

PEOPLE.  
TECHNOLOGY.

# GiGA Genie + IoT 단말 제어 서비스 제작 가이드

기가지니 개발자 포털을 통한 IoT 서비스 제작 방법

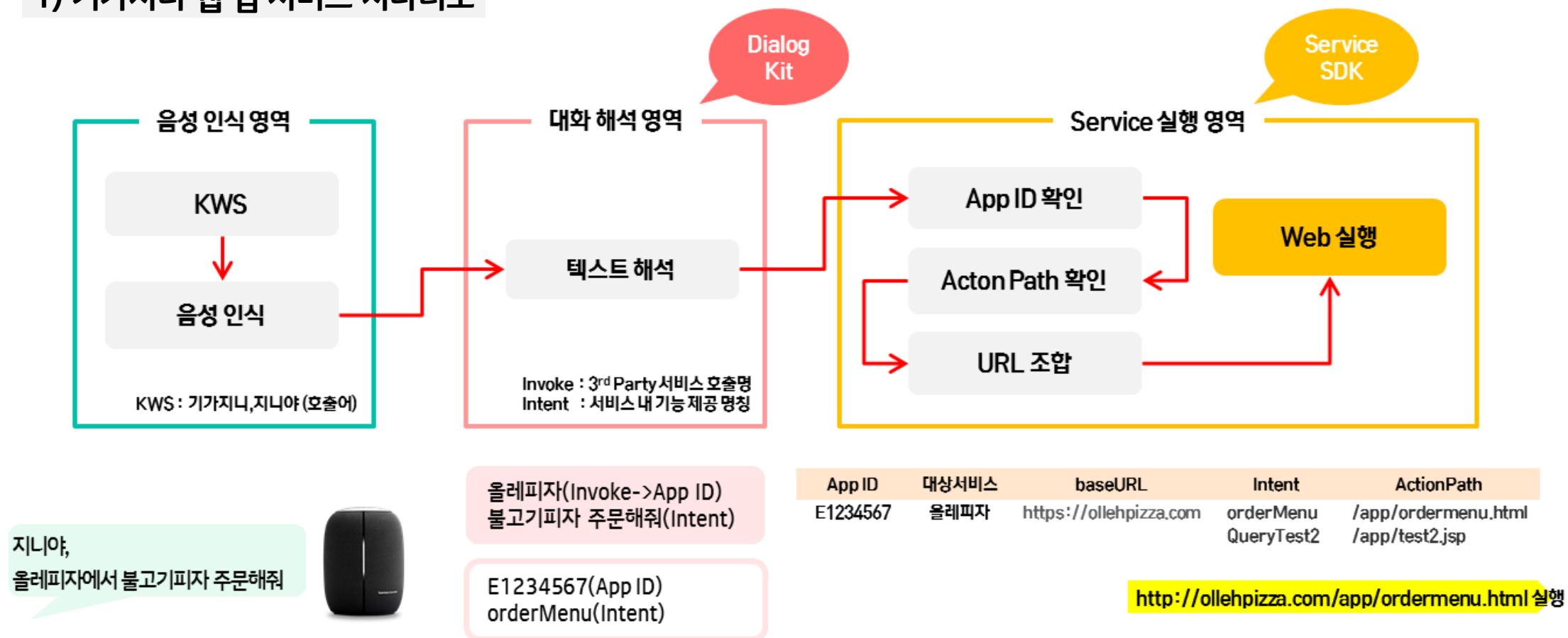
2018. 09. 07 | AI Eco 기술팀 안희정 전임

kt



# 1. 개요

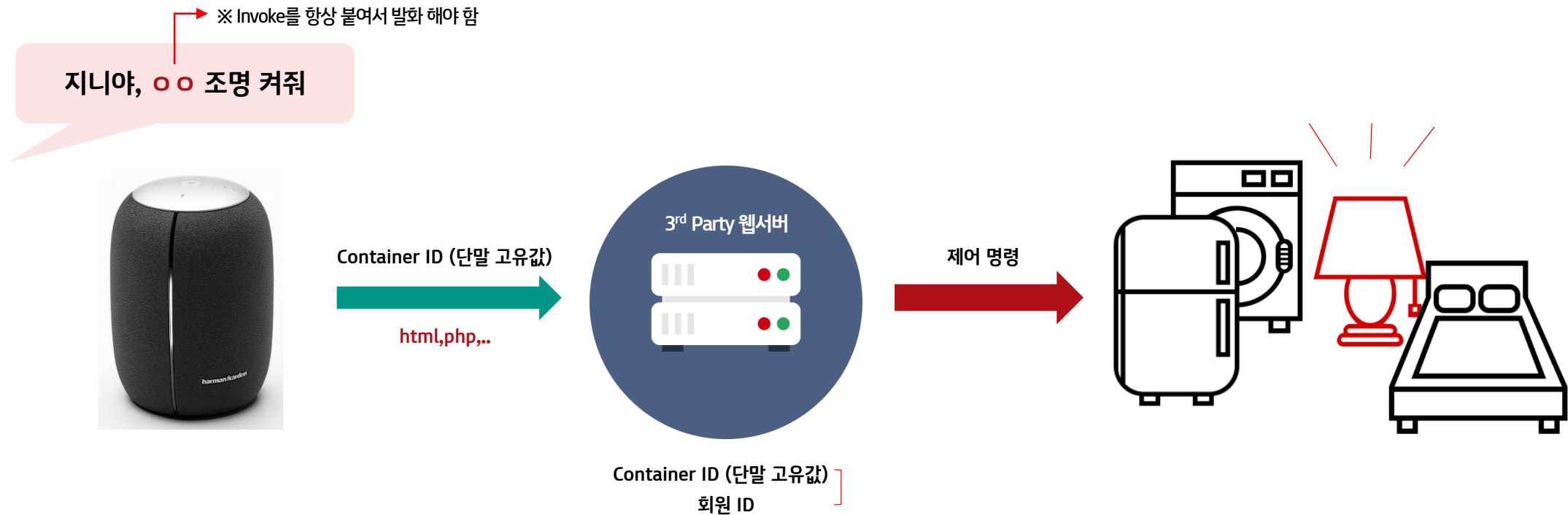
## 1) 기가지니 웹 앱 서비스 시나리오



# 1. 개요

## 2) 기가지니 IoT 제어 시나리오

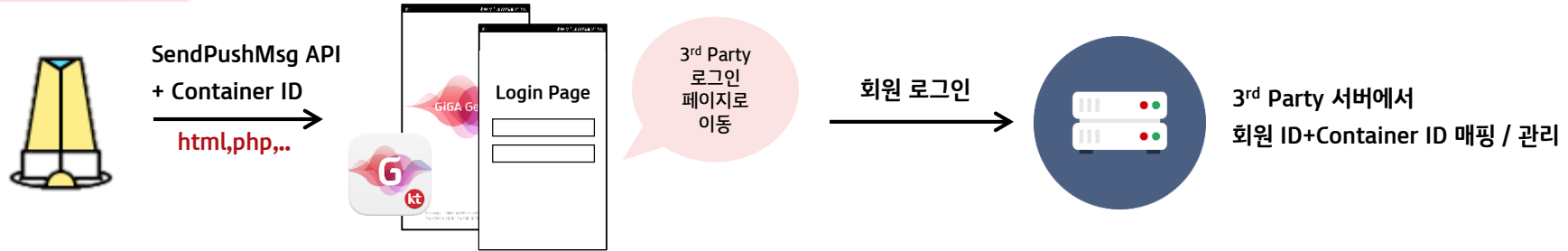
기가지니 웹 앱 서비스와 동일한 구조 + 화면이 없는 단말 제어 서비스 개발



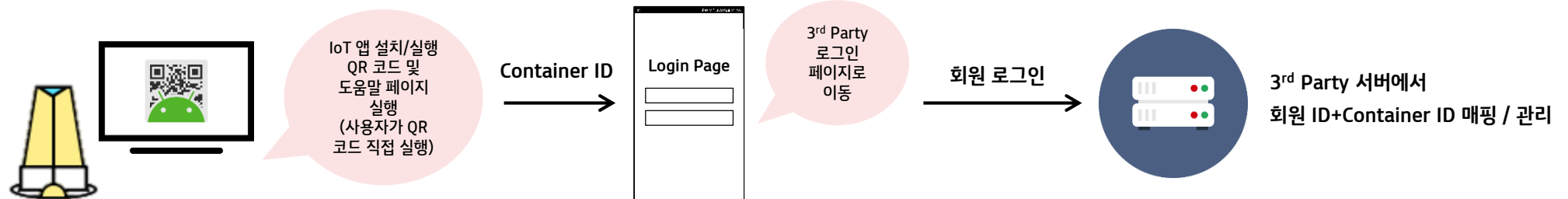
# 1. 개요

## 3) 기가지니 IoT 연동 시나리오

### 최초 연동 방법 1 - Push Msg 사용



### 최초 연동 방법 2 - QR 코드 사용



## 2. 기가지니 개발자포털 작업

My Service > 서비스 선택 > Service SDK [정보 등록/수정]

o 화면 UI 없이, IoT 기기 제어



### GiGA Genie Service SDK 내역

Window 유형 정보 (\*기가지니1은 3월 19일부터 사용 가능, 기가지니2는 사용 가능)

- Full Window With Sound : 전체 화면으로 실행(소리 포함)
- Full Window Without Sound : 전체 화면으로 실행(소리 없음)
- ① - Overlay Window With Sound : 메인화면에 Overlay 되어 실행(소리 포함)
- or - Overlay Window Without Sound : 메인화면에 Overlay 되어 실행(소리 없음)
- ② - Background With Sound : 화면 없이 Background로 실행(소리 포함)
- Background Without Sound : 화면 없이 Background로 실행(소리 없음)

Window 좌표(Overlay Window 유형에서 입력가능)

- TopX : Overlay Window의 좌측상단의 X좌표 (0 ~ 1920)
- TopY : Overlay Window의 좌측상단의 Y좌표 (0 ~ 1080)
- BottomX : Overlay Window의 좌측상단의 X좌표 (0 ~ 1920)
- BottomY : Overlay Window의 좌측상단의 Y좌표 (0 ~ 1080)
- 투명도 : Overlay Window의 투명도 (0: 불투명 → 10: 투명)



예) Overlay Window 유형  
(TopX: 0, TopY: 300, BottomX: 1920, BottomY: 1080)

## 2. 기가지니 개발자포털 작업

My Service > 서비스 선택 > Service SDK [정보 등록/수정]

○ 화면 UI 없이, IoT 기기 제어

화면 UI가 없는 경우,  
Overlay Window Type 또는 Background Type 으로 선택

“기가지니, ○○○ 연결해줘”와 같이  
IoT 기기 연동 요청 발화 시 실행되는 인텐트 (Dialog Kit 등록 guide 참고)

①

1. 인텐트 명	AppPush	삭제
서비스 URL	/app_push.php	
Window 유형	Overlay Window With Sound ▼	
Window 좌표	TopX 0 TopY 0 BottomX 1920 BottomY 1080	투명도 10

or

②

1. 인텐트 명	AppPush	삭제
서비스 URL	/app_push.php	
Window 유형	Background With Sound ▼	

# 3. 소스 작업

## 1) 연동 - App Push API

※ Service SDK -[sendPushMsg API](#)

5	Push 보내기	Companion App 또는 다른 G-Box 에 Push 메시지를 보낸다.	Web → Native → Mobile App	<a href="#">appinfo.sendPushMsg</a>
---	----------	--	---------------------------------	-------------------------------------

### ※ 주의 ※

App Push의 경우, 기가지니와 연결되어 있는 계정으로 동작  
기가지니 홈 > 설정 > 사용자 정보에서 확인 가능

App Push API를 사용할 경우,

[GetUserInfo API](#)를 사용하여, 계정 정보를 알아낸 다음

“[아빠] 계정으로 앱 푸시를 보내드렸어요. 기가지니 모바일 앱에서 확인해보세요” TTS 출력 필수

계정 정보가 없다면, “기가지니 모바일에서 계정 등록 후 사용 가능합니다.” TTS 출력 필수

## 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.

#### ※ 주의 ※

기가지니 단말과 기가지니 앱이 **페어링 되어 있어야** 가능합니다.

페어링이 되어 있지 않을 경우 페어링 요청하는 팝업을 띄워 주세요.

페어링 유무는 `appinfo.getUserInfo`의 `extra.registwithapp` 으로 확인하실 수 있습니다.

- `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`: 기가지니 홈 화면으로 이동하는 Push를 보낸다.
    - `EXEC_BANK`: 기가지니앱 बैंक 화면으로 이동하는 Push를 보낸다.
  - `options.msg` : String, Mandatory로 보내고자 하는 msg이다. `EXEC_WEB` 의 경우 web url 을, `EXEC_CALL` 의 경우 통화에 설정할 전화번호를 설정한다.
  - `options.popuptext` : String, Optional 로 팝업 문구이다. (android만 적용됨), null일 경우 Default 메시지가 전달된다.
- `result_cd` 는 다음과 같이 정의된다.
  - 200: 성공
  - 500: 시스템 Error

# 3. 소스 작업

## 2) 연동 - Container Id 정보

※ Service SDK -[getContainerId API](#)

1	AppID 조회	G-Box 의 해당 App 에 대한 ID 를 조회한다.	Web → Native	<a href="#">appinfo.getContainerId</a>
---	----------	--------------------------------	--------------	--

### 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.");
    }
});
```



## 3. 소스 작업

### 3) 기기 제어 코드 참고

#### ※ 인텐트 처리하는 법

1) 각 인텐트마다 다른 소스 호출 (AppPush.php, LightOn.php...)

2) 모든 인텐트를 하나의 소스 안에서 처리

인텐트 실행 시에 로드 되는 URL 파라미터 값으로 명령 분기

참고) 로드 되는 URL 값

- 베이스URL+서비스URL?Vsid=앱아이디&NE-사전(발화문에 포함된 NE사전)&...~&uword=사용자발화문&actioncode=인텐트명  
(window.location 등의 함수 사용하여 파라미터 값 추출)

예시)

[Dialog Kit] NE-LIGHT(대표어휘: 조명), NE-MAX(대표어휘: 최대) 생성

인텐트: LightOn (어휘사전 조합 - (NE-LIGHT)(PR-B-TURNON))

사용자 발화: “○○에서 조명 최대로 켜줘” (○○은 Invoke Name)

Load URL: (주소)...?~&NE-LIGHT:조명&NE-MAX:최대&...&actioncode=LightOn...

인텐트 조합에 없는 NE 사전이라도, 사용자 발화에 있는 모든 NE 사전 정보 추출 가능

# 3. 소스 작업

## 4) 종료 API

※ Service SDK -[svcFinished API](#)

IoT 제어  
서비스는 기기  
제어 후 강제  
종료 필수

7	서비스 종료처리 완료	서비스 종료 요청에 대한 내부 종료 처리가 완료되었음을 알린다.	Web → Native	<a href="#">voice.svcFinished</a>
---	-------------	-------------------------------------	--------------------	-----------------------------------

### ※ 주의 ※

기기에 명령 전달 후, 시간 지연 함수 (예: `setTimeout`) 사용하여 서비스 종료 (timeout은 1-2초 내로 하여, 사용자가 다음 발화 하기 전에 종료)

## 서비스종료 API

### `gigagenie.voice.svcFinished`

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

종료시 `options.returnParent`를 설정하면 RunApp 으로 실행된 경우 RunApp을 호출한 이전 앱의 `actionCode`로 다시 호출해준다.

- `function svcFinished(options,callback)`
- `options` 은 다음과 같다.
  - `options.returnParent` : boolean, Optional 값으로 RunApp 에 의해 호출된 앱일 경우 종료 후 호출한 앱을 다시 실행한다. ([gigagenie.init.runApp](#) 에서 설정한 `callbackParameter`로 파라미터 전달)
- `extra` 값은 null 이다.
- 사용 예

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

## 4. 참고사항

### ※ Dialog Kit 작성 참고

1. 목적: 기기 제어 - 조명 키기/끄기
2. 발화 가능한 문장: “지니렌탈 조명 켜줘”, “지니렌탈 조명 꺼줘”, “지니렌탈 조명 최대 밝기로 켜줘”, “지니렌탈 조명 최소 밝기로 켜줘”
3. Invoke명: 지니렌탈
4. 인텐트명: LightOn
5. 어휘사전 등록: NE-LIGHT(대표 어휘: 조명), NE-MAX(대표 어휘: 최대), NE-MIN(대표 어휘: 최소)
6. 인텐트 규칙: (NE-LIGHT)(PR-B-TURNON)

인텐트 규칙 조합에 없더라도,  
사용자 발화 문장에 포함되면 해당 어휘 추출 가능

인텐트 상세

인텐트 등록 > 대화 입력/분석 > 인텐트 규칙 추출 > 시뮬레이션

인텐트명 ⓘ

LightOn

삭제

인텐트 등록

대화입력 ⓘ

추가할 대화 (예시: 치즈피자 시켜줘)

대화분석



구문분석

※ 분석결과와의 단어를 더블클릭하시면, 바로 인텐트 규칙으로 추가됩니다.  
※ "STATUS 입력"을 클릭해서 상태 정보도 인텐트 규칙에 추가할 수 있습니다.

대화구문	분석결과	인텐트 규칙
조명 켜줘	<div> <span>NE</span> <span>PR</span> <span>사전미등록</span> </div> <div> <span>조명 [NE-LIGHT/조명]</span> <span>켜줘 [PR-B-TURNON/켜줘]</span> </div> <div> <p>기본 어휘 사전</p> </div>	<div> <span>PR-B-TURNON</span> <span>NE-LIGHT</span> </div> <div> <span>STATUS 입력</span> </div>

## 4. 참고사항

### 참고 페이지

1. 기가지니 개발자 가이드 - <https://github.com/ktadevelopers/GiGAGenie-Developers/wiki/GiGA-Genie-Developers-User-Guide>
2. 자주 묻는 질문 - <https://github.com/ktadevelopers/GiGAGenie-Developers/wiki/자주-묻는-질문>
3. Service SDK doc - <https://github.com/GiGAGenie-ServiceSDK/UserGuide/wiki>
4. Dialog Kit doc - <https://github.com/gigagenieDmt/DialogKit-deploymentGuide/wiki>
5. 개발자포럼 - <https://www.gigagenie.ai/support/qna>