/Unmute, App Pause/Resume Event 관련

기가지니 단말의 상황에 따라 Mute/Unmute, Pause/Resume Event 가 발생한다.

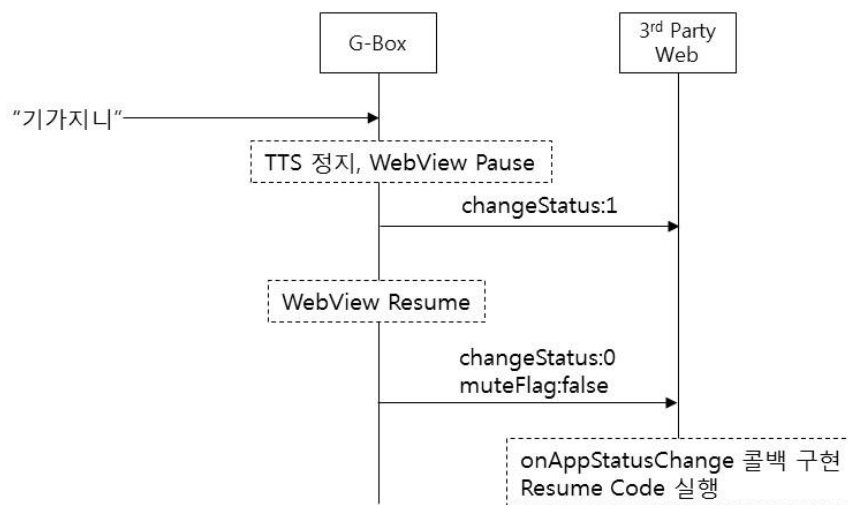
○ Pause/Resume Event

Pause/Resume Event 는 3rd Party 서비스 제공 중에 3rd Party 서비스가 종료되지 않은 상황에서 다른 앱의 화면이 앞에 표시될 때 발생한다. Pause Event 발생시 TTS 는 중지되며 3rd Party Web 은 중지 상태가 된다. (동영상 등 웹뷰 타이머 정지)
Resume Event 발생시 3rd Party Web 은 활성화 되지만, 동영상 플레이 등을 복구해 주지 않는다. 3rd Party 는 Resume Event 발생시 4.1.3 의 onAppStatusChange 의

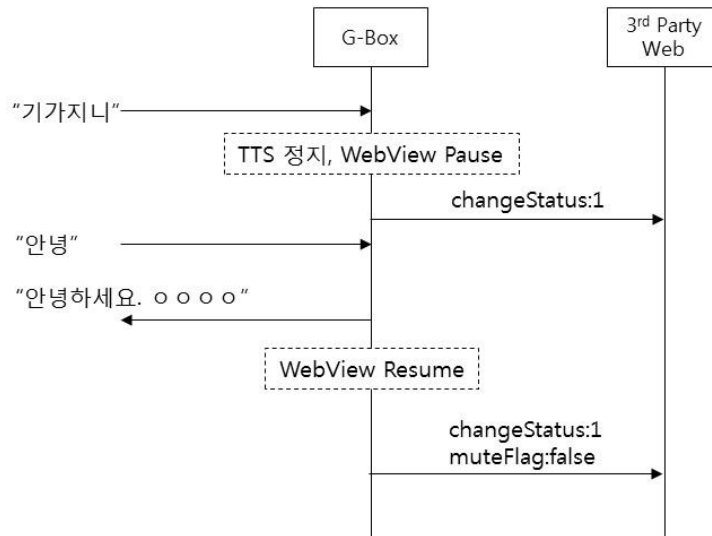
콜백을 구현하여 복구를 시도해야 한다. 또한 Resume Event 는 onAppStatusChange 콜백에 muteFlag 를 전달한다. muteFlag 가 true 인 경우 3rd Party Web 이 활성화 되었지만 라디오 재생, 음악 재생하고 있는 상태이다. muteFlag 가 false 의 경우 3rd Party Web 이 활성화 되었으며 Background 에 음성을 사용하고 있지 않은 상태이다. 3rd Party Web 은 Resume 에서 muteFlag 가 true 일 경우에는 오디오를 재생하면 안된다. Resume 시 또는 Resume 이후에 mute 상태 해제가 되면 unmute event 가 발생한다.

1) Keyword Spotting 발화(기가지니, 지니야 등), 2) 발화로 라디오 실행(기가지니, 라디오 커), 3) 발화로 음악 재생, 4) 통화 발생 시 Pause Event 가 발생되며 4.1.3 의 onAppStatusChange 로 changeStatus=1 으로 리턴한다. 그리고 각각의 상황 종료시 Resume Event 가 발생되며 4.1.3 의 onAppStatusChange 로 changeStatus=0 로 리턴한다. Resume Event 는 현재 상황에 따라 muteFlag 가 설정되어 전달된다.

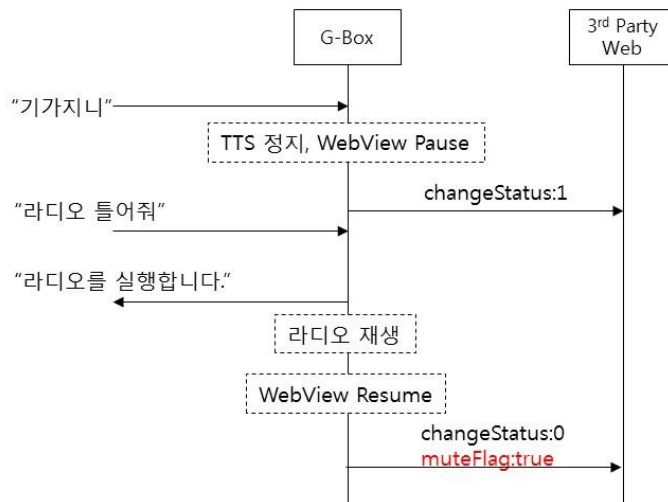
- Case1 : "기가지니" 으로 발화 할 때



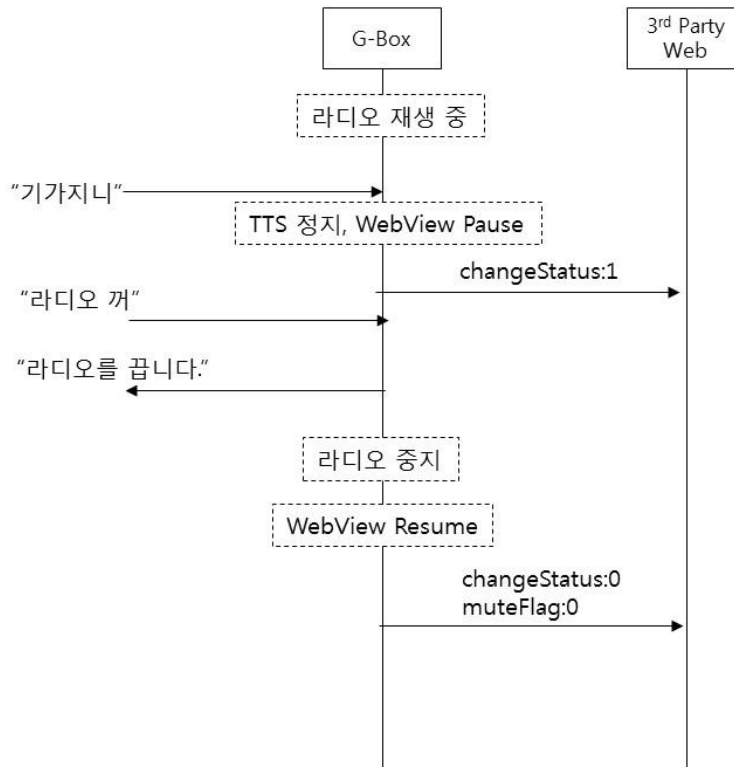
- Case2: "기가지니, 안녕"으로 발화할 때



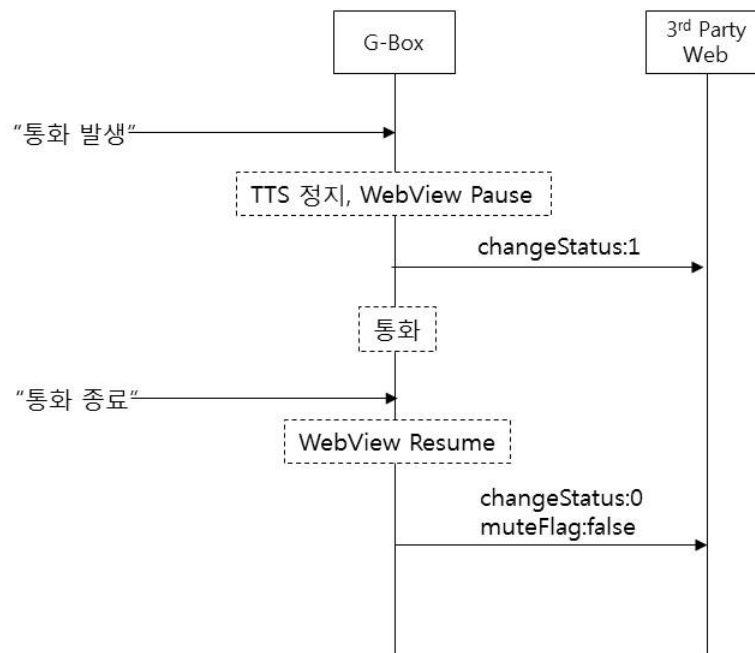
- Case3: "기가지니, 라디오 켜"로 발화할 때 (음악 켜도 동일)



- Case4: Case3 상황에서 "기가지니, 라디오 꺼"로 발화할 때(음악 꺼도 동일)



- Case5: 통화 발생 시



○ Mute/Unmute Event

Mute/Unmute 는 화면에 보여지는 안내 없이, Background 에서 서비스가 실행되어야 하는 경우(USB 뮤직 실행시) 발생된다. 이 경우 3rd Party Web 이 정지되지 않으며,

Mute 에는 Audio 를 중지시키는 코드가, unmute 는 이를 복구하는 코드가 들어가야 한다. (4.6.1 Mute 이벤트 API 참고)

하위 호환성을 위해서 pause event 발생시 mute event 를 보내고, resume 시에 muteflag 가 true 일 경우 unmute event 를 보낸다.

3. Namespace 별 서비스 제공 목록

3.1 API 초기화 (Namespace: gigagenie)

순번	서비스명	설명	비고
1	API 초기화	GiGA Genie Service API 를 초기화 한다.	Web → Native
2	ContainerId 변경	화자인식된 ContainerID 로 ContainerID 를 변경한다.	Web → Native
3	Pause/Resume	Application 의 Pause/Resume 이벤트를 전달한다.	Native → Web

3.2 Voice API 리스트 정보 (Namespace: gigagenie.voice)

순번	서비스명	설명	비고
1	음성 인식 요청	사용자에게 안내를 하고 음성 인식 대기한다.	Web → Native
2	TTS	입력 Text 를 사용자에게 TTS 로 전달한다.	Web → Native
3	음성 명령 수신	이전페이지, 다음페이지 음성 명령 코드를 수신한다.	Native → Web
4	음성 명령 수신 설정	KWS 이후 번호 선택과 확인/취소 등을 웹에 전달하도록 설정한다.	Web → Native
5	음성 선택 번호 수신	사용자가 특정 번호를 발화하면 발화된 번호를 수신한다.	Native → Web
6	서비스 종료 요청 수신	음성명령(종료), 리모컨 나가기 키 수신 시 서비스 종료 요청을 전달한다.	Native → Web
7	서비스 종료처리 완료	서비스 종료 요청에 대한 내부 종료 처리가 완료되었음을 알린다.	Web → Native
8	Action 수신	대화에서 현재 실행중인 앱의 ActionCode (대화 SDK: Intent 명) 수신시 해당 Action 을 전달한다.	Native → Web
9	TTS 중단	TTS 를 중단한다.	Web → Native
10	음성 필터 설정	Keyword Spotting(예: 기가지니), 이후 발화 내용 중 App 에서 Catch 하고자 하는 단어를 설정한다.	Web → Native
11	음성 필터 삭제	음성 필터 설정을 제거한다.	Web → Native
12	음성 필터 결과 수신	음성 필터에 설정한 단어가 발화 내용에 포함될 경우 콜백으로 전달한다.	Native → Web

13	화자식별 목소리 등록 수신	화자식별/인증 목소리 등록 결과를 수신한다.	Native → Web
14	음성인식 도메인 설정	음성인식 도메인을 설정한다.	Native → Web
15	도메인설정 타임아웃 수신	도메인 설정 타임아웃 이벤트를 수신한다.	Native → Web

3.3 통화 API 리스트 정보 (Namespace: gigagenie.call)

순번	서비스명	설명	비고
1	통화 요청	입력한 번호로 GiGA Genie 의 통화 서비스를 요청한다.	Web → Native

3.4 데이터 관리 (Namespace: gigagenie.appdata)

순번	서비스명	설명	비고
1	데이터네임스페이스 생성 API	G-Box 의 특정 앱에 네임스페이스를 생성한다.	Web → Native
2	데이터네임스페이스 삭제 API	G-Box 의 특정 앱에 생성된 네임스페이스를 삭제한다.	Web → Native
3	데이터조회 API	G-Box 의 특정 앱의 특정 네임스페이스의 특정 Key 에 대한 데이터를 조회한다.	Web->Native
4	데이터설정 API	G-Box 의 특정 앱의 특정 네임스페이스의 특정 Key 에 대한 데이터를 설정한다.	Web->Native
5	데이터삭제 API	G-Box 의 특정 앱의 특정 네임스페이스의 특정 Key 에 대한 Key-Data 를 삭제한다.	Web->Native
6	Key 찾기 API	G-Box 의 특정 앱/네임스페이스에서 text 로 Key 를 조회한다.	Web-Native

3.5 G-Box 정보 관리 (Namespace: gigagenie.appinfo)

※ 현재 대기/동작 모드 전환 API 는 제공되지 않습니다.

대신에 화면을 활용하려면 TTS API 를 활용하여 TV 를 키라는 안내 멘트는 제공 가능합니다.

순번	서비스명	설명	비고
1	ContainerID 조회	G-Box 의 해당 App 에 대한 Container ID 를 조회한다.	Web → Native
2	AuthKey 조회	G-Box 의 해당 App 에 설정된 AuthKey 를 조회한다.	Web -> Native
3	AuthKey 설정	G-Box 의 해당 App 의 AuthKey 를 설정한다.	Web -> Native
4	AuthKey 삭제	G-Box 의 해당 App 의 AuthKey 를 삭제한다.	Web -> Native
5	Push 보내기	Companion App 또는 다른 G-Box 에 Push 메시지를 보낸다.	Web -> Native -> Mobile App
6	Screen 제한 조회 API	Screen 이용이 제한되는지 조회한다.	Web->Native
7	우리집 주소 조회	Companion App 에서 설정한 우리집 주소를 조회한다.	Web->Native
8	User 정보 조회	ContainerID 에 대한 User 정보를 조회한다.	Web->Native
9	User 설정 정보 조회	사용자가 설정한 정보를 조회한다.	Web->Native
10	주소록 조회	간편송금등을 위한 주소록을 조회한다.	Web->Native
11	상세주소 조회	Companion App 에서 설정한 우리집 상세 주소를 조회한다.	Web->Native
12	계정변경 Event 수신	계정변경 이벤트를 수신한다.	Web->Native

3.6 Media API (Namespace: gigagenie.media)

순번	서비스명	설명	비고
1	Mute 이벤트	Mute/Mute 해제 이벤트 전달	Native → Web
2	소프트키보드 표시	화면에 소프트 키보드를 표시한다.	Web->Native
3	리모콘키 이벤트 수신	리모콘키 이벤트를 수신한다.	Native → Web
4	음성 녹음 API	음성 녹음 및 녹음 데이터를 전달한다.	Web->Native
5	음성 녹음 수신	음성 녹음 API 에 의한 녹음 결과를 받는 Event Callback 이다.	Native → Web
6	음성 녹음 중지	음성 녹음을 중지한다.	Web->Native

7	Channel 생성	기가지니와 P2P 로 데이터를 전송하기 위한 Channel 을 생성한다.	Web->Native
8	Channel 제거	생성한 Channel 을 제거한다.	Web->Native
9	WebRTC Audio 연결	생성한 channel 에 WebRTC 프로토콜 기반 Audio Stream 을 연결한다.	Web->Native
10	WebRTC Audio 연결 해제	생성한 webrtcudio 연결을 해제한다.	Web->Native
11	Channel 연결 유지	생성한 Channel 연결을 유지한다.	Web->Native
12	Channel 이벤트 수신	생성한 Channel 의 이벤트를 수신한다.	Native → Web

3.7 Payment API (Namespace: gigagenie.payment)

순번	서비스명	설명	비고
1	OTV 사용자 확인	OTV 결제를 위한 사용자 인증 및 상태 체크를 요청한다.	Web->Native
2	OTV 결제 토큰 요청	OTV 결제 토큰을 요청한다.	Web->Native
3	OTV 결제 인증-결제 토큰 요청	OTV 결제를 위한 인증-결제 토큰을 요청한다.	Web->Native

3.8 영상인식 API (Namespace: gigagenie.vision)

순번	서비스명	설명	비고
1	영상인식 서비스 시작	영상인식 서비스를 시작한다	Web->Native
2	영상인식 서비스 중지	영상인식 서비스를 중지한다.	Web->Native
3	영상인식 정보 수신	영상인식 정보를 수신한다.	Native -> Web

Admin api??

4.API 설명

4.1 API 초기화 (gigagenie.init)

4.1.1 API 초기화 (gigagenie.init)

GiGA Genie Service API 를 초기화 한다. GiGA Genie Service API 를 이용하기 위해서는 API 초기화가 반드시 진행되어야 한다. Init 이 성공하지 않은 경우 다른 API 가 호출되지 않으며, 다른 API 호출시 406 오류로 리턴한다.

- function init(options,callback)
- options 은 다음과 같이 설정한다.
 - options.apikey : String, Mandatory 이며 개발자 사이트에서 받은 apikey 를 설정해야 하며, 등록된 apikey 가 아닌 경우에는 오류가 발생한다.
 - options.keytype : String, Mandatory 이며 api key 유형으로 다음의 유형을 지원한다.
 - ◆ options.keytype="GBOXDEVM" : G-Box 개발 키 (G-Box 개발 키 (개발자모드가 설정되어 있어야 합니다.)
 - ◆ options.keytype="GBOXCOMM" : G-Box 상용 키
- result_cd 가 200 으로 initialization 이 성공하지 않았을 경우 API 는 동작하지 않는다. API 사용전에 callback 에서 result_cd 가 200 으로 리턴했는지 확인해야 한다.
- result_cd 가 200 인 경우 다음의 값이 전달된다.
 - extra.sdkversion : String, Mandatory 로 이용가이드 버전이 전달된다.
 - extra.devicetype : String, Mandatory 로 디바이스 유형이 전달된다. 디바이스 유형은 다음과 같다. (sdkversion 1.4 에서 지원)
 - ◆ extra.devicetype=="GGENIE" : 기가지니
 - extra.displayflag : String, Mandatory 로 해당 디바이스가 화면을 지원하는지 여부를 전달한다. "Y" 의 경우 display 가 지원되며, "N"의 경우 display 가 지원되지 않는다.

- 사용 예

```
//callback 방식
var options={};
options.apikey="asdnblkb1231lkj3nrlkjlksfsdf";
```

```
options.keytype="GBOXDEVM"; // 개발자 모드를 설정하고 사용하세요.
//options.keytype="GBOXCOMM"; // 개발자센터에서 승인이 되어야 사용하실 수 있습니다.
gigagenie.init(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("initialize success");
    };
});
```

4.1.2 화자인식된 ContainerId 로 변경 (gigagenie.init.changeContainerId)

화자인식된 ContainerId 로 App 실행 환경을 변경한다. 가장 최근에 화자 인증된 ContainerId 로만 변경 가능하다. ContainerID 로 변경하면 변경한 ContainerID 의 AuthKey 와 AppData 를 조회 가능하며 변경된 ContainerID 의 User 에 대한 userNickName 과 비밀번호 설정 유무로 조회된다.

- function changeContainerId(options,callback)
- options 은 다음과 같이 설정한다.
 - options.containerId : 변경하고자 하는 ContainerId
- result_cd 는 다음과 같이 정의된다.
 - 200: 변경 성공
 - 404: 화자인증된 ContainerId 가 아님
 - 500: 시스템 Error
- 사용 예

```
//callback 방식
var options={};
options.containerId="asdnblkb1231lkj3nrlkjlksfsdf";
gigagenie.init.changeContainerID(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("containerID changed");
    };
});

//promise 방식
var options={};
options.containerId="asdnblkb1231lkj3nrlkjlksfsdf";
gigagenie.init.changeContainerId(options).then(function(extra){
```

```

    console.log("containerid changed");
  }).catch(function(result_cd,result_msg,extra){
    console.log("containerid change fail");
  })
}

```

4.1.3 Application Pause/Resume Event API (gigagenie.init.onAppStatusChange)

어플리케이션이 Pause 되거나 Resume 될 경우에 이벤트를 알려준다. Pause 는 사용자가 "기가지니" 등으로 Key Word Spotting 을 하거나, 다른 Application(예: 결제 요청)을 호출하게 되면 Pause 가 발생한다. 이후 다시 어플리케이션이 실행되는 상황이 되면 Resume 이벤트가 발생한다. Pause 시에 TTS, 동영상 플레이 등이 중지되므로 Resume 이벤트를 받아서 어플리케이션 실행을 복구해야 한다.

- gigagenie.init.onAppStatusChange 콜백 구현
- 콜백은 function callback(extra) 로 구현해야 하며 extra 는 다음의 int 값이 전달된다.
 - extra.changeStatus : Number, Mandatory 로 다음의 값을 가진다.
 - ◆ extra.changeStatus=0: Resume 상태 진입
 - ◆ extra.changeStatus=1: Pause 상태 진입
 - extra.muteFlag: Number, Mandatory 로 다음의 값을 가진다.
 - ◆ extra.muteFlag=0: unmute 상태 (음악 등 재생 가능)
 - ◆ extra.muteFlag=1: mute 상태 (음악 등 재생 불가)
- 사용 예

```

gigagenie.init.onAppStatusChange=function(extra){
  switch(extra.changeStatus){
    case 0:
      //복구에 필요한 코드
      break;
    case 1
      //Pause 진입
      break;
    default:
      break;
  }
};

```


4.2 Voice API (gigagenie.voice)

4.2.1 음성인식 API (gigagenie.voice.getVoiceText)

사용자에게 음성 안내 후 음성 인식을 한다. TTS 출력중에 stopTTS 를 수신하면 TTS 와 음성인식 모두 종료된다. 음성인식이 진행되면 해당 음성인식을 종료할 수 없다. (Timeout 으로 처리)

- function getVoiceText(options,callback)
- options 은 다음과 같이 설정한다.
 - options.mode : Int, Optional 로 음성 인식 결과의 후처리를 설정한다. 이 값이 없으면 default mode(mode=0) 으로 동작한다.
 - ◆ options.mode=0 : 음성인식 결과를 전달한다.
 - ◆ options.mode=1: 음성 인식 결과를 대화서버로 전달한다.
음성인식결과는 undefined 로 전달되며, 이후, onActionEvent 로 이벤트를 받거나, 다른 앱이 실행된다.
 - ◆ options.mode=2: 확인 및 번호 선택에 대한 음성인식을 수행한다.
음성인식결과는 undefined 로 넘겨지며 이후 onVoiceCommand 로 확인에 대한 인식 결과인 "confirm", "cancel"을 전달한다. 확인으로 해석되지 않는 경우 결과값이 전달되지 않는다. 확인 취소는 다음의 발화에 의해 해석된다.
 - 확인 : "확인", "그래", "응" 으로 발화한 경우이다.
 - 취소: "취소", "아니", "아니요"로 발화한 경우이다.번호 선택에 대한 음성인식결과는 undefined 으로 넘겨지며 이후 onSelectedNumber 에 인식된 번호를 전달된다. 번호가 인식되지 않으면 결과값이 전달되지 않는다. 번호는 1-6 번까지만 전달된다. (
 - options.voicemsg : String, Optional 으로 voicemsg 내용을 TTS 로 재생한 후 음성 인식을 시작한다.
 - options.ttslanguage : Number, Optional 으로 voicemsg 를 TTS 로 재생할 언어를 설정한다. 이 값이 설정되어 있지 않으면 기본으로 한국어를 설정한다.
 - ◆ options.ttslanguage=0 : 한국어 설정(default)

- ◆ options.ttslanguage=1 : 영어
- ◆ options.ttslanguage=2 : 중국어
- ◆ options.ttslanguage=3 : 일본어
- options.voicelanguage : Number, Optional 으로 음성인식에서 사용할 언어를 설정한다. 이 값이 설정되어 있지 않으면 기본으로 한국어를 설정한다.
 - ◆ options.voicelanguage=0 : 한국어 설정(default)
 - ◆ options.voicelanguage=1 : 영어
 - ◆ options.voicelanguage=2 : 중국어
 - ◆ options.voicelanguage=3 : 일본어
- options.timeout: Number, Optional 로 음성인식 대기 시간(초)을 설정한다. 10초 이하로만 설정 가능하며, 이 값이 없으면 10초가 기본 대기 시간이다.
- options.text: String, Optional 로 음성인식의 정확도를 계산하기 위하여 인식되어야 하는 Text 이다. 정확도는 voicelanguage 가 1(영어)인 경우에 대해서만 지원된다.
- result_cd 에 다음의 값이 전달된다.
 - 200 : 음성인식 성공, extra.voicetext 에 인식된 결과가 전달된다.
 - 500 : Timeout 발생 등으로 음성 인식 실패, extra.voicetext 에는 "ASR session timeout" 가 전달된다.
 - 501 : KWS(KeyWord Spotting)에 의해 정지됨
 - 502 : 재생 실패
 - 503: stopTTS 에 의한 TTS 종료
 - 504: mute 상태로 TTS 재생 불가
 - 505: 처리 용량 초과(다국어 음성인식 중 서버 용량 초과시 발생)
- extra 값에 다음이 설정되어 전달된다.
 - extra.voicetext : String, Mandatory 로 음성인식된 Text 가 전달된다.
 - extra.accuracy : Number, Optional 로 인식된 결과의 정확도이다. (options.text 가 설정되어 있고 options.voicelanguage 가 1 인 경우에만 설정되어 리턴됨)
- 사용 예

```
//callback 방식으로, 즉시 음성 인식을 할 경우
gigagenie.voice.getVoiceText(null,function(result_cd,result_msg,extra){
    if(result_cd===200){
```

```

        console.log("Received Text is "+extra.voicetext);
    };
});
//callback 방식으로, TTS로 재생한 후 음성 인식을 할 경우
var options={};
options.voicemsg="생년월일을 말해 주세요"
gigagenie.voice.getVoiceText(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("Received Text is "+extra.voicetext);
    };
});
});

```

4.2.2 TTS API (gigagenie.voice.sendTTS)

입력 Text 를 사용자에게 음성으로 전달한다. stopTTS API 에 의해서 중지된다.

- function sendTTS(options,callback)
- options 은 다음과 같이 설정한다.
 - options.ttstext : String, Mandatory 로 ttstext 를 입력 Text 로 TTS 를 실행한다. ttstext 에 다음의 TAG 를 붙일 수 있다.
 - ◆ [DS] ~ [/DS] : TAG 사이의 숫자를 끊어 읽는다. 숫자는 기본적으로 붙여서 읽도록 되어 있다.
 - 예: [DS]1234[/DS] 는 "일이삼사"로, 1234 는 "천이백삼십사"로 플레이한다.
 - ◆ [IS] ~ [/IS] : TAG 사이의 알파벳, 숫자 등을 끊어 읽는다.
 - 예: [IS]name[/IS] 는 "엔에이엠이"로 name 은 "네임"으로 플레이한다.
 - ◆ [P1] : 100ms 의 묵음을 삽입한다.
 - ◆ [P2] : 200ms 의 묵음을 삽입한다.
 - ◆ [P3] : 300ms 의 묵음을 삽입한다.
 - options.language : Number, Optional 으로, 재생할 언어를 설정한다. 이 값이 설정되어 있지 않으면 기본으로 한국어를 설정한다.
 - ◆ options.language=0 : 한국어 설정(default)
 - ◆ options.language=1 : 영어
 - ◆ options.language=2 : 중국어

◆ options.language=3 : 일본어

- extra 값은 null 이다.
- result_cd 에 다음의 값이 전달된다.
 - 200 : TTS 재생 성공
 - 409 : sendTTS 로 TTS 재생중임
 - 501 : KWS(KeyWord Spotting)에 의해 정지됨
 - 502 : 재생 실패
 - 503: 다른 App 에서 TTS 중지 메시지를 보냄(stopTTS 에 의한 종료)
 - 504: mute 상태로 TTS 재생 불가
- 사용 예

```
//callback 방식
var options={};
options.ttstext="안녕기가지니[P2]일반숫자[P2]1234[P2]끊어읽기[DS]1234[DS]";
gigagenie.voice.sendTTS(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        //do next action
    } else {
        //result_cd, result_msg 오류 전달
    };
});
```

4.2.3 음성명령수신 API (gigagenie.voice.onVoiceCommand)

다음페이지, 이전페이지 등 음성 명령 코드를 수신한다.

- gigagenie.voice.onVoiceCommand 콜백 구현
- 콜백은 function callback(event,extra) 로 구현해야 하며 event 는 다음의 음성 명령에 대해서 Text 로 전달한다. 확인, 취소는 4.2.1 의 getVoiceText 의 mode=2 에 대한 응답을 전달한다.
 - 다음페이지: nextPage
 - 이전페이지: prevPage
 - 다음으로: naviNext
 - 이전으로: naviPrev
 - 확인:confirm
 - 취소:cancel

○ 사용 예

```
gigagenie.voice.onVoiceCommand=function(event){
  switch(event){
    case 'nextPage':
      //navigate next page
      break;
    case 'prevPage':
      //navigate prev page
      break;
    default:
      break;
  }
};
```

4.2.4 음성명령 수신설정 API (gigagenie.voice.setKwsVoiceRecv)

음성명령은 기본적으로는 번호 선택은 OTV 채널 선택, 확인/취소 등은 기본 메시지가 전달된다. 이를 setKwsVoiceRecv 로 3rd Party Web 에서 전달받을 수 있다.

(음성 선택 번호 수신) 번호로 선택 가능한 UI 에서 사용자가 KWS 이후 발화한 번호를 3rd Party App 에 전달하도록 설정한다.(디폴트는 OTV 채널 실행) 앱을 다시 실행하거나, 다른 웹페이지로 이동하면 디폴트로 변경된다. 4.2.1 음성인식 API 의 mode 2 번의 경우 3rd Party App 에서 직접 요청을 하는 상황에서 사용되며 4.2.4 는 KWS 이후 발화 해석에서 번호 선택시 4.2.5 의 onSelectedIndex 로 전달한다.

(음성 확인/취소 수신) KWS 이후 발화한 내용이 확인/취소에 해당할 경우 해당 내용을 3rd Party App 에 전달하도록 설정한다. 앱을 다시 실행하거나, 다른 웹페이지로 이동하면 디폴트로 변경된다. 4.2.1 음성인식 API 의 mode 2 번의 경우 3rd Party App 에서 직접 요청하는 상황에서 사용되며 4.2.4 는 KWS 이후 발화 해석에서 확인/취소로 수신시 4.2.3 의 onVoiceCommand 로 전달한다. 확인 및 취소 발화는 4.2.1 의 내용과 동일하다.

- function setKwsVoiceRecv(options,callback)
- options 은 다음의 값을 가진다.
 - options.flag : number, mandatory 로 다음의 값을 가진다.
 - ◆ options.flag=0 : OTV 채널 실행으로 변경

- ◆ options.flag=1 : ContainerApp 에서 음성선택번호 및 확인/취소 수신
- result_cd 에 다음의 값이 전달된다.
 - 200 : 성공
 - 500: 시스템 에러
- Extra 는 null 이다.
- 사용 예

```
var options={};
options.flag=1;
gigagenie.voice.setKwsVoiceRecv (options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("setKwsVoiceRecv success");
    }
});
```

4.2.5 음성선택번호수신 API (gigagenie.voice.onSelectedIndex)

4.2.4 의 음성 선택번호 수신 설정 API 호출에 따라 사용자가 KWS 이후 발화한 번호 정보를 전달한다. 번호는 1-6 번까지만 인식 가능하다.

- gigagenie.voice.onSelectedIndex 콜백 구현
- 콜백은 function callback(event) 로 구현해야 하며 event 는 선택된 숫자가 넘겨진다.
- 사용 예

```
gigagenie.voice.onSelectedIndex=function(event){
    switch(event){
        case 1:
            //1번 선택
            break;
        case 2:
            //2번 선택
            break;
        case 3:
            //3번 선택;
        case 4:
            //4번 선택
            break;
```

```

    default:
        break;
});

```

4.2.6 서비스종료수신 API (gigagenie.voice.onRequestClose)

음성명령(종료, 닫기 발화에 따른)과, 리모컨 나가기 키 클릭시 서비스 종료 이벤트를 전달한다. onRequestClose 콜백에서 종료 처리를 할 경우 마지막에 gigagenie.voice.svcFinished 를 호출해 줘야 한다.

- gigagenie.voice.onRequestClose 콜백 구현
- 콜백은 function callback() 로 구현해야 한다.
- 사용 예

```

gigagenie.voice.onRequestClose=function(){
    // 3rd party 내부 웹 서비스 종료 처리...
    ....
    gigagenie.voice.svcFinished(null,function(result_cd,result_msg,extra){
    };
});

```

4.2.7 서비스종료 API (gigagenie.voice.svcFinished)

음성명령(닫기)이나 리모컨 나가기 키에 의한 서비스 종료 요청을 수신하거나 화면상의 '닫기' 버튼 클릭 시 자체 종료처리(e.g. 앱 세션 정리)를 진행하고 종료 처리한다. 종료 요청을 수신하고 자체 종료 처리 후 서비스종료 API 를 명시적으로 호출해 주어야 해당 서비스가 종료된다.

- function svcFinished(options,callback)
- options 은 null 이다.
- extra 값은 null 이다.
- 사용 예

```
//callback 방식
gigagenie.voice.svcFinished(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("service finish success");
    }
});
//promise 방식
gigagenie.voice.svcFinished(options).then(function(extra){
    console.log("service finish success");
}).catch(function(result_cd,result_msg,extra){
    console.log("service finish fail");
})
```

4.2.8 Action 수신 API (gigagenie.voice.onActionEvent)

등록된 Action 에 대한 수신 Event 이다. 등록된 Action 은 대화 SDK 를 통해서 ActionCode (대화 SDK: Intent 명)가 등록되어 있어야 한다. 현재 실행중인 appid 에 대한 Action 수신시 Event 를 전달한다.

- gigagenie.voice.onActionEvent 콜백 구현
- 콜백은 function callback(extra) 로 구현해야 하며 extra 에는 actioncode 와 actionpath 및 parameter 가 전달된다.
 - actioncode : 발화 Intent 와 동일
 - actionpath : 등록한 이동 Path
 - voiceid : 화자인식된 ContainerID 값(인식 안된 경우 "UNKNOWN", 화자식별등록이 안되어 있으면 "NONE" 전달)
 - uword : 발화한 문구
 - parameter : 대화서버에서 전달하는 parameter JSON 값
- onActionEvent 콜백이 정의되지 않으면 해당 actionpath 로 자동 이동한다.
- 사용 예

```
gigagenie.voice.onActionEvent=function(extra){
    switch(extra.actioncode){
        case 'QueryHello':
            //QueryHello 처리
            break;
        case 'QueryWorld':
```



```
//QQueryWorld 처리
break;
default:s
break;
});
```

4.2.9 TTS 중단 API (gigagenie.voice.stopTTS)

TTS 를 중단한다. 음성인식도 중지된다.

- function stopTTS(options,callback)
- options 은 null 이다.
- extra 값은 null 이다.
- result_cd 에 다음의 값이 전달된다.
 - 200 : 중단 성공
 - 404 : TTS, 음성인식 실행 중이 아님
 - 500 : 실행 오류
- 사용 예

```
//callback 방식
gigagenie.voice.sendTTS(null,function(result_cd,result_msg,extra){
  if(result_cd===200){
    //do next action
  } else {
    //result_cd, result_msg 오류 전달
  };
});
```

4.2.10 음성 필터 설정 API (gigagenie.voice.setVoiceFilter)

기가지니의 Keyword Spotting(예: 기가지니) 이후 발화된 내용중에서 App 에서 전달받고자 하는 단어를 설정한다. 음성 인식 결과는 대화 SDK 를 거치지 않고 이벤트 수신을 통해 전달한다. 권한을 가진 appid 만 호출 가능하다.

※ 설정한 단어가 첫 번째 단어로 인식될 때만 이벤트로 전달한다.

예: ['정답'] 으로 등록했을 경우 '정답 1 번'은 수신, '1 번 정답'은 수신 안됨

- gigagenie.voice.setVoiceFilter

- options 은 다음과 같이 설정한다.
 - options.keyword : String Array, Mandatory 로 Keyword 를 설정한다.
- 사용 예

```
//callback 방식
var options={};
options.keyword=["정답","힌트"];
gigagenie.voice.setVoiceFilter(options,function(result_cd,result_msg,extra){
  if(result_cd===200){
    console.log("Keyword setting is success.");
  };
});
//promise 방식
var options={};
options.callnumber=["정답","힌트"];
gigagenie.voice.setVoiceFilter(options).then(function(extra){
  console.log("Keyword setting is success.");
}).catch(function(result_cd,result_msg,extra){
  //
})
```

4.2.11 음성 필터 삭제 API (gigagenie.voice.resetVoiceFilter)

음성필터 설정을 초기화 한다. 권한을 가진 appid 만 호출 가능하다.

- gigagenie.voice.resetVoiceFilter
- options 는 null 이다.
- 사용 예

```
//callback 방식
var options={};
gigagenie.voice.resetVoiceFilter(options,function(result_cd,result_msg,extra){
  if(result_cd===200){
    console.log("Reset Keyword");
  };
});
//promise 방식
var options={};
gigagenie.voice.resetVoiceFilter(options).then(function(extra){
  console.log("Reset Keyword");
}).catch(function(result_cd,result_msg,extra){
  //
})
```

4.2.12 음성 필터 결과 수신 API (gigagenie.voice.onVoiceFilterMsg)

등록된 Keyword 에 대한 수신 Event 이다. Keyword 는 Keyword 설정 API 를 통해 등록된 값이다. 권한을 가진 appid 만 호출 가능하다.

- gigagenie.voice.onVoiceFilterMsg 콜백 구현
- 콜백은 function callback(extra) 로 구현해야 하며 message 에는 Keyword 를 포함한 음성 인식 Text 가 전달된다.
 - message : 음성 인식 Text
- 사용 예

```
//callback 방식
gigagenie.voice.onVoiceFilterMsg=function(message){
    console.log("Received Text is "+message);
});
//promise 방식
gigagenie.voice.onVoiceFilterMsg(null).then(function(message){
    console.log("Received Text is "+message);
}).catch(function(result_cd,result_msg,message){
    //
})
```

4.2.13 화자식별 목소리 등록 이벤트 수신 API (gigagenie.voice.onRegToneResults)

화자인식(식별/인증) 목소리 등록 요청 Push 를 보내고 해당 결과를 수신하는 콜백이다.

- gigagenie.voice.onRegToneResults 콜백 구현
- 콜백은 function callback(extra) 로 구현해야 하며 extra 이는 다음의 값이 전달된다.
 - extra.regResult: String, Mandatory 로 등록 이벤트 이다. 다음의 값을 가진다
 - ◆ extra.regEvent="01": (화자식별) 목소리 등록 모드
 - ◆ extra.regEvent="03": (화자식별) 목소리 종료 모드
 - ◆ extra.regEvent="04": (화자식별) 목소리 취소 모드
 - ◆ extra.regEvent="11": (화자인증) 목소리 등록 모드
 - ◆ extra.regEvent="13": (화자인증) 목소리 종료 모드
 - ◆ extra.regEvent="14": (화자인증) 목소리 취소 모드
- 사용 예

```
//callback 방식
gigagenie.voice.onReqToneResult=function(extra){
    if(extra.reqEvent=="01"){
    } else if ( .....
        ...
    }
});
```

4.2.14 음성인식 도메인 설정 API (gigagenie.voice.setVoiceRecogDomain)

기가지니의 음성인식을 위한 도메인을 설정한다. 앱이 다시 시작되거나, 로드되는 페이지가 바뀔 경우 그리고 5 분이 지나면 공통으로 초기화 된다. 특정 시간이 지나서 초기화 되는 경우 4.2.15 onVocDomainTimeout 이 발생한다.

- function setVoiceRecogDomain
- options 은 다음과 같이 설정한다.
 - options.domainCode : Number, Mandatory 다음의 값을 가진다.
 - ◆ options.domainCode = 0 : 공통
 - ◆ options.domainCode = 1 : 금융
 - ◆ options.domainCode = 2 : 비밀번호
 - ◆ options.domainCode = 3 : 전화번호
- result_cd 에 다음의 값이 전달된다.
 - 200 : 성공
 - 500: 시스템 에러
- Extra 는 null 이다.
- 사용 예

```
//callback 방식
var options={};
options.domainCode=1;
gigagenie.voice.setVoiceRecogDomain(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("Keyword setting is success.");
    }
});
```

4.2.15 도메인 설정 타임아웃 Event API (gigagenie.voice.onVocDomainTimeout)

4.2.14 setVoiceRecogDomain 에서 공통 도메인 이외의 도메인 설정에 대한 유효 시간이 만료 되었음을 알려주는 이벤트이다. 공통 도메인 설정의 경우 타임아웃이벤트가 발생하지 않는다.

- gigagenie.voice.onVocDomainTimeout 콜백 구현
- 콜백은 function callback(extra) 로 extra.flag 이 boolean 으로 true 가 전달된다.
- 사용 예

```
//callback 방식
gigagenie.voice.onVocDomainTimeout=function(extra){
    console.log("onVocDomainTimeout Occurred");
};
```

4.3 통화 API (gigagenie.call)

4.3.1 통화요청 API (gigagenie.call.requestCall)

입력한 번호로 GiGA Genie 통화 서비스를 요청한다.

- function requestCall(options,callback)
- options 은 다음과 같이 설정한다.
 - options.callnumber : String, Mandatory 로 callNumber 에 통화를 요청한다.
 - options.calltype : Int, Mandatory 로 0 은 음성통화, 1 은 영상통화를 요청한다.
- 사용 예

```
//callback 방식
var options={};
options.callnumber="021234567";
options.calltype=0;
gigagenie.call.requestCall(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("Received Call Number is "+extra.callnumber);
    }
});

//promise 방식
var options={};
options.callnumber="021234567";
options.calltype=0;
```

```
gigagenie.call.requestCall(options).then(function(extra){
    console.log("Received Call Number is "+extra.callNumber);
}).catch(function(result_cd,result_msg,extra){
    //
})
```

4.4 데이터관리 API (gigagenie.appdata)

4.4.1 데이터네임스페이스생성 API (gigagenie.appdata.createNameSpace)

특정 이름의 데이터네임스페이스를 생성한다..

- function createNameSpace(options,callback)
- options 은 다음과 같이 설정한다.
 - options.namespace : String, Mandatory 로 생성하고자 하는 NameSpace 이름이다.
 - options.shareflag : String, Mandatory 로 해당 NameSpace 에 대한 공유 여부를 설정한다. (이용가이드 V1.0 에서는 options.shareflag 는 'N' 만 지원)
 - ◆ options.shareflag = 'N' : App 전용 데이터 생성
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 409: 이미 존재하는 namespace 임
 - 500: 시스템 Error
- 사용 예

```
//callback 방식
var options={};
options.namespace='userappprofile';
options.shareflag='N';
gigagenie.appdata.createNameSpace(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("Namespace creation is success.");
    } else {
        console.log("Namespace creation is fail.");
    }
})
```

```

    }
  });

  //promise 방식
  var options={};
  options.namespace='userappprofile';
  options.shareflag='N';
  gigagenie.appdata.createNameSpace (options).then(function(extra){
    console.log("Namespace creation is success.");
  }).catch(function(result_cd,result_msg,extra){
    console.log("Namespace creation is fail.");
  })
}

```

4.4.2 데이터네임스페이스삭제 API (gigagenie.appdata.destroyNameSpace)

특정 이름의 데이터네임스페이스를 생성한다. 네임스페이스 삭제 시, 해당 네임스페이스의 모든 데이터도 삭제된다.

- function destroyNameSpace(options,callback)
- options 은 다음과 같이 설정한다.
 - options.namespace : String, Mandatory 로 삭제하고자 하는 NameSpace 이름이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 404: 존재하지 않는 네임스페이스임
 - 500: 시스템 Error
- 사용 예

```

//callback 방식
var options={};
options.namespace='userappprofile';
gigagenie.appdata.destroyNameSpace(options,function(result_cd,result_msg,extra){
  if(result_cd===200){
    console.log("Namespace destroy is success.");
  } else {
    console.log("Error");
  }
});

```

```
//promise 방식
var options={};
options.namespace='userappprofile';
gigagenie.appdata.destroyNameSpace (options).then(function(extra){
    console.log("Namespace destroy is success.");
}).catch(function(result_cd,result_msg,extra){
    console.log("Namespace creation is fail.");
})
```

4.4.3 데이터 조회 API (gigagenie.appdata.getKeyData)

특정 이름의 데이터네임스페이스의 Key 에 대한 Data 를 조회한다.

- function getKeyData(options,callback)
- options 은 다음과 같이 설정한다.
 - options.namespace : String, Mandatory 조회하고자 하는 appdata Namespace
 - options.key : String, Mandatory 로 해당 NameSpace 에서 조회하고자 하는 데이터 Key
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 403: 해당 NameSpace 가 존재하지 않음
 - 404: 해당 Key 가 존재하지 않음
 - 500: 시스템 Error
- extra 값에 다음이 설정되어 전달된다.
 - extra.data : String, Mandatory 로 key 에 대한 text 데이터가 전달된다.
- 사용 예

```
//callback 방식
var options={};
options.namespace='userappprofile';
options.key='appusername';
gigagenie.appdata.getKeyData(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log(options.key + ":" + extra.data);
    } else {
        console.log("Error");
    }
})
```



```

    }
  });
  //promise 방식
  var options={};
  options.namespace='userappprofile';
  options.key='appusername';
  gigagenie.appdata.getKeyData (options).then(function(extra){
    console.log(options.key + ":" + extra.data);
  }).catch(function(result_cd,result_msg,extra){
    console.log("Error");
  })
}

```

4.4.4 데이터 설정 API (gigagenie.appdata.setKeyData)

특정 이름의 데이터네임스페이스에 Key 의 Data 를 설정한다.

- function setKeyData(options,callback)
- options 은 다음과 같이 설정한다.
 - options.namespace : String, Mandatory 설정하고자 하는 key 의 appdata Namespace
 - options.key : String, Mandatory 로 해당 NameSpace 에서 설정하고자 하는 데이터의 Key
 - options.data : String, Mandatory 로 해당 NameSpace 의 Key 에 설정된 데이터
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 403: 해당 NameSpace 가 존재하지 않음
 - 406: 저장 Key 개수 초과
 - 413: 데이터 용량 초과
 - 500: 시스템 Error
- 사용 예

```

//callback 방식
var options={};
options.namespace='userappprofile';
options.key='appusername';

```

```

options.data='helloappdata';
gigagenie.appdata.setKeyData(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log(options.key + ":" + extra.data + " is set");
    } else {
        console.log("Error");
    }
});
//promise 방식
var options={};
options.namespace='userappprofile';
options.key='appusername';
options.data='helloappdata';
gigagenie.appdata.setKeyData (options).then(function(extra){
    console.log(options.key + ":" + extra.data + " is set");
}).catch(function(result_cd,result_msg,extra){
    console.log("Error");
})

```

4.4.5 데이터 삭제 API (gigagenie.appdata.delKeyData)

특정 이름의 데이터네임스페이스에 Key 와 Data 를 삭제한다.

- function delKeyData(options,callback)
- options 은 다음과 같이 설정한다.
 - options.namespace : String, Mandatory 삭제하고자 하는 key 의 appdata Namespace
 - options.key : String, Mandatory 로 해당 NameSpace 에서 삭제하고자 하는 Key
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 403: 해당 NameSpace 가 존재하지 않음
 - 404: 해당 Key 가 존재하지 않음
 - 500: 시스템 Error
- 사용 예

```
//callback 방식
```

```
var options={};
options.namespace='userappprofile';
options.key='appusername';
gigagenie.appdata.delKeyData(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log(options.key + ":" + extra.data + " is deleted");
    } else {
        console.log("Error");
    }
});
//promise 방식
var options={};
options.namespace='userappprofile';
options.key='appusername';
gigagenie.appdata.delKeyData (options).then(function(extra){
    console.log(options.key + ":" + extra.data + " is deleted");
}).catch(function(result_cd,result_msg,extra){
    console.log("Error");
})
```

4.4.6 Key 찾기 API (gigagenie.appdata.findKeys)

특정 이름의 데이터네임스페이스에서 Key 를 찾는다.

- function findKeys(options,callback)
- options 은 다음과 같이 설정한다.
 - options.namespace : String, Mandatory 조회하는 key 의 appdata Namespace
 - options.keytext : String, Mandatory 로 해당 NameSpace 에서 조회하고자 하는 KeyText
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 403: 해당 NameSpace 가 존재하지 않음
 - 404: 해당 Key 가 존재하지 않음
 - 500: 시스템 Error
- extra 값에 다음이 설정되어 전달된다.

- extra.keys : String Array, Mandatory 로 해당 keytext 로 찾은 Key 배열이 전달된다.

- 사용 예

```
//callback 방식
var options={};
options.namespace='userappprofile';
options.keytext='appuser';
gigagenie.appdata.findKeys(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("Find Key Count=" + extra.keys.length);
        extra.keys.forEach(function(key){
            console.log("Key is " + key);
        });
    } else {
        console.log("Error");
    }
});

//promise 방식
var options={};
options.namespace='userappprofile';
options.keytext='appuser';
gigagenie.appdata.findKeys (options).then(function(extra){
    console.log("Find Key Count=" + extra.keys.length);
    extra.keys.forEach(function(key){
        console.log("Key is " + key);
    });
}).catch(function(result_cd,result_msg,extra){
    console.log("Error");
})
```

4.5 G-Box 정보 관리 API (gigagenie.appinfo)

4.5.1 ContainerId 조회 API (gigagenie.appinfo.getContainerId)

해당 앱에 대한 G-Box 의 ContainerId 를 조회한다.

- function getContainerId(options,callback)
- options 은 null 이다.

- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 값에 다음이 설정되어 전달된다.
 - extra.containerid: String, Mandatory 로 Container 의 ID 값이다.
- 사용 예

```
//callback 방식
var options={};
gigagenie.appinfo.getContainerId(options,function(result_cd,result_msg,extra){
  if(result_cd===200){
    console.log("The container id is " + extra.containerid);
  } else {
    console.log("getContainerId is fail.");
  }
});
//promise 방식
var options={};
gigagenie.appinfo.getContainerId(options).then(function(extra){
  console.log("The container id is " + extra.containerid);
}).catch(function(result_cd,result_msg,extra){
  console.log("getContainerId is fail.");
})
```

4.5.2 AuthKey 조회 API (gigagenie.appinfo.getAuthKey)

해당 앱에 대한 G-Box 의 AuthKey 를 조회한다.

- function getAuthKey(options,callback)
- options 은 null 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 404: AuthKey 설정되어 있지 않음
 - 500: 시스템 Error
- result_cd 가 200 이면 extra 값에 다음이 설정되어 전달된다.
 - extra.authkey: String, Mandatory 로 authKey 값이다.

- extra.settime: String, Mandatory 로 authKey 가 설정된 시간이다. (형식: YYYYMMDDhhmmss, 20170519184202)
- extra.duetime: String, Mandatory 로 authKey 의 만료 예정 시간이다.

○ 사용 예

```
//callback 방식
var options={};
gigagenie.appinfo.getAuthKey(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("Key value is " + extra.authkey);
        console.log("SetTime value is " + extra.settime);
        console.log("DueTime value is " + extra.duetime);
    } else if(result_cd===404){
        console.log("Key is not set.");
    } else {
        console.log("getAuthKey is fail.");
    }
});

//promise 방식
var options={};
gigagenie.appinfo.getAuthKey(options).then(function(extra){
    console.log("Key value is " + extra.authkey);
    console.log("SetTime value is " + extra.settime);
    console.log("DueTime value is " + extra.duetime);
}).catch(function(result_cd,result_msg,extra){
    if(result_cd===404){
        console.log("Key is not set.");
    } else {
        console.log("getAuthKey is fail.");
    }
})
```

4.5.3 AuthKey 설정 API (gigagenie.appinfo.setAuthKey)

해당 앱에 대한 G-Box 의 AuthKey 를 설정한다.

- function setAuthKey(options,callback)
- options 은 다음과 같다.

- options.authkey : String, Mandatory 로 설정하고자 하는 authkey 이다.
(자리수 제한 있음:256Byte, 2048Bit)
- options.duetime : String, Optional 로 설정하고자 하는 authkey 의 만료
시간이다. (형식: YYYYMMDDhhmmss, 20170519184202)
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 400: authkey 제한 초과
 - 500: 시스템 Error
- 사용 예

```
//callback 방식
var options={};
options.authkey='asdasldkjalskdasd';
options.duetime='20180519184202';
gigagenie.appinfo.setAuthKey(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("AuthKey Set is Success");
    } else {
        console.log("AuthKey Set is fail.");
    }
});

//promise 방식
var options={};
options.authkey='asdasldkjalskdasd';
options.duetime='20180519184202';
gigagenie.appinfo.setAuthKey(options).then(function(extra){
    console.log("AuthKey Set is Success");
}).catch(function(result_cd,result_msg,extra){
    if(result_cd===400){
        console.log("AuthKey Size exceeded.");
    } else {
        console.log("AuthKey Set is fail.");
    }
})
```

4.5.4 AuthKey 삭제 API (gigagenie.appinfo.delAuthKey)

해당 앱에 대한 G-Box 의 AuthKey 를 삭제한다.

- function delAuthKey(options,callback)
- options 은 null 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- 사용 예

```
//callback 방식
var options={};
gigagenie.appinfo.delAuthKey(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("AuthKey Deleting is Success");
    } else {
        console.log("AuthKey Deleting is fail.");
    }
});
//promise 방식
var options={};
gigagenie.appinfo.setAuthKey(options).then(function(extra){
    console.log("AuthKey Deleting is Success");
}).catch(function(result_cd,result_msg,extra){
    console.log("AuthKey Deleting is fail.");
})
```

4.5.5 Push 보내기 API (gigagenie.appinfo.sendPushMsg)

해당 GiGA Genie 에 등록된 GiGA Genie Companion App 에 Push 메시지를 보낸다.

Push 메시지는 모바일 단말에 푸시 팝업이 보여지며 해당 팝업 클릭시 해당 msg(URL)로 모바일 단말의 웹 브라우저를 실행시킨다.

- function sendPushMsg(options,callback)
- options 은 다음과 같다.
 - options.target : String, Mandatory 로 다음을 설정한다.
 - ◆ COMP_APP: 해당 G-Box 에 연계된 Companion App 을 타겟으로 한다.

- options.msgtype: String, Mandatory 로 다음을 설정한다.
 - ◆ EXEC_WEB : Web Browser 의 URL 에 해당 msg 를 전달-실행한다.
 - ◆ EXEC_CALL : 모바일 통화연결 요청 Push 를 전송한다.
 - ◆ EXEC_TONE : 화자식별/ 등록 요청한다.
 - ◆ EXEC_AGREE: 기가지니앱 홈 화면으로 이동하는 Push 를 보낸다.
 - ◆ EXEC_BANK: 기가지니앱 은행 화면으로 이동하는 Push 를 보낸다.
- options.msg : String, Mandatory 로 보내고자 하는 msg 이다. EXEC_WEB 의 경우 web url 을, EXEC_CALL 의 경우 통화에 설정할 전화번호를 설정한다.
- options.popuptext : String, Optional 로 팝업 문구이다. (android 만 적용됨), null 일 경우 Default 메시지가 전달된다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- 사용 예

```
//callback 방식
var options={};
options.target='COMP_APP';
options.msgtype='EXEC_WEB';
options.msg='http://hello.world/hello?msg=world';
gigagenie.appinfo.sendPushMsg(options,function(result_cd,result_msg,extra){
  if(result_cd===200){
    console.log("Sending Push msg to companion app is Success");
  } else {
    console.log("Sending Push msg to companion app Set is fail.");
  }
});
```

4.5.6 Screen 제한 조회 API (gigagenie.appinfo.getScreenLimit)

Screen 이용 제한 상태를 조회한다.

- function getScreenLimit(options,callback)
- options 은 null 이다.
- result_cd 는 다음과 같이 정의된다.

- 200: 성공
- 500: 시스템 Error
- extra 에는 다음의 값이 전달된다.
 - limit : true 이면 Screen 이용 불가 상태(대기모드에서 호출)
- 사용 예

```
//callback 방식
var options={};
gigagenie.appinfo.getScreenLimit(options,function(result_cd,result_msg,extra){
  if(result_cd===200 && extra.limit===true){
    console.log("Display is available");
  } else {
    console.log("getDisplayAvailable is fail.");
  }
});
```

4.5.7 우리집 주소 조회 API (gigagenie.appinfo.getAddress)

GiGA Genie 에 설정된 우리집 주소를 조회한다. 우리집 주소는 Companion App 에서 설정한 주소 정보이다.

- function getAddress(options,callback)
- options 은 null 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 404: 저장된 주소값 없음
 - 500: 시스템 Error
- result_cd 가 200 이면 extra 값에 다음이 설정되어 전달된다.
 - extra.address : 우리집 주소로 설정된 주소 String
- 사용 예

```
//callback 방식
gigagenie.appinfo.getAddress(null,function(result_cd,result_msg,extra){
  if(result_cd===200){
    var address=extra.address;
    console.log("GiGA Genie Home Address="+address+" "+subaddress);
  } else if(result_cd===404){
```

```

var address=extra.address;
console.log("Address is not found");
} else {
    console.log("Getting Home Address fail.");
}
});

```

4.5.8 User 정보 조회 API (gigagenie.appinfo.getUserInfo)

현재 ContainerID 에 대한 User 정보를 조회한다.

- function getUserInfo(options,callback)
- options 은 null 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- result_cd 가 200 이면 extra 값에 다음이 설정되어 전달된다.
 - extra.username : 사용자 닉네임 String
 - extra.ispin : 유저 비밀번호 설정 여부 Number (0 은 미설정, 1 은 설정)
 - extra.kwsid : 호출어 설정 값 String (이 값은 containerid 전환에 영향을 받지 않으며, 기가지니 단말에 설정된 사용자의 kwsid 값이 넘겨진다.)
 - extra.regspeaker: 화자식별 목소리 등록 여부, Number (0 은 미등록, 1 은 등록) (이 값은 containerid 전환에 영향을 받지 않으며, 기가지니 단말에 설정된 사용자의 regspeaker 값이 넘겨진다.)
 - extra.regauthspeaker: 화자인증 목소리 등록 여부, Number (0 은 미등록, 1 은 등록) (이 값은 containerid 전환에 영향을 받지 않으며, 기가지니 단말에 설정된 사용자의 regspeaker 값이 넘겨진다.)
 - extra.registwithapp : boolean 값으로 기가지니 앱과 페어링 되었는지 확인할 수 있는 값이다. true 이면 기가지니 앱과 페어링(유저 등록) 되어 있는 경우이고, false 이면 기가지니 앱과 페어링 되어 있지 않은 경우이다.
 - extra.telavailable : boolean 값으로 true 이면 기가지니로 전화 통화가 가능하며 false 이면 기가지니에서 전화 통화가 불가능하다.
- 사용 예

```
//callback 방식
gigagenie.appinfo. getUserInfo (null,function(result_cd,result_msg,extra){
    if(result_cd===200){
        var usernickname=extra.usernickname;
        var kwsid=extra.kwsid;
        var ispin=extra.ispin;
        var regspeaker=extra.regspeaker
        console.log("UserNickName:"+usernickname+" kwsid:"+kwsid+"
        regspeaker:"+regsepaker);
        if(ispin) console.log("ispin is true");
        else console.og("ispin is false");
    } else {
        console.log("getUserInfo fail.");
    }
});
```

4.5.9 User 설정 정보 조회 API (gigagenie.appinfo.getUserSetInfo)

User 가 설정한 정보를 조회한다. 이 정보는 Companion App 에서 기가지니 사용자의 주문 등을 위해서 입력한 정보(이름, 전화번호, 상세 주소값)이다. 이 값은 containerid 전환에 영향을 받지 않는다.

- function getUserSetInfo(options,callback)
- options 은 null 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- result_cd 가 200 이면 extra 값에 다음이 설정되어 전달된다. 사용자가 설정한 정보가 없으면 null 이 전달된다.
 - extra.name : 사용자 이름
 - extra.telno : 사용자 전화번호
 - extra.address : 사용자 주소(동단위)
 - extra.subaddress : 상세 주소
 - extra.gpsX: WGS84 경도 좌표
 - extra.gpsY: WGS84 위도 좌표

- extra.posX : GRS80(UTM-K) 경도 좌표
- extra.posY : GRS80(UTM-K) 위도 좌표

○ 사용 예

```
gigagenie.appinfo.getUserSetInfo (null,function(result_cd,result_msg,extra){
  if(result_cd===200){
    var username=extra.name;
    var telno=extra.telno;
    var address=extra.address;
    console.log("Name:"+username+" telno:"+telno+" address:"+address);
  } else {
    console.log("getUserInfo fail.");
  }
});
```

4.5.10 주소록 조회 API (gigagenie.appinfo.queryAddressBook)

간편송금 주소록을 조회한다. 이름으로 조회하거나, 번호로 조회한다.

- function queryAddressBook(options,callback)
- options 은 다음과 같다
 - options.queryby : Mandatory, String 이며 다음의 값을 가진다.
 - ◆ "byname" : 이름으로 조회
 - ◆ "list" : 등록된 모든 번호 조회
 - options.name: Optional, String 이며 별칭 또는 실명이다. 별칭 또는 실명 중 매치되는 것을 전달한다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 404: 조회된 연락처 없음
 - 500: 시스템 Error
- result_cd 가 200 이면 extra 값에 다음이 전달된다.
 - extra.addresscnt: 조회된 address 갯수
 - extra.address : 조회된 address 리스트로 다음으로 구성된 JSON 객체 리스트이다.
 - ◆ nickname: String 으로 별칭이다.
 - ◆ name: String 으로 실명이다.

◆ telno : 모바일 번호이다.

○ 사용 예

```
//callback 방식
var options={};
options.queryby="bytelno"
gigagenie.appinfo.queryAddressBook(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("Address is:"+JSON.stringify(extra.address));
    } else {
        console.log("getUserInfo fail.");
    }
});
```

4.5.11 우리집 상세 주소 조회 API (gigagenie.appinfo.getAddressDetail)

GiGA Genie 에 설정된 우리집 주소를 조회한다. 우리집 주소는 Companion App 에서 설정한 주소 정보이다. userinfo 권한이 있는 appid 만 호출 가능하다.

- function getAddress(options,callback)
- options 는 null 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 404: 저장된 주소값 없음
 - 500: 시스템 Error
- result_cd 가 200 이면 extra 값에 다음이 설정되어 전달된다.
 - extra.address : 우리집 주소로 설정된 주소 String
 - extra.subaddress: 상세 주소 String (없으면 전달 안됨)
 - extra.gpsX: WGS84 경도 좌표
 - extra.gpsY: WGS84 위도 좌표
 - extra.posX : GRS80(UTM-K) 경도 좌표
 - extra.posY : GRS80(UTM-K) 위도 좌표

○ 사용 예

```
//callback 방식
gigagenie.appinfo.getAddress(null,function(result_cd,result_msg,extra){
    if(result_cd===200){
```

```

var address=extra.address;
var subaddress=extra.subaddress;
console.log("GiGA Genie Home Address="+address+" "+subaddress);
console.log("WGS84 X:"+extra.gpsX+" WGS84 Y:"+extra.gpsY);
console.log("GRS80 X:"+extra.posX+" GRS80 Y:"+extra.posY);
else if(result_cd===404){
    var address=extra.address;
    console.log("Address is not found");
} else {
    console.log("Getting Home Address fail.");
}
});

```

4.5.12 계정변경 Event 수신 (gigagenie.appinfo.onChangeUser)

계정변경 이벤트를 수신한다. 3rd Party App 이 실행중일 때, 발화로 계정변경을 하였을 경우 이벤트가 전달된다. 4.1.2 changeContainerId 실행시에 본 이벤트는 발생하지 않는다. 다음의 콜백 함수를 구현한다.

- gigagenie.appinfo.onChangeUser 콜백 구현
- 콜백은 function(extra) 로 구현해야 하며 extra 에는 다음의 Parameter 가 전달된다.
 - extra.changedContainerId : String, Mandatory 로 변경된 계정의 ContainerId
 - extra.eventType : String, Mandatory 로 계정변경 유형이다. 계정 변경 또는 계정 변경은 1, 계정 등록 해제는 0 으로 전달된다. (페어링 해제 경우 등록된 사용자가 없다. ContainerId 는 Mac-Id 기반으로 생성됨)

```

//callback 방식
gigagenie.appinfo.onChangeUser=function(extra){
    console.log("Changed User:"+extra.changedContainerId);
};

```

4.6 Media API (gigagenie.media)

4.6.1 Mute 이벤트 API (gigagenie.media.onMuteRequest)

G-Box 의 복합 상황에서 ContainerApp 에 Mute 요청 이벤트 수신하는 콜백이다. G-Box 복합 상황은 ContainerApp 에서 준수해야 되는 사항으로 HTML 에서 동영상, 오디오 플레이시에 onMuteRequest 이벤트를 수신하여 동영상, 오디오의 음성을 Off 해야 한다. (강제 사항으로, App 차단 가능함)

- gigagenie.media.onMuteRequest
- 콜백은 function callback(extra)로 구현해야 하며 extra 에는 mute 값이 전달된다.
 - mute: 오디오 On/Off 에 대한 Boolean 값으로, true 인 경우 오디오를 Off 시켜야 하며, false 인 경우 mute 가 해제된 경우 수신된다.
- 사용 예

```
//callback 방식
var options={};
gigagenie.media.onMuteRequest=function(extra){
  if(extra.mute){
    console.log("mute requested ");
    //동영상, 오디오 Mute
  } else {
    console.log("mute is released.");
    //동영상, 오디오 Mute 해제
  }
};
```

4.6.2 소프트 키보드 표시 (gigagenie.media.showSoftKeyboard)

화면에 소프트키보드를 표시한다.

- function showSoftKeyboard(options,callback)
- options 은 다음과 같다.
 - options.showFlag : Boolean, Softkeyboard 표시 여부
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 값은 200 일 경우 다음이 전달된다.
 - extra.authResult: String 으로 인증 결과가 전달된다.


```
//callback 방식
var options={};
options.showFlag=true;
gigagenie.media.showSoftKeyboard(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log('showSoftKeyboard Success..');
    } else {
        console.log("showSoftKeyboard Fail");
    }
});
```

4.6.3 리모컨 키 이벤트 수신 API (gigagenie.media.onRemoteKeyEvent)

JS Keycode 가 없는 리모컨 키 이벤트를 수신한다. (JS Keycode 는 5.2 참고)

- gigagenie.media.onRemoteKeyEvent
- 콜백은 function callback(extra)로 구현해야 하며 extra 에는 key 값이 전달된다.

- extra.key : 키 코드 값

◀ : "prev"

▶ : "next"

* : "asterisk"

: "sharp"

- 사용 예

```
//callback 방식
gigagenie.media.onRemoteKeyEvent=function(extra){
    console.log("Getting onRemoteKeyEvent."+extra..key);
};
```

4.6.4 음성 녹음 API (gigagenie.media.startRecordAudio)

음성 녹음 및 녹음 데이터를 전달한다. 음성 녹음 API 호출시 즉시 결과값이 리턴되며 녹음이 완료되면 onVoiceRecordComplete Event(4.6.6 참고) 가 발생하면서 데이터가 전달(Javascript 의 AudioBuffer 또는 HTTP Post)된다.

- function startRecordAudio(options,callback)
- options 는 다음과 같다.

- options.delivery : String, Mandatory 로 녹음파일 전달 방법이다. 다음의 두가지 방법이 지원된다.
 - ◆ options.delivery='memory' : 녹음 데이터를 JavaScript 의 audioBuffer 로 전달한다.
 - ◆ options.delivery='webhook' : 녹음 데이터를 HTTP Post 로 전달한다. 이 경우 다음의 options 을 설정해야 한다.
 - options.url : 녹음 데이터를 전달 받기 위한 http post url
- options.recordTime: Number, Mandatory 로 녹음할 시간(초단위)이다. 음성 녹음은 최대 5 분까지 가능하다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 400: 요청 파라미터가 없거나, 5 분 이상의 녹음 요청에 대한 오류
 - 405: 잘못된 delivery 옵션
 - 500: 시스템 Error
- 사용 예
 - 사용 예는 4.6.5 음성 녹음 수신 API 와 같이 설명한다.

4.6.5 음성 녹음 수신 API (gigagenie.media.onVoiceRecordComplete)

음성 녹음 API 에 의한 녹음 결과를 받는 Event Callback 이다. 음성 녹음 API 에서 delivery 를 memory 로 하였을 경우 콜백에 AudioBuffer Javascript Object 가 전달된다. delivery 를 webhook 으로 하였을 경우 http post request 를 실행 전 녹음이 완료 되었음을 알려준다. Webhook 의 경우 http post request 도중에 발생된 에러는 리포트 하지 않는다.

- gigagenie.media.onVoiceRecordComplete 콜백 구현
- 콜백은 function callback(result_cd, buffer) 로 구현해야 한다.
 - result_cd 는 다음의 값을 가진다.
 - ◆ 200: 레코딩 성공
 - ◆ 500: 시스템 에러
 - ◆ 503: 취소됨
 - buffer 에는 녹음된 PCM 파일(wav 포맷)이 AudioBuffer JavaScript Object 로 전달된다. (webhook 방식의 경우 null)

○ 사용 예

```
//1. delivery를 memory로 했을 경우
var options={};
options.delivery='memory';
options.recordTime=10;
var audioBuffer=null;
gigagenie.media.onVoiceRecordComplete=function(result_cd,inBuffer){
    if(result_cd===200) {
        console.log('recording success');
        audioBuffer=inBuffer;
    } else console.log('recording fail:'+result_cd)
}
gigagenie.media.startRecordAudio(options,function(result_cd,result_msg,extra){
    console.log(result_cd);
});

//AudioBuffer로 웹에서 출력하고자 하는 경우
var source=context.createBufferSource();
source.buffer=audioBuffer;
source.connect(context.destination);
source.start(0);
//-----

//2. delivery를 webhook 으로 했을 경우
var options={};
options.delivery='webhook';
options.recordTime=10;
options.url='https://yourhostname/path'
gigagenie.media.onVoiceRecordComplete=function(result_cd,inBuffer){
    if(result_cd===200) {
        console.log('recording success');
        //웹서버에서 확인하는 코드?
    } else console.log('recording fail:'+result_cd)
}
gigagenie.media.startRecordAudio(options,function(result_cd,result_msg,extra){
    console.log(result_cd);
});
//post 한 파일을 플레이하고자 하는 경우
```

```

var audioBuffer=null;
function loadSoundFromUrl(url) {
    var request = new XMLHttpRequest();
    request.open('GET', url, true);
    request.responseType = 'arraybuffer';
    request.onload = function() {
        context.decodeAudioData(request.response, function(buffer) {
            audioBuffer = buffer;
        }, function(){
            console.log('error');
        });
    }
    request.send();
}
loadSoundFromUrl('https://yourhosstname/path');
var source=context.createBufferSource()
source.buffer=audioBuffer;
source.connect(context.destination)
source.start(0)

```

4.6.6 음성 녹음 중지 API (gigagenie.media.stopRecordAudio)

음성 녹음을 중지한다. 음성 녹음 중지 API 는 즉시 응답하며, onVoiceRecordComplete 에 503 으로 오류 이벤트가 전달된다.

- function stopRecordAudio(options,callback)
- options 는 null 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 404: 음성녹음 요청 중이 아님
 - 500: 시스템 Error
- 사용 예

```

gigagenie.media.onVoiceRecordComplete=function(result_cd,inBuffer){
    if(result_cd===200) {
        console.log('recording success');
    } else if(result_cd===503){
        console.log('recording interrupted');
    }
}

```

```

    } else console.log('recording fail:'+result_cd)
  };
  gigagenie.media.stopRecordAudio(null,function(result_cd,result_msg,extra){
    console.log(result_cd);
  });

```

4.6.7 Channel 생성 (gigagenie.media.createChannel)

기가지니와 P2P 로 데이터를 전송하기 위한 Channel 을 생성한다. 이 API 는
기가지니에 가상 키 입력을 전달하거나, WebRTC 로 Audio/Video 스트림을 연결하기
위한 Signal Channel 에 활용된다.

- function createChannel(options,callback)
- options 은 다음과 같다.
 - options.channeltype : String Mandatory 로 생성하고자 하는 Channel
유형이다. 다음의 유형을 지원한다.
 - ◆ channeltype='webrtc.audio' : WebRTC Audio Stream Signal Channel
생성(참고 사항, 5.4, 6. 별첨 참고)
 - options.usecontainerid : boolean 값으로 channelId 로 ContainerID 를
이용한다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 값은 200일 경우 다음이 전달된다.
 - extra.sessionid: String 으로 Channel 서버에서 채널 참여를 위한 id 이다.
(options.useContainerId 가 true 일 경우 containerid 를 반환)
- 사용 예

```

//callback 방식
var options={};
options.channeltype='audio';
options.usecontainerid=true;
gigagenie.media.createRelayChannel(options,function(result_cd,result_msg,extra){
  if(result_cd===200){
    console.log('createRelayChannel Success..');
  }
});

```

```

    } else {
        console.log("createRelayChannel Fail");
    }
});

```

4.6.8 Channel 제거 (gigagenie.media.destroyChannel)

4.6.7 에서 생성한 Channel 을 제거한다.

- function destroyChannel(options,callback)
- options 은 다음과 같다.
 - options.channeltype: 4.6.7 에서 생성한 channel type 이다.
 - options.sessionid : 4.6.7에서 생성한 sessionid 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 값은 null 이다.
- 사용 예

```

//callback 방식
var options={};
options.sessionid='12345';
gigagenie.media.destroyChannel(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log('createRelayChannel Success..');
    } else {
        console.log("createRelayChannel Fail");
    }
});

```

4.6.9 WebRTC Audio 연결 (gigagenie.media.startWebRTCAudio)

4.6.7 에서 생성한 channel 예(channeltype='webrtc.audio') WebRTC 프로토콜 기반 Audio Stream 을 연결한다. 현재는 단방향으로만 지원한다.

- function startWebRTCAudio(options,callback)
- options 은 다음과 같다.
 - options.sessionid : String, Mandatory 로 4.6.7 에서 생성한 webrtcudio Channel 의 sessionid 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 값은 null 이다.
- 사용 예

```
//callback 방식
var options={};
options.sessionid='123456;
gigagenie.media.startWebRTCAudio(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log('startWebRTCAudio Success..');
    } else {
        console.log("startWebRTCAudio Fail");
    }
});
```

4.6.10 WebRTC Audio 연결 해제 (gigagenie.media.stopWebRTCAudio)

4.6.9 에서 생성한 webrtcudio 연결을 해제한다.

- function stopWebRTCAudio(options,callback)
- options 은 다음과 같다.
 - options.sessionid : String, Mandatory 로 4.6.7에서 생성한 KeyStream Channel 의 sessionid 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 값은 null 이다.
- 사용 예

```
//callback 방식
var options={};
options.sessionid='123456;
gigagenie.media.stopWebRTCAudio(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log('stopWebRTCAudio Success..');
    } else {
        console.log("stopWebRTCAudio Fail");
    }
});
```

4.6.11 Channel 연결 유지 (gigagenie.media.keepaliveChannel)

4.6.7 에서 생성한 Channel 연결을 유지한다. 설정하지 않을 경우 70 초 후 연결이 해제된다.

- function keepaliveChannel(options,callback)
- options 은 null 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 값은 null 이다.
- 사용 예

```
//callback 방식
var ticker = 0;
setInterval(function(){
    ticker++;
    if(ticker % 30 ===7){
        console.log('keepalive:'+ticker);
        gigagenie.media.keepaliveChannel(null,function(result_cd,result_msg,extra){
            console.log("keepaliveChannel");
        });
    }
},1000);
```


4.6.12 Channel 이벤트 수신 (gigagenie.media.onChannelEvent)

4.6.7 에서 생성한 Channel 의 이벤트를 수신한다.

- gigagenie.media.onChannelEvent 콜백 구현
- 콜백은 function callback(extra) 로 구현해야 한다.
 - extra 는 웹 소켓 응답이 전달된다. (6. 별첨 참고)
- 사용 예

```
//callback 방식
gigagenie.media.onChannelEvent=function(result_cd,result_msg,extra){
    console.log('onChannelEvent extra:'+JSON.stringify(extra));
    alert('onChannelEvent extra:'+JSON.stringify(extra));
};
```

4.7 Payment API (gigagenie.payment)

4.7.1 OTV 결제 사용자 확인(gigagenie.payment.otvCheckUser)

OTV 결제를 위한 사용자 인증 및 상태체크를 요청한다. 결제 이용 가능 사용자가 아닌 경우 가입 단계를 진행한다.

- function otvCheckUser(options,callback)
- options 은 다음과 같다.
 - options.homeShoppingCode : String, Mandatory 제휴사 코드
 - options.payOption: Number, Mandatory 인증방법(1: 리모콘, 2: 음성)
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 410 : 요청 중 신규 요청으로 중지됨
 - 500: 시스템 Error
- extra 값은 200 일 경우 다음이 전달된다.
 - extra.authResult: String 으로 인증 결과가 전달된다.

```
//callback 방식
var options={};
options.homeShoppingCode='1111111111';
options.payOption=2;
```

```
gigagenie.payment.otvCheckUser(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        if(extra.authResult==='Success'){
            console.log("User Can Pay..");
        }
    } else {
        console.log("setAudioRecog Fail");
    }
});
```

4.7.2 OTV 결제 토큰 요청 (gigagenie.payment.otvReqPayToken)

OTV 결제 토큰을 요청한다. 결제가 불가능한 경우(목소리 미등록, OPAY 회원 미가입 등등) 에러코드만 리턴한다.

- function otvReqPayToken(options,callback)
- options 은 다음과 같다.
 - options.homeShoppingCode : String, Mandatory 제휴사 코드
 - options.payOption: Number, Mandatory 인증방법(1: 리모콘, 2: 음성)
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 값은 200 일 경우 다음이 전달된다.
 - extra.reqResult: String 으로 인증 결과가 전달된다.
 - extra.purchaseAuthToken : OTV 결제 토큰

```
//callback 방식
var options={};
options.homeShoppingCode='1111111111';
options.payOption=2;
gigagenie.payment.otvReqPayToken(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        if(extra.reqResult==='Success'){
            console.log('PayToken:'+extra.purchaseAuthToken);
        }
    } else {
        console.log("reqPayToken Fail");
    }
});
```

```

    }
  });

```

4.7.3 OTV 결제 인증-토큰 요청 (gigagenie.payment.otvReqPayTokenWithAuth)

OTV 결제 토큰을 요청한다. 결제가 불가능한 경우(목소리 미등록, OPAY 회원 미가입 등등)의 경우 해당 인증-가입 단계를 진행하고 결제 토큰 발급이 된다.

- function otvReqPayTokenWithAuth(options,callback)
- options 은 다음과 같다.
 - options.homeShoppingCode : String, Mandatory 제휴사 코드
 - options.payOption: Number, Mandatory 인증방법(1: 리모콘, 2: 음성)
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 값은 200 일 경우 다음이 전달된다.
 - extra.reqResult: String 으로 인증 결과가 전달된다.
 - extra.purchaseAuthToken : OTV 결제 토큰

```

//callback 방식
var options={};
options.homeShoppingCode='1111111111';
options.payOption=2;
gigagenie.payment.otvReqPayTokenWithAuth(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        if(extra.reqResult=='Success'){
            console.log('PayToken:'+extra.purchaseAuthToken);
        }
    } else {
        console.log("reqPayToken Fail");
    }
});

```

4.8 영상인식 API (gigagenie.vision)

4.8.1 영상인식 서비스 시작(gigagenie.vision.startFaceAnalysis)

Camera 가 부착된 GiGA Genie 에서 영상인식 서비스를 시작한다. 영상인식 서비스가 시작되면 4.8.3 의 onFaceAnalysisResult 에 영상인식정보 Event 가 올라온다.

- function startFaceAnalysis(options,callback)
- options null 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 값은 null 이다.
- 사용 예

```
//callback 방식
gigagenie.vision.startFaceAnalysis(null,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("startFaceAnalysis Success");
    } else {
        console.log("startFaceAnalysis Fail");
    }
});
```

4.8.2 영상인식 서비스 중지(gigagenie.vision.stopFaceAnalysis)

Camera 가 부착된 GiGA Genie 에서 영상인식 서비스를 시작한다.

- function stopFaceAnalysis(options,callback)
- options null 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 값은 null 이다.

```
//callback 방식
gigagenie.vision.stopFaceAnalysis(null,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("stopFaceAnalysis Success");
    } else {
        console.log("stopFaceAnalysis Fail");
    }
});
```

```
});
```

4.8.3 영상인식 정보 수신 API (gigagenie.vision.onFaceAnalysisResult)

영상인식 서비스를 시작하면 인식 결과를 받는 Event Callback 이다.

- gigagenie.vision.onFaceAnalysisResult 콜백 구현
- 콜백은 function callback(extra) 로 구현해야 하며 extra 에는 다음의 Parameter 가 전달된다.
 - userNameList : 인식된 사용자 이름의 JSON Array(String Array)
 - userIdList : 인식된 사용자 ContainerID JSON Array(String Array)
 - userAgeList : 인식된 사용자의 연령추정 값 JSON Array(String Array)
 - userGenderList : 인식된 사용자의 성별추정 값 JSON Array (String Array 임, 값은 M 이면 남, F 이면 여)
 - userExpressionList: 인식된 사용자의 표정 추정 값으로 JSON Array (String Array 임, 값은 Neutral, Smile, null 임)
 - userCount : 감지된 사용자 수, Number
 - envAccuracy: 정확도, Number
 - ◆ 1: QVGA 가로(320x240)
 - ◆ 2: VGA 세로(480x640)
 - ◆ 3: VGA 가로(640x480)
 - ◆ 4: HD (1280x720)
 - ◆ 5: FHD (1920x1080)
- 사용 예

```
gigagenie.vision.onFaceAnalysisResult=function(extra){
    for(var i=0;i<userCount;i++){
        console.log("name:"+userNameList[i]);
        console.log("id:"+userIdList[i]);
        console.log("age:"+userAgeList[i]);
        .....
    }
};
```

4.9 Admin API (gigagenie.admin)

4.9.1 앱인증정보 조회(gigagenie.admin.getAuthAppList)

AuthKey 가 설정된 앱 리스트를 조회한다. ktadmin 권한을 가진 appid 만 호출 가능하다.

- function getAuthAppList(options,callback)
- options 은 null 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- result_cd 가 200 일 경우 다음의 extra 값이 전달된다.
 - extra.appidlist : String Array 로 AuthKey 가 설정된 appid 리스트 JSON Object 이다.

```
//callback 방식
gigagenie.admin.getAuthAppList(null,function(result_cd,result_msg,extra){
    if(result_cd===200){
        var appidlist=extra.appidlist;
        for(var i=0;i<appidlist.length;i++) console.log(appidlist[i]);
    } else {
        console.log("getAppSetFlag Fail");
    }
});
```

4.9.2 앱 AuthKey 정보 삭제 (gigagenie.admin.delAuthKey)

AuthKey 가 설정된 앱 리스트를 조회한다. ktadmin 권한을 가진 appid 만 호출 가능하다.

- function delAuthKey(options,callback)
- options 은 다음과 같다.
 - options.appid : String, Mandatory 로 조회하고자 하는 appid 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 는 null 이다.

```
//callback 방식
```

```

var options={};
options.appid="X0000000";
gigagenie.admin.delAuthKey(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("delAuthKey Success");
    } else {
        console.log("delAuthKey Fail");
    }
});

```

4.9.3 앱설정 데이터 삭제 (gigagenie.admin.delAppData)

AuthKey가 설정된 앱 리스트를 조회한다. ktadmin 권한을 가진 appid 만 호출 가능하다.

- function delAppData(options,callback)
- options 은 다음과 같다.
 - options.appid : String, Mandatory 로 조회하고자 하는 appid 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- extra 는 null 이다.

```

//callback 방식
var options={};
options.appid="X0000000";
gigagenie.admin.delAppData(options,function(result_cd,result_msg,extra){
    if(result_cd===200){
        console.log("delAppData Success");
    } else {
        console.log("delAppData Fail");
    }
});

```

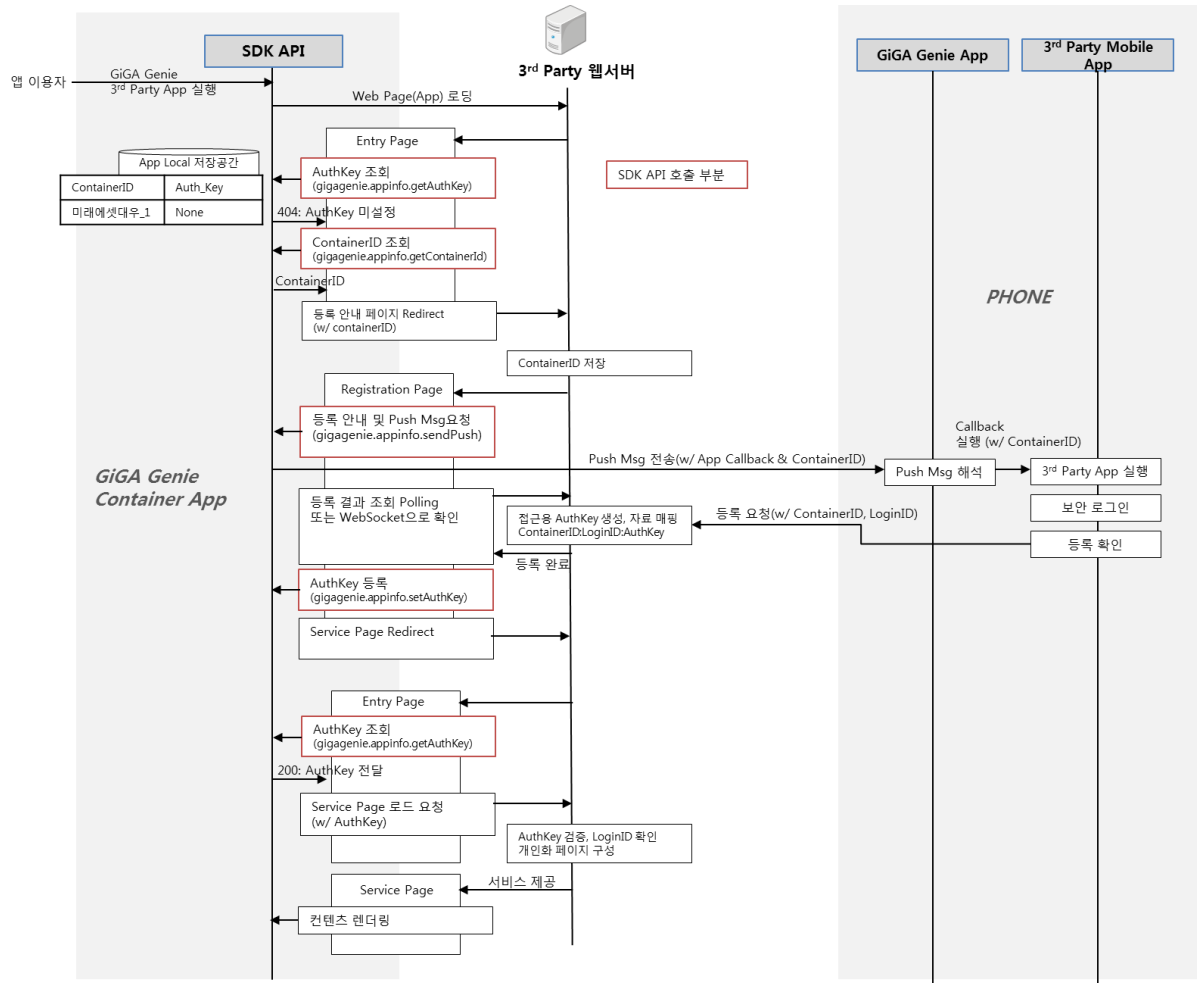
4.9.4 앱설정정보조회 (gigagenie.admin.getAppSetFlag)

특정 앱의 상세 정보(인증, 데이터유무)를 조회한다. ktadmin 권한을 가진 appid 만 호출 가능하다.

- function getDetailInfo(options,callback)
- options 은 다음과 같다.
 - options.appid : String, Mandatory 로 조회하고자 하는 appId 이다.
- result_cd 는 다음과 같이 정의된다.
 - 200: 성공
 - 500: 시스템 Error
- result_cd 가 200 일 경우 다음의 extra 값이 전달된다.
 - extra.authkeyflag : boolean 으로 AuthKey 가 설정되어 있으면 true, 아니면 false 이다.
 - extra.dataflag : boolean 으로 해당 appId 로 데이터가 저장되어 있으면 true, 그렇지 않으면 false 이다.

```
//callback 방식
var options={};
options.appid='X00000000';
gigagenie.admin.getAppSetFlag(null,function(result_cd,result_msg,extra){
    if(result_cd===200){
        if(extra.authkeyflag) console.log("AuthKey exists");
        if(extra.authkeyflag) console.log("App Data exist");
    } else {
        console.log("getAppSetFlag Fail");
    }
});
```


○ Push API 를 이용한 3rd Party App 로그인 정보 처리 Flow



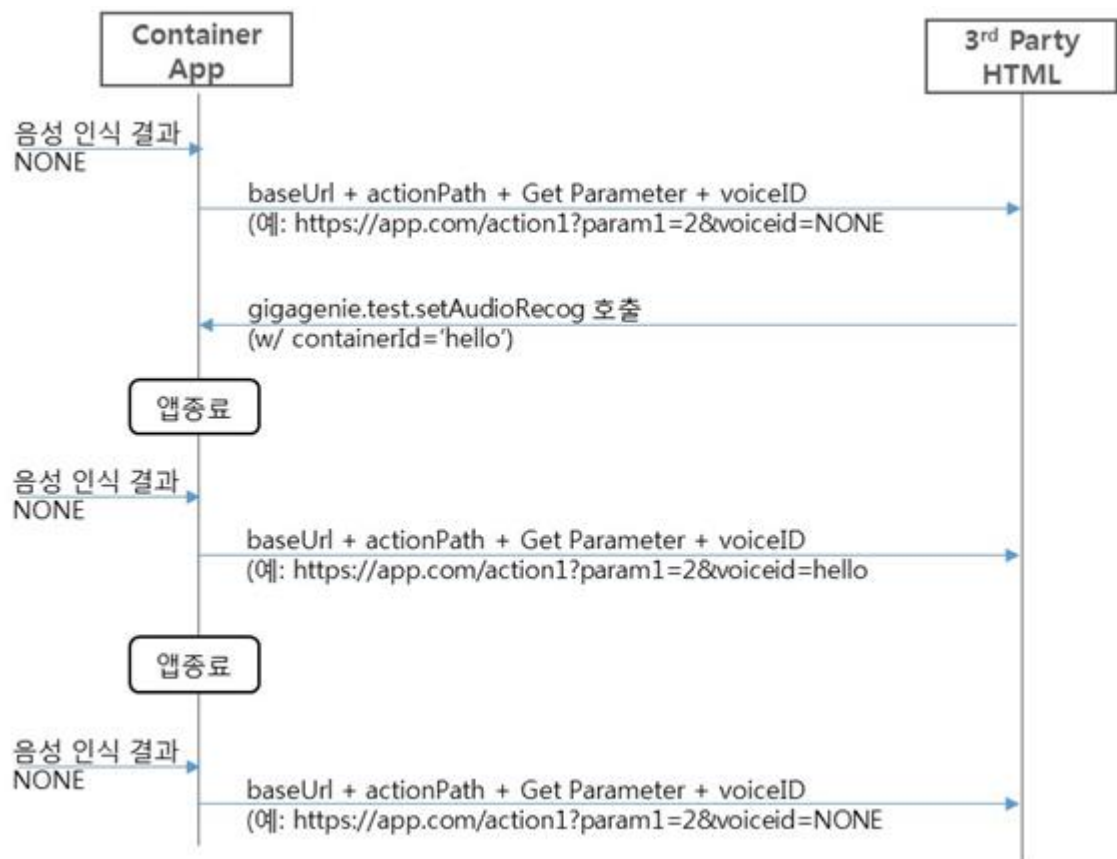
5.2 리모컨 Key Code 가이드

○ 리모컨 Key Code 와 JS Key Code 값

Key code		
리모컨 Key	기능	JS KeyCode
상	위로 포커스 이동	38
하	아래로 포커스 이동	40
좌	왼쪽으로 포커스 이동	37
우	오른쪽으로 포커스 이동	39
확인	선택/확인	13
1	숫자 키	49
2	숫자 키	50
3	숫자 키	51
4	숫자 키	52
5	숫자 키	53

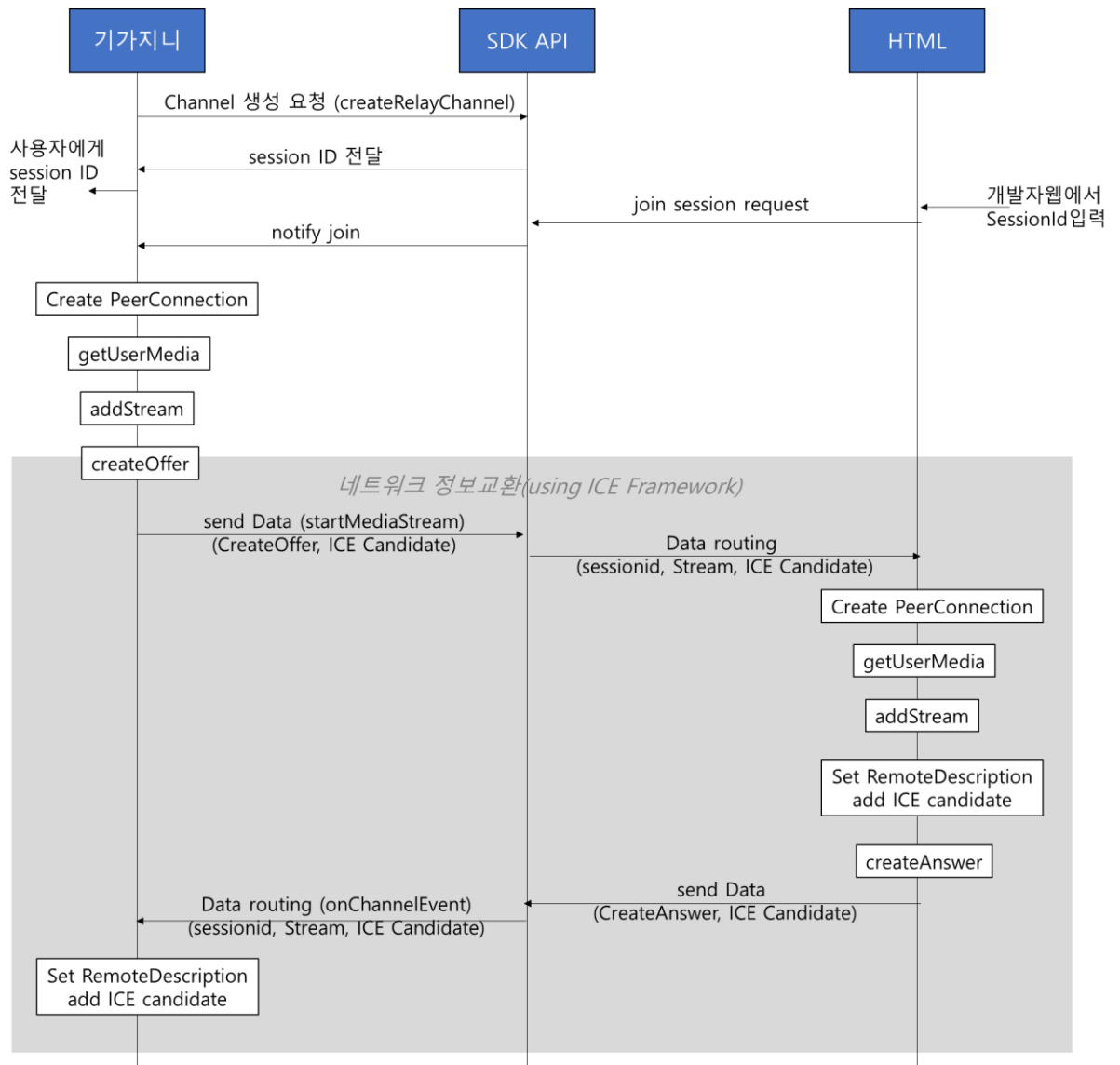
6	숫자 키	54
7	숫자 키	55
8	숫자 키	56
9	숫자 키	57
0	숫자 키	48
이전	앱 종료/뒤로가기	
나가기	앱 종료	
재생/일시정지	선택/확인	179
채널 위로		33
채널 아래로		34

5.3 gigagenie.test.setAudioRecog 이용 플로우



- setAudioRecog 이후 앱이 다시 실행되면 음성인식 결과가 설정한 containerId 로 전달되며, 이때 changeContainerId 로 ContainerApp 을 설정된 containerId 로 모드 변경이 가능하다.

5.4 WebRTC 이용 플로우 (AudioStream)

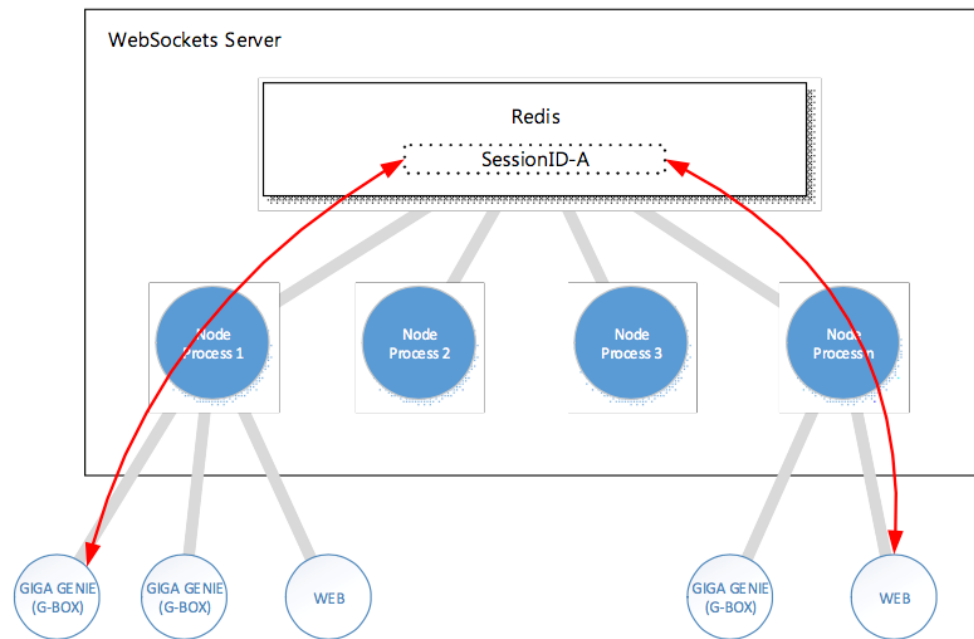


- 기가지니 단말에서 Offer 를 보내고 3rd Party HTML 에서 Answer 를 보내야 하며 반대로는 지원하지 않는다.

6. 별첨

6.1 웹 소켓 이용 방법

각 클라이언트는 분산된 Node process 에서 접속하더라도 공통된 세션을 공유하여 데이터를 송/수신 할 수 있습니다.



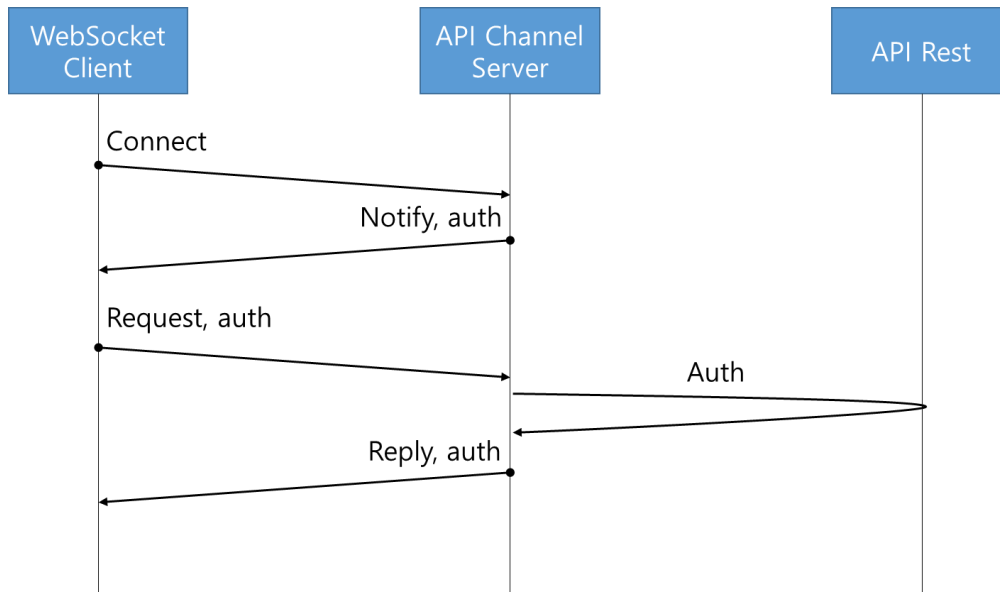
- 웹 소켓 서버 정보 (SSL 만 이용 가능)
 - wss://svccapp.gigagenie.ai/channel
- WebSocket API Payload (예약어)
 - msgtype : 메시지 송/수신 유형
 - ① request: 서비스 클라이언트에서의 요청, trxid 를 설정해서 트랜잭션을 구분해야 함
 - ② reply: request 에 대한 서버에서의 처리 결과 전달, trxid 로 transaction 구별
 - ③ notify: 서버에서 클라이언트로 전달되는 event 알림, trxid 없음
 - ④ data: Data 전달, trxid 없음, 데이터 전달 채널 정보(TBD)
 - operation : 인증, 세션 생성/소멸, 특정 데이터 전달 등의 작업 구분자
 - ① auth: 인증, 서버에 클라이언트가 인증 요청 시 사용
 - ② iam: 인증, 서버에서 WebSocket 접속한 클라이언트에 notify 할 때 사용
 - ③ create_session: 채널 세션 생성

- ④ join_session: 생성된 채널 세션에 참여
- ⑤ ras_alive: WebSocket 연결 유지를 위한 PING-PONG 시 사용
- ⑥ destroy_session: 생성된 채널 세션을 종료
- ⑦ set_input_history: 테스트를 위한 입력된 URL 저장(UUID 기준, 개발자 지원)
- ⑧ get_input_history: 테스트를 위한 입력된 URL 조회(UUID 기준, 개발자 지원)
- ⑨ set_appid_devmode: 해당 APP_ID 의 개발자모드 설정
- ⑩ check_appid_devmod: 해당 APP_ID 가 개발자모드로 설정되었는지 확인
- ⑪ confirm_appid_devmode: 해당 APP_ID 를 개발자모드로 설정(개발자 센터용, REST API 로 대체 가능)
- channeltype: 웹 소켓 채널의 용도 구분자
 - ① rtcaudio: WebRTC 오디오 채널 유형
 - ② system: 서버에서 발생하는 notify 를 전달하는 채널 유형 (예, ras_alive)

6.2 연결 (Connection)

- WebSocket Protocol 은 서비스에 따라 임의로 지정하여 사용할 수 있습니다.
 - WebSocket Protocol 표준 준수(Header: Sec-WebSocket-Protocol)
 - 연결예시) ["wss://svcapp.gigagenie.ai/channel", "ws-channel-v1"]

6.3 인증 (AUTH)



- ① 최초 Client 는 서버에 WebSocket connect 를 시도합니다.
- ② 서버는 접속한 클라이언트에 인증이 필요함을 알립니다.

■ Message sample :

```

{
  "msgtype": "notify",
  "operation": "iam",
  "channeltype": "system"
}

```

■ Client 에서의 수신 처리는 아래의 WebSocket 클라이언트 샘플 소스 참고.

- ③ Client 는 인증 Request 를 통해 인증을 요청합니다.

■ Message sample :

```

{
  "msgtype": "request",
  "operation": "auth",
  "uuid": "aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa",
  "channeltype": "webrtcadio"
}

```

■ Client 에서 Request 처리는 아래의 WebSocket 클라이언트 샘플 소스 참고.

- ④ 서버사이드에서는 내부 인증 연동을 통하여 그 결과를 reply 합니다.

■ Message sample :

```

{
  "msgtype": "reply",
}

```

```

"operation": "auth",
"channeltype": "system",
"result_cd": 200,
"result_msg": "authorized"
}

```

■ Client 에서 Reply 처리는 아래의 WebSocket 클라이언트 샘플 소스 참고.

- ⑤ 인증에 성공하면 접속은 유지되고, 통신을 할 수 있는 상태가 됩니다.
- ⑥ 만약, 일정시간 Client 가 인증에 응하지 않으면 접속은 자동 종료됩니다.

```

socket.onmessage = function(message) {
    console.log('Socket server message', message.data);
    document.getElementById('response').innerHTML = message.data;
    let pdata = JSON.parse(message.data);
    console.log(pdata);
    if (pdata.msgtype === 'notify') {
        if (pdata.operation === 'iam') {
            var auth_message = {
                "msgtype": "request",
                "operation": "auth",
                "uuid": uuid,
                "channeltype": "webrtcudio",
            };
            socket.send(JSON.stringify(auth_message));
            console.log('auth message sent.');
```



```

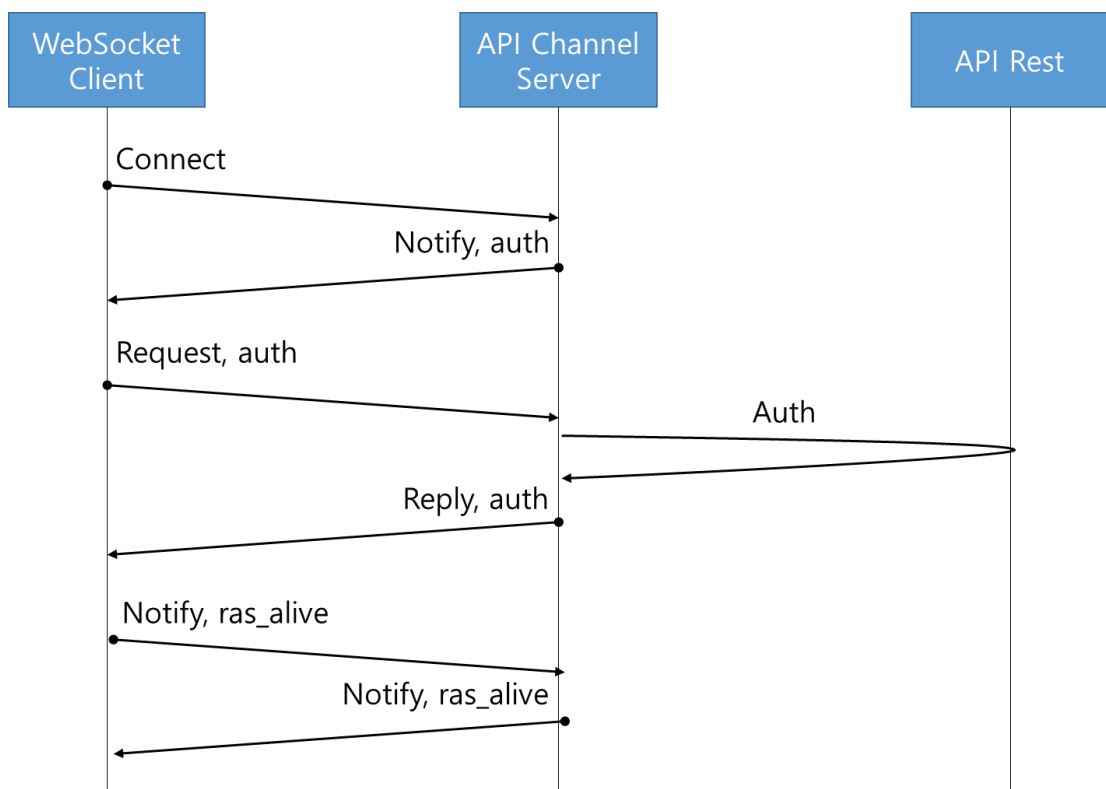
        "channeltype": " webrtcudio",
        "uuid": uuid,
        "trxid": "K0000000:1501463079077:00002"
    };
    socket.send(JSON.stringify(create_session));
    console.log('create_session message sent.');
```

```

    } else {
    }
}
} else if (pdata.msgtype === 'data') {
}
};

```

6.4 접속유지 (Alive test)



GATE-API Channel 서버는 접속한 클라이언트가 일정한 시간 이내에 Alive 메시지를 발송하지 않으면 접속을 강제로 종료합니다.

■ Message sample :

```
{  
  "msgtype": "notify",  
  "operation": "ras_alive",  
  "channeltype": "system"  
}
```

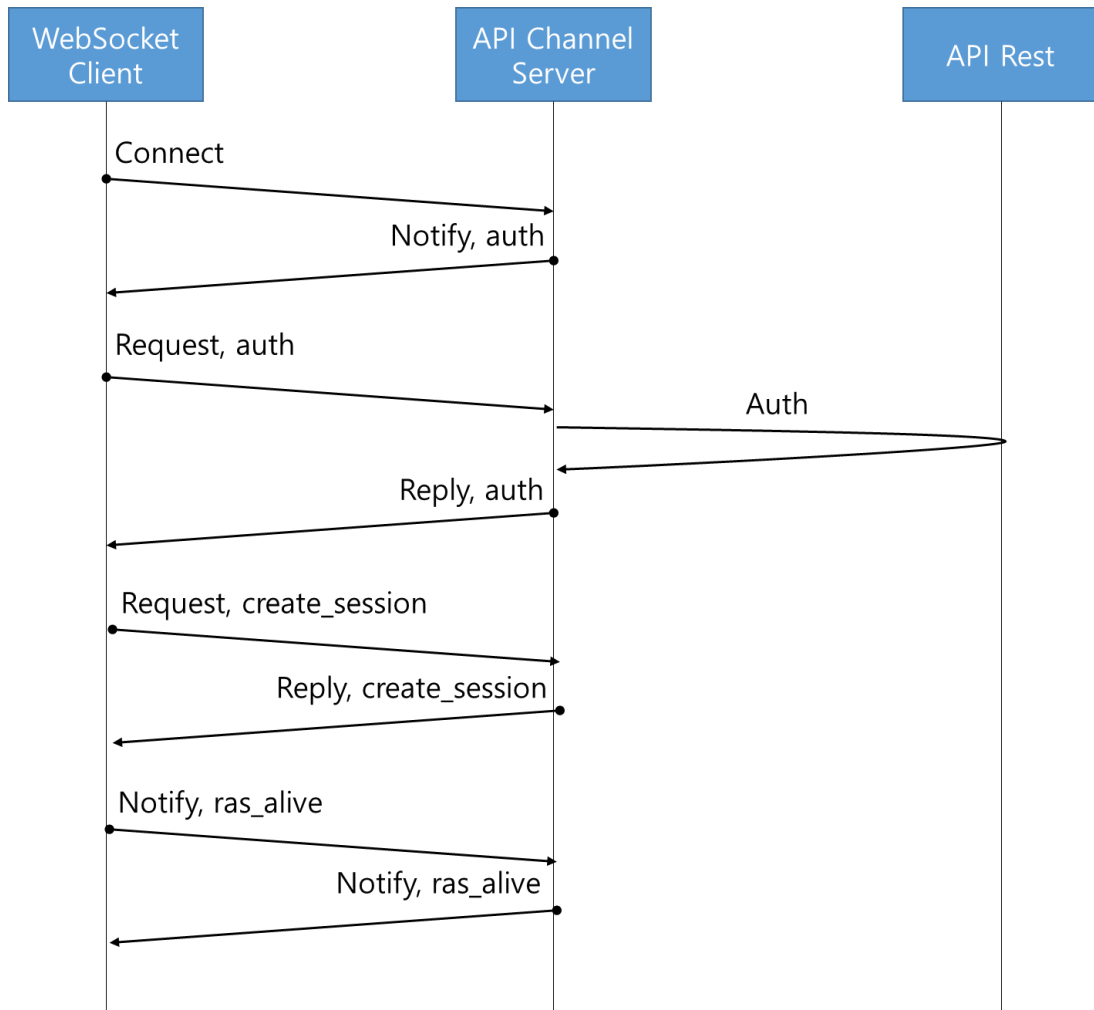
- Client 에서 alive 처리는 다음의 WebSocket 클라이언트 샘플 소스 참고.

```
let ticker = 0;  
setInterval(function() {  
  ticker++;  
  if (ticker % 30 === 7) {  
    console.log('@ras-alive:' + ticker);  
    var alive_session = {  
      "msgtype": "notify",  
      "operation": "ras_alive",  
      "channeltype": "system"  
    };  
    socket.send(JSON.stringify(alive_session));  
  }  
}, 1000);
```

본 과정은 Dead connection 을 처리하기 위한 하나의 절차로 현재 버전에서는 하기의 조건으로 처리합니다.

- 서버 접속 종료 시간: 70 seconds
- 클라이언트 권장 주기: 30 ~ 60 seconds

6.5 Session 생성 요청 (request, create_session)



Session 생성을 요청합니다. Session id 는 기본적으로 서버에서 생성하며, Session id 를 지정하면 지정된 이름으로 생성합니다.

■ Default session id 의 message sample:

```

{
  "msgtype": "request",
  "operation": "create_session",
  "channeltype": "webrtcudio",
  "uuid": "aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa",
  "trxid": "K0000000:1501463079077:00002"
}
  
```

■ Specific session id 의 message sample:

```

{
  "msgtype": "request",
  
```

```

"operation": "create_session",
"channeltype": "webrtc_audio",
"sessionid": "abcd",
"uuid": "aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa",
"trxid": "K0000000:1501463079077:00002"
}

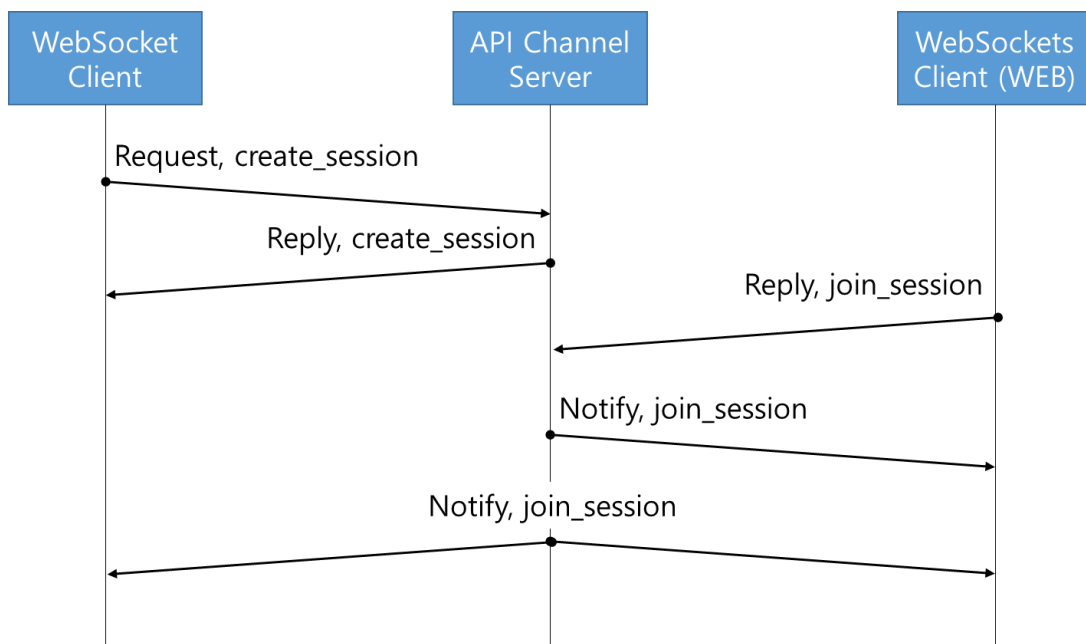
```

- Client 에서 상기 create_session 까지의 일련의 과정은 WebSocket 클라이언트 샘플 소스 참고.

클라이언트는 주기적으로 alive 메시지를 발송해야 접속을 유지할 수 있습니다.

6.6 Session 입장 요청 (request, join_session)

○



Session 입장을 요청합니다.

주의) 새로운 Client 는 초기 인증과정을 처리해야만 합니다.

- ① Request join 을 요청합니다.

- Message sample :

```

{
  "msgtype": "request",
  "operation": "join_session",

```

```

"channeltype": "webrtc.audio",
"sessionid": "aaaa",
"uuid": "aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa",
"trxid": "K0000000:1501463079077:00002"
}

```

② Request join 에 대한 Reply 처리가 됩니다.

● Message sample :

```

{
  "msgtype": "reply",
  "operation": "join_session",
  "result_cd": 200,
  "result_msg": "Success",
  "trxid": "K0000000:1501463079077:00002"
}

```

③ 입장이 되면 해당 Channel 에 Notify join_session 이 발생합니다.

● Message sample :

```

{
  "msgtype": "notify",
  "operation": "join_session",
  "sessionid": "aaaa",
  "channeltype": "webrtc.audio"
}

```

6.7 Session 종료 요청 (request, destroy_session)

Session 종료를 요청합니다.

① Request destroy 을 요청합니다.

● Message sample :

```

{
  "msgtype": "request",
  "operation": "destroy_session",
  "channeltype": "webrtc.audio",
  "sessionid": "aaaa",
}

```

```
{
  "uuid": "aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa",
  "trxid": "K0000000:1501463079077:00003"
}
```

② Request join 에 대한 Reply 처리가 됩니다.

- Message sample :

```
{
  "msgtype": "reply",
  "operation": "destroy_session",
  "result_cd": 200,
  "result_msg": "Success",
  "trxid": "K0000000:1501463079077:00003"
}
```

③ 입장이되면 해당 Channel 에 Notify destroy_session 이 발생합니다.

- Message sample :

```
{
  "msgtype": "notify",
  "operation": "destroy_session",
  "sessionid": "aaaa",
  "channeltype": "rtcaudio"
}
```

6.8 Data 전송 (Data)

