

Dialogflow와 Heroku로 카카오톡 챗봇 만들기

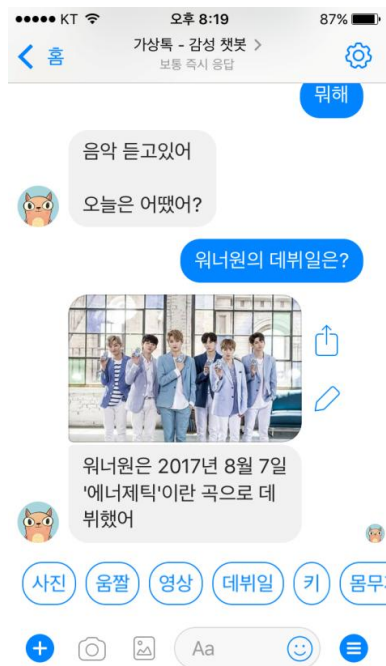
페이스북 챗봇 개발자 모임

우종하

발표자 소개

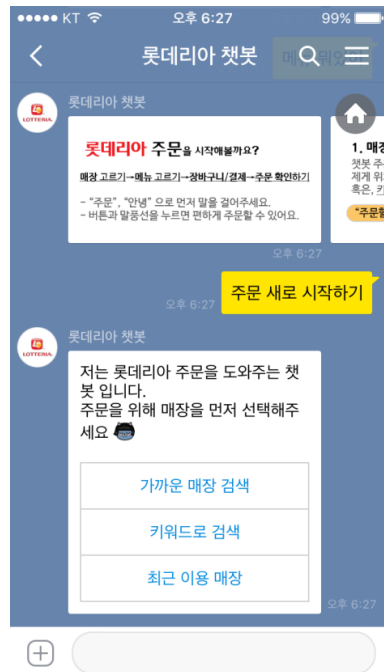
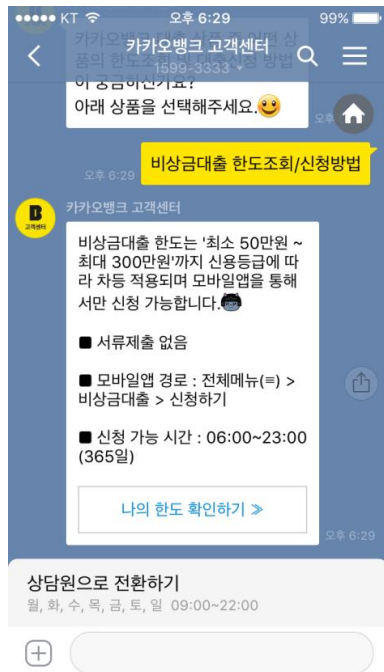
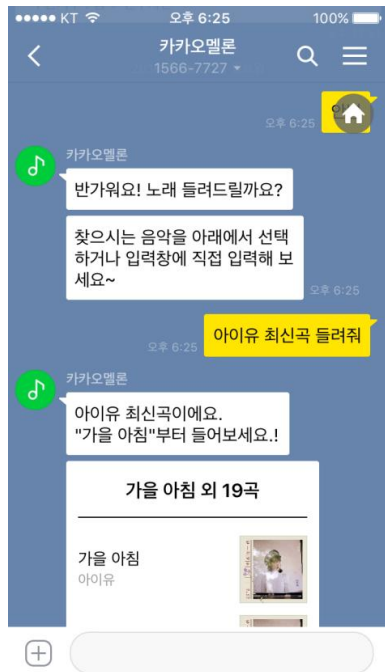


페이스북 챗봇 개발자 모임 운영



가상톡 챗봇 개발

오늘 발표할 내용



나도 카카오톡 챗봇을 만들어보자!

밑바닥부터 시작하는 챗봇 개발?

형태소분석
엔터티 추출
인텐트 분류
스크립트 처리
...

우왕 어렵당 ㅠㅠ

Frameworks



Beep
Boop!

Botkit



Authoring

Chatfuel



Converse

PullString™

text it

Analytics



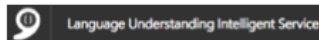
Botlytics

BotMetrics

Language / NLP



init.ai



msg.ai



Playground

wit.ai

RECAST.AI

Connectors

FLOWXO

gupshup

smooch.io

Prompt

챗봇 빌더를 사용하여 손쉽게!

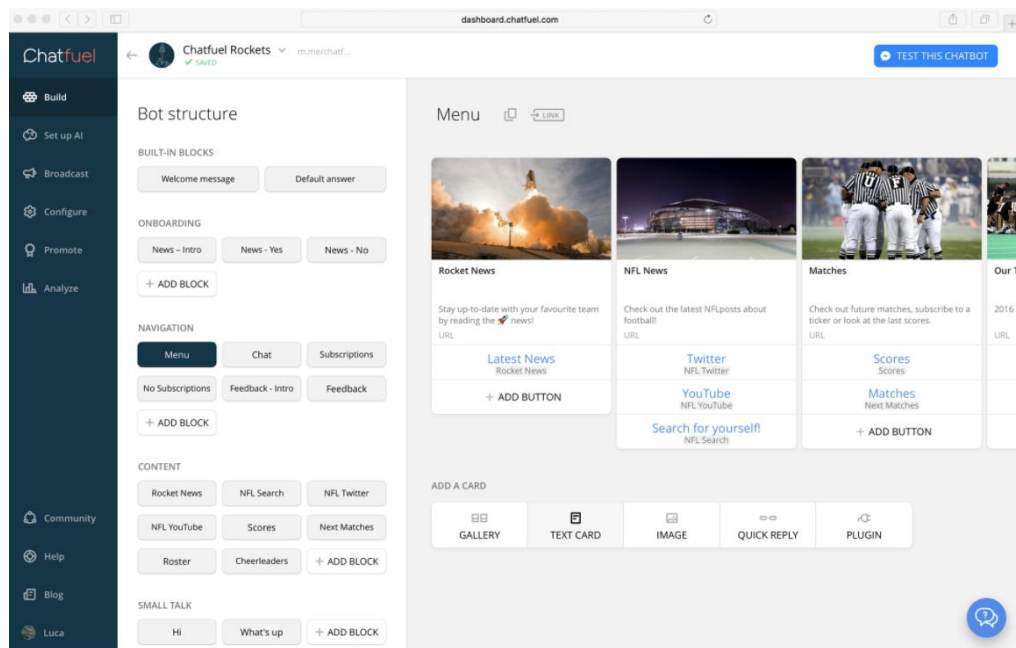
챗봇 빌더 Top 3

The screenshot displays the Watson Conversation web interface. On the left is a dark blue sidebar with icons for workspace management, intents, entities, and dialog. The main area is titled 'Watson Conversation' and shows the workspace 'Workspaces / 피자챗봇'. The 'Intents' tab is selected, displaying a table of intents. The table has columns for 'Intent (1)', 'Description', 'Modified', and 'Examples'. One intent is listed: '#주문' (Order), modified '20 hours ago' with '1' example. Above the table is an 'Add intent' button and icons for refresh, download, and delete. On the right, a 'Try it out' chat window is open, showing a Korean greeting '안녕하세요? 무엇을 도와드릴까요?' (Hello? How can I help you?). The chat window includes a 'Clear' button, a 'Manage Context' button with a blue circle containing the number '2', and a close button. At the bottom of the chat window, there is a text input field with the placeholder 'Enter something to test your bot' and a note 'Use the up key for most recent'.

Intent (1) ▲	Description	Modified ▼	Examples
<input type="checkbox"/> #주문		20 hours ago	1

Watson Assistant

성능은 좋지만 유료...



Chatfuel

무료에 사용이 간단하지만 페이스북만 지원...



무료 이용 가능 (분당 180개 api 요청)

한글 자연어처리 지원

많이 사용하므로 자료 검색 용이

소스 코드 다운받기

deepseasw / dialogflow_heroku_kakaotalk_chatbot

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Dialogflow와 Heroku로 카카오톡 챗봇 만들기

Add topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request

Create new file Upload files Find file Clone or download

deepseasw Add new files

dialogflow	Add new files
doc	Add new files
Profile	Add new files
README.md	Add new files
pizzabot.py	Add new files 4 minutes ago
requirements.txt	Add new files 4 minutes ago

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/deepseasw/dialogflow_heroku_kakaotalk_chatbot

Open in Desktop Download ZIP

Download ZIP 클릭

https://github.com/deepseasw/dialogflow_heroku_kakaotalk_chatbot

피자 주문 챗봇 만들기

Build natural and rich conversational experiences

Give users new ways to interact with your product by building engaging voice and text-based conversational interfaces, such as voice apps and chatbots, powered by AI. Connect with users on your website, mobile app, the Google Assistant, Amazon Alexa, Facebook Messenger, and other popular platforms and devices.

SIGN UP FOR FREE



https://dialogflow.com 접속

GO TO CONSOLE 클릭

Test

CREATE



DEFAULT LANGUAGE ?

Korean (South Korea) — ko



Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

(GMT+9:00) Asia/Tokyo



Date and time requests are resolved using this timezone.

GOOGLE PROJECT

Create a new Google project




Enables Cloud functions, Actions on Google and permissions management.

새로운 프로젝트 생성

제목을 한글로 하면 에러 발생. 영어로 하고 나중에 수정.

메뉴를 살펴보면

 Dialogflow

피자챗봇

ko

Intents

Entities

Knowledge ^[beta]

Fulfillment

Integrations

Training

History

Analytics

Prebuilt Agents

Intents

CREATE INTENT

Search intents

Default Fallback Intent

Default Welcome Intent

pizza_info

pizza_order

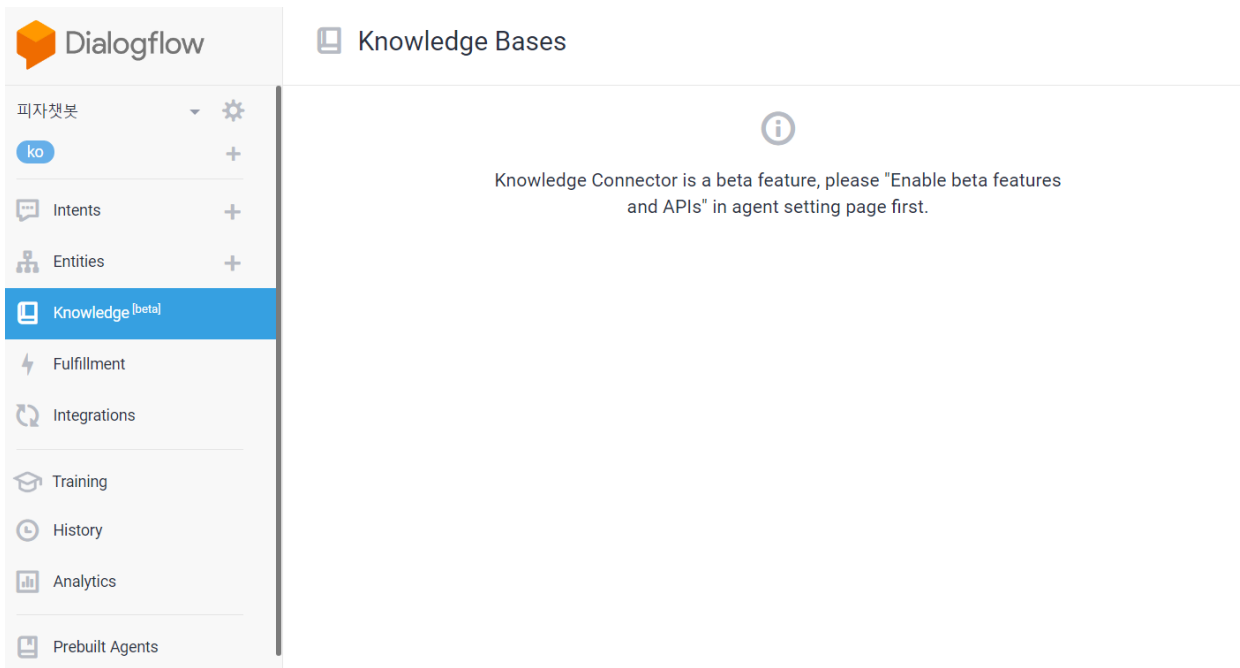
Intents

문장의 의도 및 대답 처리

The screenshot shows the Dialogflow console interface. On the left, a sidebar contains navigation links: '피자챗봇' (with a dropdown and settings icon), 'ko' (with a plus icon), 'Intents' (with a plus icon), 'Entities' (highlighted with a plus icon), 'Knowledge [beta]' (with a plus icon), 'Fulfillment' (with a plus icon), 'Integrations' (with a plus icon), 'Training', 'History', 'Analytics', and 'Prebuilt Agents'. The main content area is titled 'Entities' and features a 'CREATE ENTITY' button. Below the title is a search bar labeled 'Search entities' with a magnifying glass icon. A list of entities is displayed below the search bar, showing '@pizza_type'.


Entities

문장안에 들어있는 개체 정보



Knowledge Bases

웹페이지나 문서에서 정보를 찾아 대답. 현재 영어만.

 Dialogflow

피자챗봇

ko

+

Intents

+

Entities

+

Knowledge ^[beta]

⚡ Fulfillment

↻ Integrations

🎓 Training

🕒 History

📊 Analytics

📄 Prebuilt Agents

⚡ Fulfillment

Webhook

ENABLED ☒

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL*

BASIC AUTH

Enter username

Enter password

HEADERS

Enter key

Enter value

Enter key

Enter value

+ Add header

DOMAINS

Disable webhook for all domains

Inline Editor


(Powered by Cloud Functions for Firebase)

DISABLED ☐

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

Fulfillment

외부 서버와 Restful API로 정보를 주고받음

 Dialogflow

피자챗봇


ko

Intents

Entities

Knowledge ^[beta]

Fulfillment


 Integrations


Training




History

Analytics

Prebuilt Agents

 Integrations






Google Assistant


Build Actions for the Google Assistant to reach users through Google Home, Android phones, and more devices.

[INTEGRATION SETTINGS](#)




Web Demo

☐




Facebook Messenger

☐



Dialogflow Phone Gateway ^{BETA}

☐



Slack

☐

Integrations

메신저나 웹, 음성비서와 연동. 카카오톡 지원 안함.

다운로드한 파일 로드

The screenshot shows the Dialogflow web interface. On the left, a sidebar contains the Dialogflow logo and a list of items: '피자챗봇' (with a gear icon highlighted by a red dashed box), 'ko', 'Intents', 'Entities', 'Fulfillment', and 'Integrations'. The main area is titled '피자챗봇' and has a tab bar with 'General', 'Languages', 'ML Settings', 'Export and Import' (which is selected), and 'Share'. Below the tab bar, there are three blue buttons: 'EXPORT AS ZIP', 'RESTORE FROM ZIP' (highlighted with a red dashed box), and 'IMPORT FROM ZIP'. To the right of each button is a descriptive text: 'Create a backup of the agent', 'Replace the current agent version \ version will be deleted.', and 'Upload new intents and entities wit the same name will be replaced wit'.

dialogflow/pizzabot_dialogflow.zip

파일에 저장된 Dialogflow 프로젝트를 복원

엔터티 추가

pizza_type

SAVE



Define synonyms ?



Allow automated expansion

불고기피자	불고기피자, 불고기맛피자
포테이토피자	포테이토피자, 포태이토피자
페퍼로니피자	페퍼로니피자, 페파로니피자
Click here to edit entry	

+ Add a row

엔터티란 문장에서 추출하는 핵심 단어

같은 항목에 동의어를 여러개 입력 가능

가장 중요한 인텐트 설정!

Search intents



Default Fallback Intent



Default Welcome Intent



pizza_info



pizza_order ▾

기본 인텐트 두개

Fallback은 일치하는 의도가 없을때 실행. Welcome은 인사말.

• pizza_info

SAVE



Training phrases ?

Search training phrases



” Add user expression

” 포테이토피자는 어떤거야?

” 불고기피자에 대해 설명해봐

” 페퍼로니피자가 뭐야?

” 포테이토피자 알려줘

” 불고기피자 보여줘

” 메뉴

” 메뉴 보여줘

피자 정보 인텐트 추가

Training phrases에 예시 문장들을 입력

- pizza_info

SAVE

Action and parameters ?

pizza_info

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?	PROMPTS ?
<input checked="" type="checkbox"/>	pizza_type	@pizza_type	\$pizza_type	<input type="checkbox"/>	피자 종류를 선택하세요 1...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

+ New parameter

Action and parameters

이름에 'pizza_info' 입력. 예문에서 자동으로 엔터티 인식.

- pizza_info

SAVE

Action and parameters ?

pizza_info

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?	PROMPTS ?
<input checked="" type="checkbox"/>	pizza_type	@pizza_type	\$pizza_type	<input type="checkbox"/>	피자 종류를 선택 하세요 1...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

+ New parameter

REQUIRED 체크

슬롯 채우기. 문장에 엔터티가 없다면 PROMPTS 내용을 질문.

Prompts for "pizza_type" ×

NAME	ENTITY	VALUE
pizza_type	@pizza_type	\$pizza_type

PROMPTS

1 피자 종류를 선택하세요

1. 불고기피자
2. 페퍼로니피자
3. 포테이토피자

2 Enter a prompt variant

Close

PROMPTS

필요한 엔터티를 입력하도록 질문

• pizza_info

SAVE

Responses ?



DEFAULT +

Text response



1 \$pizza_type의 정보

2 Enter a text response variant



ADD RESPONSES



Set this intent as end of conversation ?

Responses 설정

파라미터에서 정의된 엔터티 변수 \$pizza_type 사용

불고기피자에 대해 알려줘



See how it works in [Google Assistant](#).



Agent

Domains

USER SAYS

COPY CURL

불고기피자에 대해 알려줘



DEFAULT RESPONSE



불고기피자의 정보

INTENT

[pizza_info](#)

ACTION

[pizza_info](#)

PARAMETER

VALUE

[pizza_type](#)

불고기피자

메뉴 보여줘



See how it works in [Google Assistant](#).



Agent

Domains

USER SAYS

COPY CURL

메뉴 보여줘



DEFAULT RESPONSE



피자 종류를 선택하세요 1. 불고기피자 2. 페퍼로니피자 3. 포테이토피자

CONTEXTS

[RESET CONTEXTS](#)

[pizza_info_dialog_context](#)

[pizza_info_dialog_params_pizza_type](#)

45eca394-80d3-4679-9df1-aeaa1e8a0fd8_id_dialog_context

여기까지가 가장 기본적인 기능

• pizza_info

SAVE

Responses ?



DEFAULT +

Text response



1 \$pizza_type의 정보

2 Enter a text response variant



ADD RESPONSES



Set this intent as end of conversation ?

파라미터에 따라 다른 대답을 보여줄 수 없을까?

불고기피자 -> 불고기피자 소개, 포테이토피자 -> 포테이토피자 소개

Fulfillment로 기능 확장

• pizza_info

SAVE

Fulfillment ?




Enable webhook call for this intent



Enable webhook call for slot filling

pizza_info 인텐트에서 Fulfillment 체크

 Dialogflow

피자챗봇

ko

+

Intents

+

Entities

+

Knowledge (beta)

+

⚡ Fulfillment

↻ Integrations

🎓 Training

🕒 History

📊 Analytics

👤 Prebuilt Agents

⚡ Fulfillment

Webhook

ENABLED ☒

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL*

BASIC AUTH

HEADERS

➕ Add header

DOMAINS

Inline Editor

(Powered by Cloud Functions for Firebase)

DISABLED ☐

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

Fulfillment

Webhook과 Inline Editor 중 하나만 선택 가능

⚡ Fulfillment

webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

Inline Editor (Powered by Cloud Functions for Firebase)



Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

```
index.js package.json
1 'use strict';
2
3 const functions = require('firebase-functions');
4 const {WebhookClient} = require('dialogflow-fulfillment');
5 const {Card, Suggestion} = require('dialogflow-fulfillment');
6
7 process.env.DEBUG = 'dialogflow:debug';
8
9 exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
10   const agent = new WebhookClient({ request, response });
11   const parameter = request.body.queryResult.parameters;
12
13   function welcome(agent) {
14     agent.add('Welcome to my agent!');
15   }
16
17   function fallback(agent) {
18     agent.add('I didn't understand');
19   }
20
21   let intentMap = new Map();
22   intentMap.set('Default Welcome Intent', welcome);
23   intentMap.set('Default Fallback Intent', fallback);
24   agent.handleRequest(intentMap);
25 });
```

[View execution logs in the Firebase console](#)

Last deployed on 07/29/2018 15:13

DEPLOY

Inline Editor 사용

파이어베이스의 클라우드 함수. 자바스크립트로 홈페이지에서 직접 코드 작성.

```
3 'use strict';
4
5 const functions = require('firebase-functions');
6 const {WebhookClient} = require('dialogflow-fulfillment');
7 const {Card, Suggestion} = require('dialogflow-fulfillment');
8
9 process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements
10
11 exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
12   const agent = new WebhookClient({ request, response });
13   const parameter = request.body.queryResult.parameters;
14   console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
15   console.log('Dialogflow Request body: ' + JSON.stringify(request.body));
```

함수 구조

agent로 웹훅 메시지 처리. parameter로 엔터티 정보 얻음.

- pizza_info

SAVE

Action and parameters ?

pizza_info						
REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?	PROMPTS ?	
<input checked="" type="checkbox"/>	pizza_type	@pizza_type	\$pizza_type	<input type="checkbox"/>	피자 종류를 선택하세요 1...	
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—	

+ New parameter

Action and parameters

액션과 파라미터의 정보를 Inline editor에 보냄

```

26 function pizza_info(agent) {
27     var answer;
28
29     if( parameter.pizza_type == "불고기피자" )
30     {
31         answer = '한국의 맛 불고기를 부드러운 치즈와 함께!';
32     }
33     else if( parameter.pizza_type == "페퍼로니피자" )
34     {
35         answer = '고소한 페퍼로니햄이 쫄득한 치즈위로 씹썩!';
36     }
37     else if( parameter.pizza_type == "포테이토피자" )
38     {
39         answer = '저칼로리 감자의 담백한 맛!';
40     }
41
42     agent.add(answer);
43 }
44
45 // Run the proper function handler based on the matched Dialogflow intent name
46 let intentMap = new Map();
47 intentMap.set('Default Welcome Intent', welcome);
48 intentMap.set('Default Fallback Intent', fallback);
49 intentMap.set('pizza_info', pizza_info);
50 agent.handleRequest(intentMap);
51 });

```

pizza_info 인텐트 추가

intentMap에 함수 이름 설정. parameter의 pizza_type에 따라 대답 설정.

⚡ Fulfillment

```
16
17 function welcome(agent) {
18   agent.add('Welcome to my agent!');
19 }
20
21 function fallback(agent) {
22   agent.add('I didn't understand');
23   agent.add('I'm sorry, can you try again?');
24 }
25
26 function pizza_info(agent) {
27   var answer;
28
29   if( parameter.pizza_type == "볼고기피자" )
30   {
31     answer = '한국의 맛 볼고기를 부드러운 치즈와 함께!';
32   }
33   else if( parameter.pizza_type == "페퍼로니피자" )
34   {
35     answer = '고소한 페퍼로니 햄이 풍부한 치즈위로 톡톡!';
36   }
37   else if( parameter.pizza_type == "포테이토피자" )
38   {
39     answer = '저칼로리 감자의 달콤한 맛!';
40   }
41
42   agent.add(answer);
43 }
44
45 // Run the proper function handler based on the matched Dialogflow intent name
46 let intentMap = new Map();
47 intentMap.set('Default Welcome Intent', welcome);
48 intentMap.set('Default Fallback Intent', fallback);
49 intentMap.set('pizza_info', pizza_info);
50 agent.handleRequest(intentMap);
51 });
52
```

[View execution logs in the Firebase console](#)

Last deployed on 07/31/2018 17:57

DEPLOY

DEPLOY로 파이어베이스 서버에 전송

View execution logs in the Firebase console에서 디버그 가능

불고기피자가 뭐야?



See how it works in [Google Assistant](#).



Agent

USER SAYS

[COPY CURL](#)

불고기피자가 뭐야?



DEFAULT RESPONSE



한국의 맛 불고기를 부드러운 치즈와 함께!

INTENT

[pizza_info](#)

ACTION

[pizza_info](#)

PARAMETER

VALUE

[pizza_type](#)

불고기피자

포테이토피자에 대해 알려줘



See how it works in [Google Assistant](#).



Agent

USER SAYS

[COPY CURL](#)

포테이토피자에 대해 알려줘



DEFAULT RESPONSE



저칼로리 감자의 담백한 맛!

INTENT

[pizza_info](#)

ACTION

[pizza_info](#)

PARAMETER

VALUE

[pizza_type](#)

포테이토피자

⚡ Fulfillment

webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

Inline Editor (Powered by Cloud Functions for Firebase)

ENABLED



Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

index.js package.json



```
1 'use strict';
2
3 const functions = require('firebase-functions');
4 const {WebhookClient} = require('dialogflow-fulfillment');
5 const {Card, Suggestion} = require('dialogflow-fulfillment');
6
7 process.env.DEBUG = 'dialogflow:debug';
8
9 exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
10   const agent = new WebhookClient({ request, response });
11   const parameter = request.body.queryResult.parameters;
12
13   function welcome(agent) {
14     agent.add('Welcome to my agent!');
15   }
16
17   function fallback(agent) {
18     agent.add('I didn't understand');
19   }
20
21   let intentMap = new Map();
22   intentMap.set('Default Welcome Intent', welcome);
23   intentMap.set('Default Fallback Intent', fallback);
24   agent.handleRequest(intentMap);
25 });
```

[View execution logs in the Firebase console](#)

Last deployed on 07/29/2018 15:13

DEPLOY

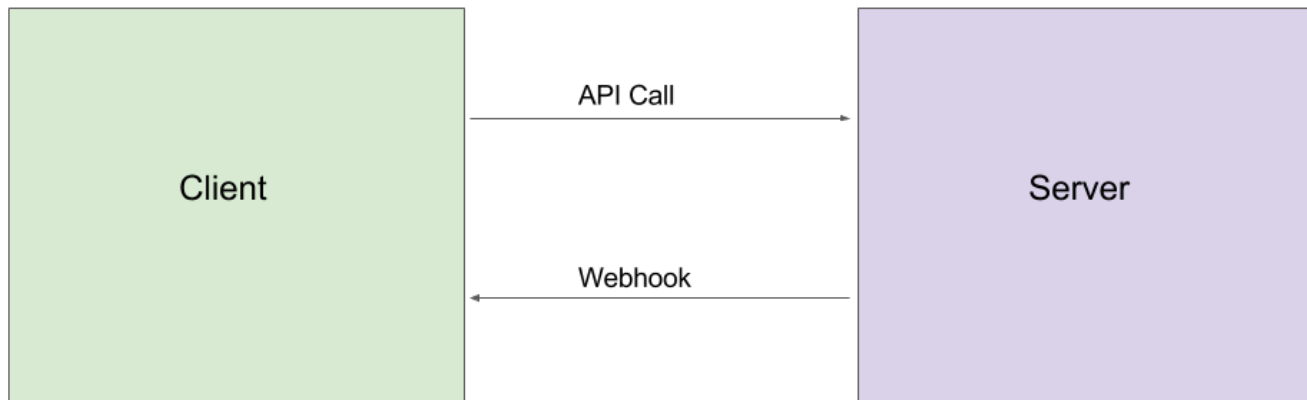
Inline Editor의 한계

확장성 떨어짐. index 파일 하나로 작업.

Webhook을 사용하자!



웹(Web)에서 내 서버의 URL을 걸어(Hook)놓음



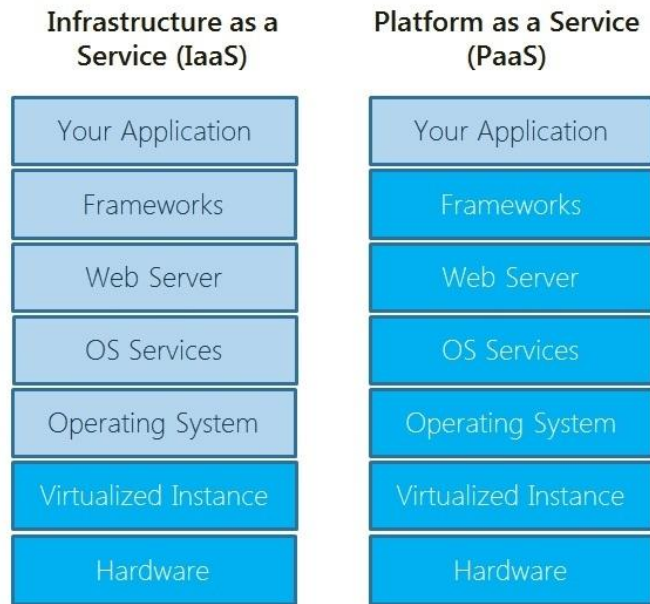
서버에서 특정 작업을 등록된 웹훅 URL로 알리는 것

Heroku로 웹서버 구축



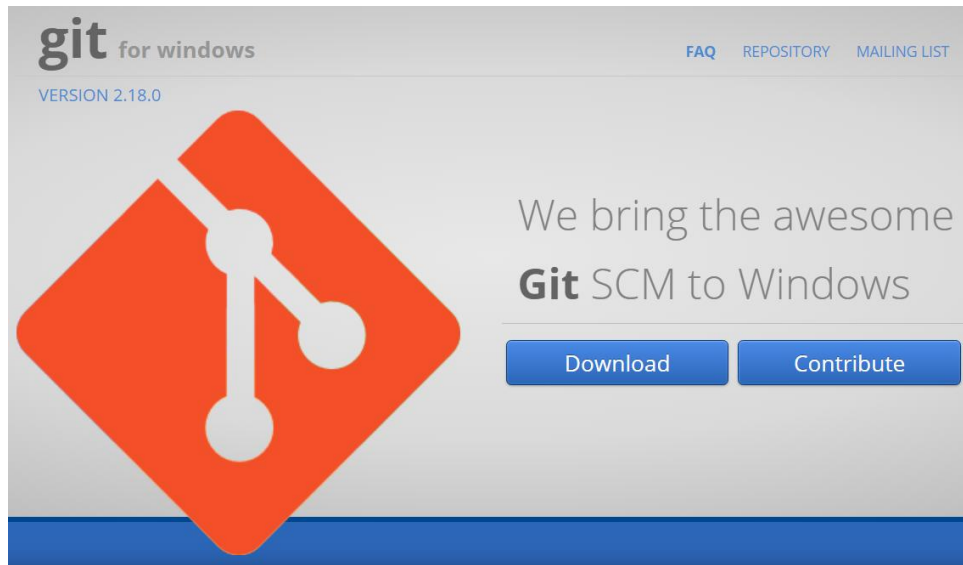
Heroku는 PaaS의 한 종류

무료 사용 가능. 손 쉽게 웹서버 구축.



AWS와 Heroku의 차이

개발 환경을 설치하고 관리할 필요 없이 코드와 데이터만 올리면 실행 가능



Heroku에 코드를 업로드하는 Git 설치

<http://dev-gabriel.tistory.com/21>



Already have an account?

Log in

Sign up for free and
experience Heroku today

Free account

Create apps, connect databases and add-on services, and collaborate on your apps, for free.

Your app platform

A platform for apps, with app management & instant scaling, for development and production.

First name *

Last name *

Email address *

Company name

Heroku 가입

<https://www.heroku.com/>

Download and install



The Heroku CLI requires Git, the popular version control system. If you don't already have Git installed, complete the following before installing the CLI:

- [Git installation](#)
- [First-time Git setup](#)

macOS

[Download the installer](#)

Also available via Homebrew:

```
$ brew install heroku/brew/heroku
```

Windows

Download the appropriate installer for your Windows installation:

[64-bit installer](#)

[32-bit installer](#)

Heroku CLI 설치

<https://devcenter.heroku.com/articles/heroku-cli>

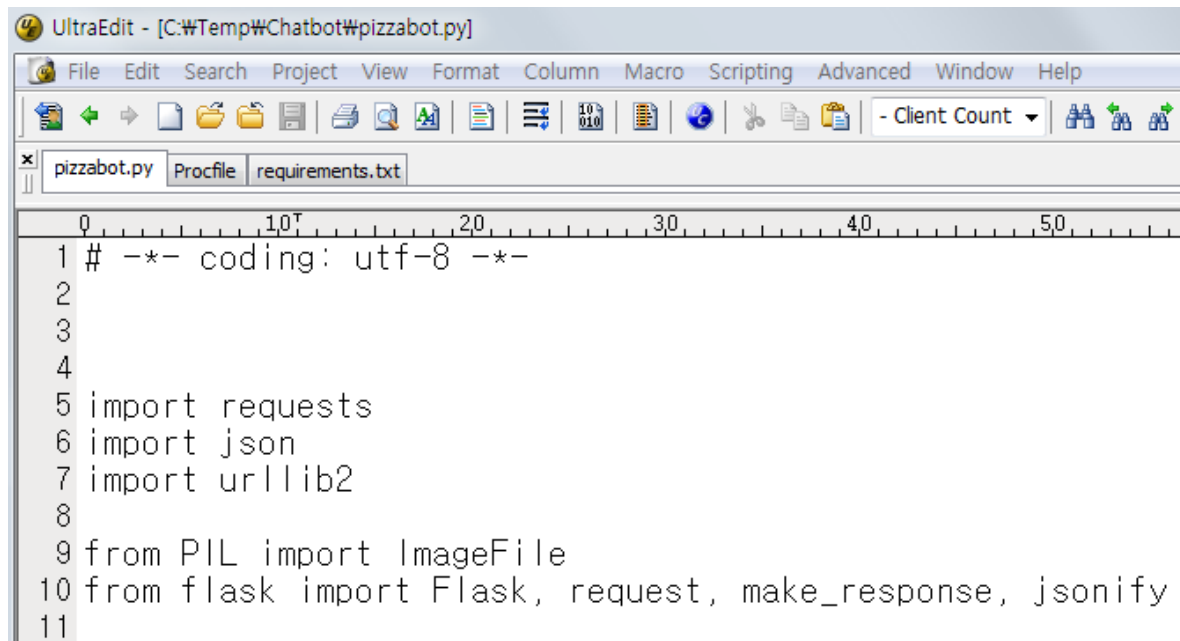
```
c:\WTemp\Chatbot>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 8256-03F8

c:\WTemp\Chatbot 디렉터리

2018-08-09 오후 06:09 <DIR> .
2018-08-09 오후 06:09 <DIR> ..
2018-08-08 오후 07:22      8,526 pizzabot.py
2018-08-08 오후 06:44        29 Procfile
2018-08-08 오후 06:31       36 requirements.txt
                3개 파일            8,591 바이트
                2개 디렉터리 44,556,173,312 바이트 남음

c:\WTemp\Chatbot>
```

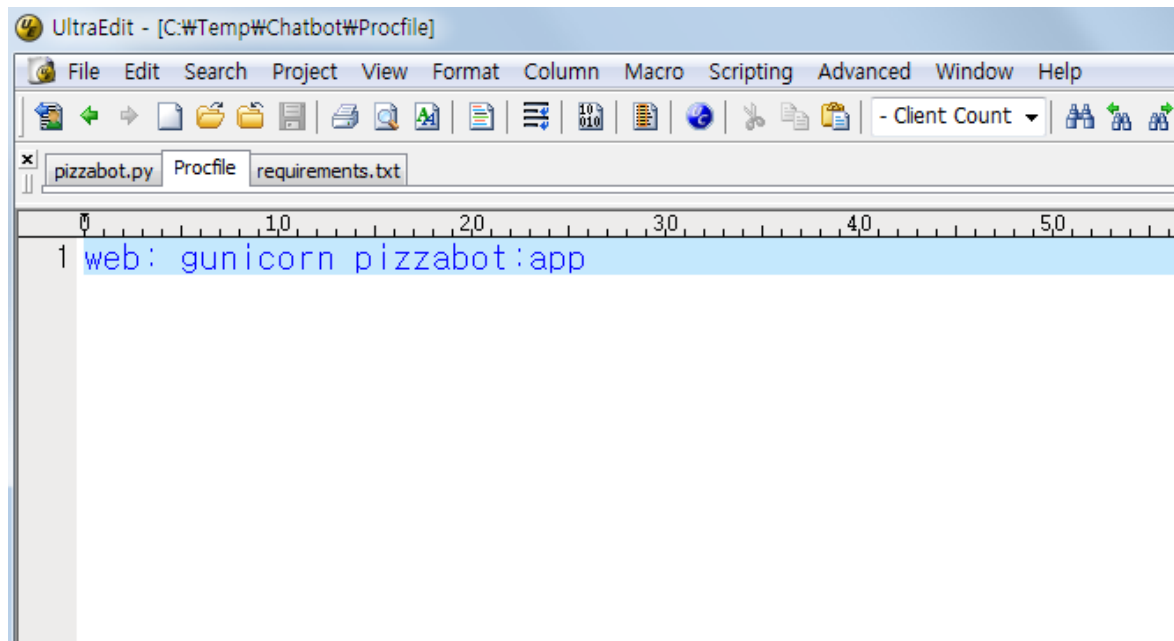
새로운 폴더에 다운로드한 파일 복사



```
UltraEdit - [C:\Temp\Chatbot\pizzabot.py]
File Edit Search Project View Format Column Macro Scripting Advanced Window Help
- Client Count
pizzabot.py Profile requirements.txt
0 10 20 30 40 50
1 # -*- coding: utf-8 -*-
2
3
4
5 import requests
6 import json
7 import urllib2
8
9 from PIL import ImageFile
10 from flask import Flask, request, make_response, jsonify
11
```

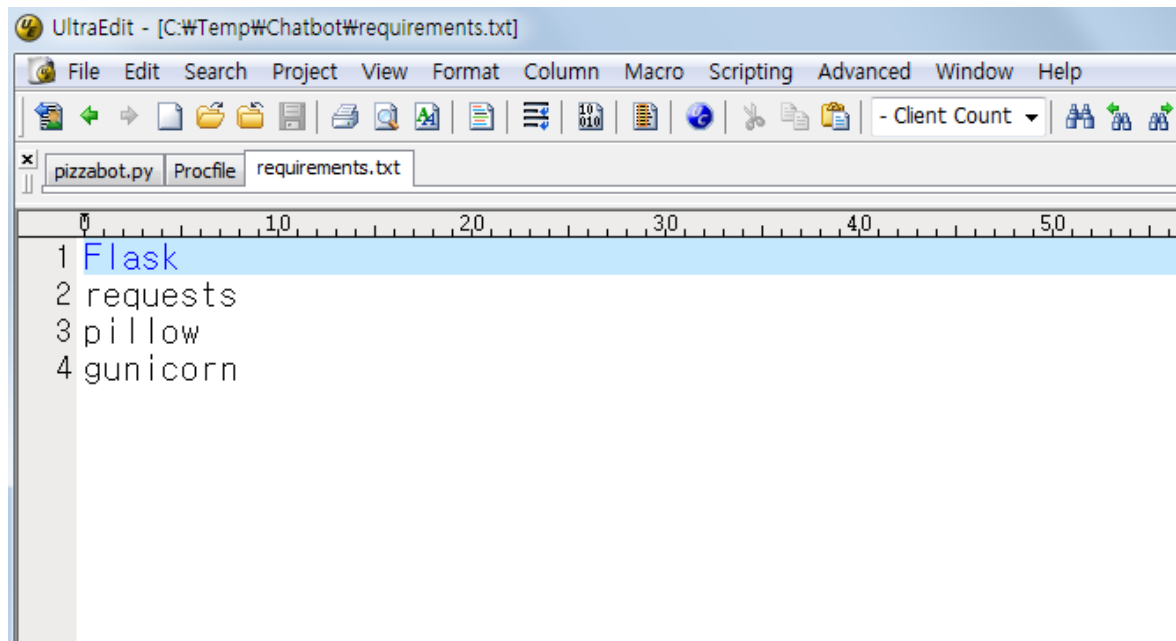
pizzabot.py

Dialogflow와 카카오톡 웹훅 파이썬 코드



Procfile

Heroku의 실행 정보 파일

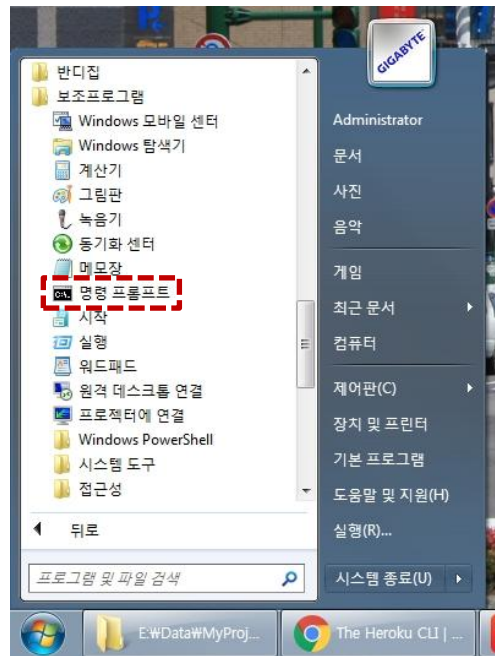
A screenshot of the UltraEdit text editor window. The title bar reads "UltraEdit - [C:\Temp\Chatbot\requirements.txt]". The menu bar includes File, Edit, Search, Project, View, Format, Column, Macro, Scripting, Advanced, Window, and Help. The toolbar contains various icons for file operations and editing. The tab bar shows "pizzabot.py", "Procfile", and "requirements.txt". The main text area displays the following content:

```
1 Flask
2 requests
3 pillow
4 unicorn
```

requirements.txt

프로그램에서 필요한 파이썬 라이브러리 목록. Heroku Deploy시 설치됨.

Heroku에 파일 업로드 및 실행



명령 프롬프트 실행

```
c:\Temp\Chatbot>heroku create dialogflow-kakao-chatbot
Creating    dialogflow-kakao-chatbot... !
!          Invalid credentials provided.
heroku: Enter your login credentials
Email: [REDACTED]@gmail.com
Password: *****
Logged in as [REDACTED]@gmail.com
Creating    dialogflow-kakao-chatbot... done
https://dialogflow-kakao-chatbot.herokuapp.com/ ! https
low-kakao-chatbot.git
```

Heroku 앱 생성

heroku create + 앱이름. 로그인 정보 입력.


```
c:\Temp\Chatbot>git init
Initialized empty Git repository in c:/Temp/Chatbot/.git/

c:\Temp\Chatbot>git add *

c:\Temp\Chatbot>git commit -m "Add files"
[master (root-commit) b22e8a1] Add files
 3 files changed, 325 insertions(+)
 create mode 100644 Procfile
 create mode 100644 pizzabot.py
 create mode 100644 requirements.txt
```

Git 설정

Git 폴더 초기화. 파일 추가. 로컬 저장소 커밋.

```
c:\Temp\Chatbot>heroku git:remote -a dialogflow-kakao-chatbot
set git remote heroku to https://git.heroku.com/dialogflow-kakao-chatbot.git

c:\Temp\Chatbot>git push heroku master
Everything up-to-date

c:\Temp\Chatbot>heroku ps
Free dyno hours quota remaining this month: 982h 58m (98%)
For more information on dyno sleeping and how to upgrade, see:
https://devcenter.heroku.com/articles/dyno-sleeping

=== web <Free>: gunicorn pizzabot:app <1> 앱 프로세스 실행 확인
web.1: up 2018/08/09 19:52:54 +0900 <~ 1m ago>
```

Git 원격 저장소 설정 및 푸시

푸시가 되면 자동으로 필요한 라이브러리 설치하며 deploy 됨

```
c:\Temp\Chatbot>heroku logs
2018-08-09T10:51:56.513614+00:00 app[api]: Enable Logplex by user
l.com
2018-08-09T10:51:56.391517+00:00 app[api]: Initial release by use
il.com
2018-08-09T10:51:56.513614+00:00 app[api]: Release v2 created by
gmail.com
2018-08-09T10:51:56.391517+00:00 app[api]: Release v1 created by
gmail.com
2018-08-09T10:52:19.000000+00:00 app[api]: Build started by user
.com
2018-08-09T10:52:46.575382+00:00 app[api]: Deploy f979836f by use
il.com
```

로그 확인

만약 앱 프로세스가 crash 되었다면 에러가 있는지 검사

```
c:\Temp\Chatbot>git add *  
  
c:\Temp\Chatbot>git commit -m "Modify files"  
[master 8a8bf8f] Modify files  
1 file changed, 1 insertion(+), 1 deletion(-)  
  
c:\Temp\Chatbot>git push heroku master  
Enumerating objects: 5, done.  
Counting objects: 100% (5/5), done.  
Delta compression using up to 4 threads.  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 349 bytes | 349.00 KiB/s, done.  
Total 3 (delta 1), reused 0 (delta 0)
```

파일 수정 후 업데이트

파일 추가. 로컬 저장소 커밋. Heroku 저장소로 푸시

Heroku 무료버전 문제 1

한달에 550시간 제한. $30\text{일} \times 24\text{시간} = 720\text{시간}$ 필요.



Jump to Favorites, Apps, Pipelines, Spaces...



Personal ▾



dialogflow-kakao-chatbot



Jongha Woo



Account settings



Notifications



Sign out

Heroku 사이트 접속

프로필 -> Account settings

Account

Applications

Billing

Billing Information

This account is billed to:  MasterCard ending in

Update Credit Card

Remove Credit Card

Current Usage

\$0.00

[View Details](#)

Platform Credits

\$0.00

Free Dyno Usage



28.60 free dyno hours (2.86%) used this month

971.40 free dyno hours remaining this month

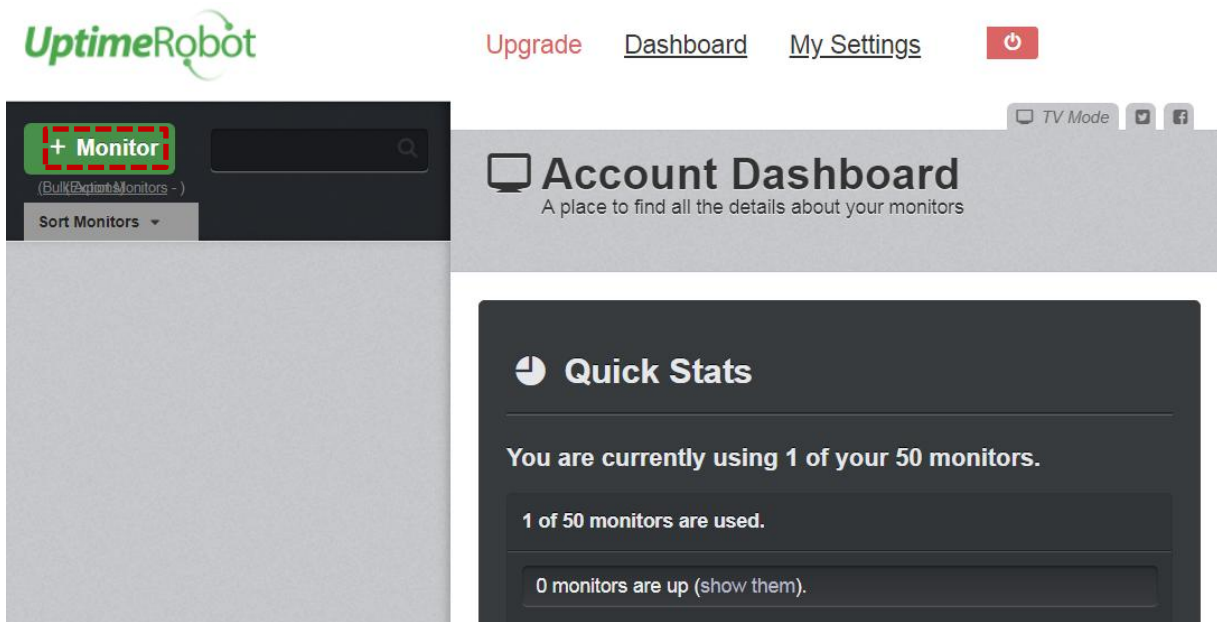
[Find out more](#)

Billing 메뉴에서 신용카드 추가

450시간 보너스. 무료버전 사용하면 결제 안됨.

Heroku 무료버전 문제 2

30분간 요청 없으면 Sleep. 새로운 요청이 오면 5~10초후 재시작.



<https://uptimerobot.com>

새로운 Monitor 추가

New Monitor

Monitor Information

Monitor Type *

HTTP(s)

Friendly Name *

Pizzabot

URL (or IP) *

https://dialogflow-kakao-chatb

Monitoring Interval *

every 5 minutes

Advanced Settings (Optional) show/hide

Select "Alert Contacts To Notify"

Selected: 0 of 1 (hide).

Search

Type

Alert Contact

☐

☒

New alert contacts can be defined from the "My Settings" page.

Close

Create Monitor

매 시간마다 자동으로 등록한 사이트에 요청 수행

URL에 https://앱이름.herokuapp.com/keyboard 입력

웹훅 파이썬 코드

```
1 # -*- coding: utf-8 -*-
2
3 from flask import Flask, request, make_response, jsonify
4
5 app = Flask(__name__)
6
7
8
9 @app.route('/', methods=[ 'POST' ])
10 def webhook():
11
12     return 'Hello'
13
14
15
16 if __name__ == '__main__':
17     app.run(host='0.0.0.0', port = 5110, threaded=True)
18
19
```

파이썬 Flask

이게 끝. 단순하지만 강력한 기능.

- pizza_info

SAVE

Action and parameters ?

pizza_info

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?	PROMPTS ?
<input checked="" type="checkbox"/>	pizza_type	@pizza_type	\$pizza_type	<input type="checkbox"/>	피자 종류를 선택하세요 1...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

+ New parameter

Action and parameters

액션과 파라미터의 정보를 Webhook에 보냄

```

1
2 @app.route('/', methods=[ 'POST' ])
3 def webhook():
4
5     # 액션 구함
6     req = request.get_json(force=True)
7     action = req[ 'result' ][ 'action' ]
8
9     # 액션 처리
10    if action == 'pizza_info':
11        pizza_name = req[ 'result' ][ 'parameters' ][ 'pizza_type' ]
12        answer = process_pizza_info(pizza_name)
13    else:
14        answer = 'error'
15
16    res = { 'speech': answer }
17
18    return jsonify(res)
19

```

피자 정보 웹훅 처리

action 이름으로 구분. pizza_type에 따라 대답 처리.


```

1
2 def process_pizza_info(pizza_name):
3
4     if pizza_name == u'불고기피자':
5         answer = '<Photo>http://www.pizzamaru.co.kr/UpFile/Menu/c1a8_F_01.jpg</Photo>'
6         answer += '한국의 맛 불고기를 부드러운 치즈와 함께!'
7     elif pizza_name == u'페퍼로니피자':
8         answer = '<Photo>http://www.pizzamaru.co.kr/UpFile/Menu/c1a6_F_01.jpg</Photo>'
9         answer += '고소한 페파로니햄이 쫄득한 치즈위로 뚝뚝!'
10    elif pizza_name == u'포테이도피자':
11        answer = '<Photo>http://www.pizzamaru.co.kr/UpFile/Menu/c1a7_F_01.jpg</Photo>'
12        answer += '저칼로리 감자의 담백한 맛!'
13
14    return answer
15

```

피자 이름에 맞게 대답 설정

<Photo></Photo>는 사진 Url. 나중에 카카오톡 형식에 맞게 변형.

 Dialogflow

피자챗봇

ko

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

SAVE

General

Languages

ML Settings

Export and Import

Speech

Share

☐

V2 API




Use [Cloud API](#) as default for the agent. Your webhook will receive [V2 format requests](#) and should return [V2 format responses](#).

☒

V1 API

Legacy APIs

API KEYS (V1)

Client access token	adfb4242e4a04	 
Developer access token	48a8bb88df264	

설정에서 V1 API로 변경

V2는 기능이 많지만 복잡. V1으로 Inline editor 사용 불가

Dialogflow

피자챗봇

ko

Intents

Entities

Knowledge [beta]

Fulfillment

Integrations

Training

History

Analytics

Prebuilt Agents

⚡ Fulfillment

Webhook

ENABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL*

https://dialogflow-kakao-chatbot.herokuapp.com/

BASIC AUTH

Enter username

Enter password

HEADERS

Enter key

Enter value

Enter key

Enter value

+ Add header

DOMAINS

Disable webhook for all domains

Inline Editor

(Powered by Cloud Functions for Firebase)

DISABLED

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

Fulfillment 설정

URL에 https://앱이름.herokuapp.com 입력

불고기피자가 뭐야?



See how it works in [Google Assistant](#).



Agent

Domains

USER SAYS

[COPY CURL](#)

불고기피자가 뭐야?



DEFAULT RESPONSE



<Photo><http://www.pizzamaru.co.kr/UpFile/...>
한국의 맛 불고기를 부드러운 치즈와 함께!

INTENT

[pizza_info](#)

ACTION

[pizza_info](#)

PARAMETER

VALUE

[pizza_type](#)

불고기피자

포테이토피자에 대해 알려줘



See how it works in [Google Assistant](#).



Agent

Domains

USER SAYS

[COPY CURL](#)

포테이토피자에 대해 알려줘



DEFAULT RESPONSE



<Photo><http://www.pizzamaru.co.kr/UpFile/...>
저칼로리 감자의 담백한 맛!

INTENT

[pizza_info](#)

ACTION

[pizza_info](#)

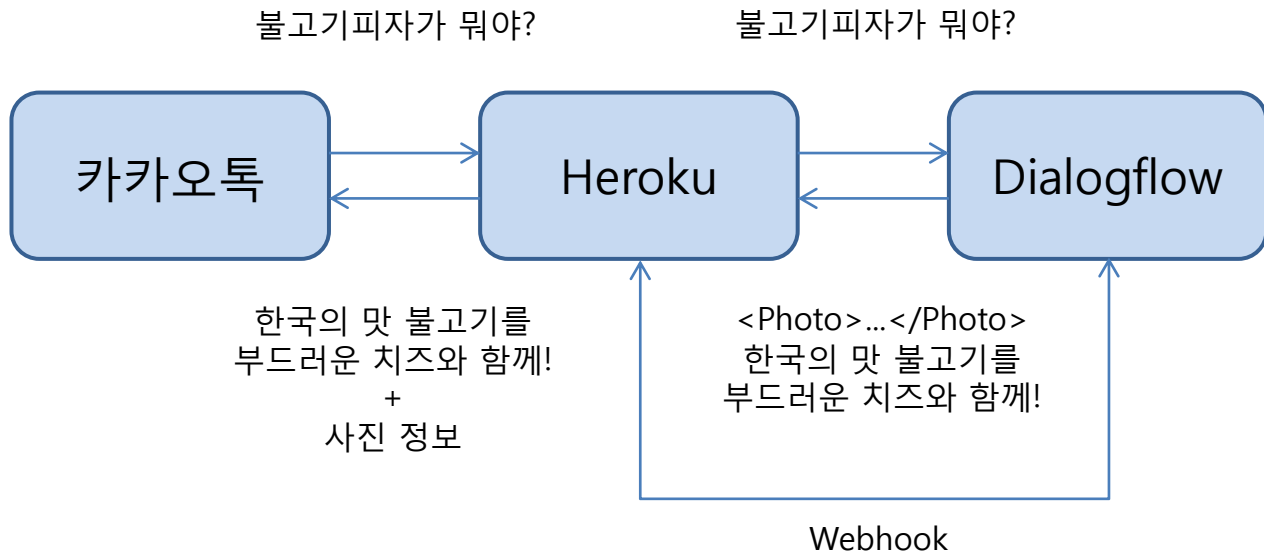
PARAMETER

VALUE

[pizza_type](#)

포테이토피자

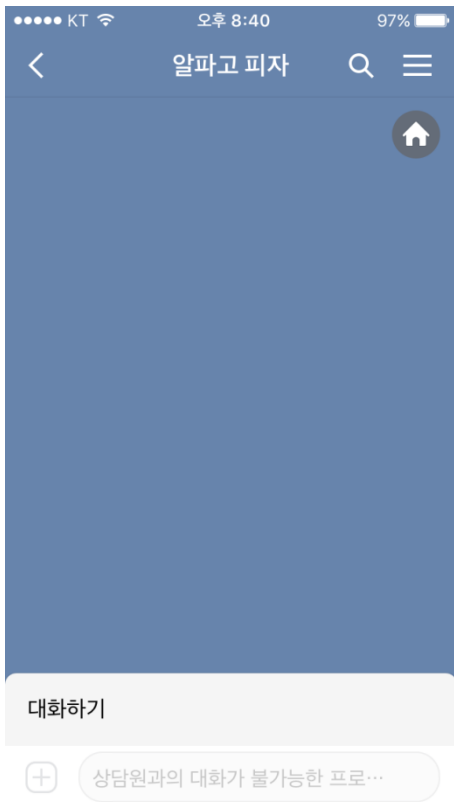
카카오톡과 연동



```
1
2 @app.route("/keyboard")
3 def keyboard():
4
5     res = {
6         "type": "buttons",
7         "buttons": [ "대화하기" ]
8     }
9
10    return jsonify(res)
11
```

키보드 처리

카카오톡 처음 접속시 요청. 메뉴 표시.



```

1
2 @app.route('/message', methods=['POST'])
3 def message():
4
5     # 메시지 받기
6     req = request.get_json()
7     user_key = req['user_key']
8     content = req['content']
9
10    if len(user_key) <= 0 or len(content) <= 0:
11        answer = ERROR_MESSAGE
12
13    # 답변 구함
14    answer = get_answer(content, user_key)
15
16    # 메시지 설정
17    res = {
18        'message': {'text': answer}
19    }
20
21    return jsonify(res)
22

```

메시지 처리

카카오톡에서 user_key와 content 받음. Dialogflow에서 대답을 구해 전송.

```

1
2 def get_answer(text, user_key):
3
4     # Dialogflow에 요청
5     data_send = {
6         'lang': 'ko',
7         'query': text,
8         'sessionId': user_key,
9         'timezone': 'Asia/Seoul'
10    }
11
12    data_header = {
13        'Content-Type': 'application/json; charset=utf-8',
14        'Authorization': 'Bearer adfb4242e4a0412f00111f0000000000'
15    }
16
17    dialogflow_url = 'https://api.dialogflow.com/v1/query?v=20150910'
18
19    res = requests.post(dialogflow_url,
20                        data=json.dumps(data_send),
21                        headers=data_header)
22

```

Dialogflow에 API 요청

Authorization에 설정 메뉴의 Client access token 입력


```
1
2 def get_answer(text, user_key):
3
4     #...
5
6     # 대답 처리
7     if res.status_code != requests.codes.ok:
8         return ERROR_MESSAGE
9
10    data_receive = res.json()
11    answer = data_receive[ 'result' ][ 'fulfillment' ][ 'speech' ]
12
13    return answer
14
```

Dialogflow에서 결과 받음

JSON 형식의 데이터를 파싱하여 대답 구함

```

1
2 @app.route( '/message', methods=[ 'POST' ])
3 def message():
4     # ...
5
6
7     #-----
8     # 메뉴 구함
9     #-----
10    answer, menu = get_menu(answer)
11
12    #-----
13    # 메뉴 버튼 설정
14    #-----
15    menu_button = get_menu_button(menu)
16
17    if menu_button != None:
18        res["keyboard"] = menu_button
19

```

키보드 메뉴 처리

카카오톡 형식에 맞게 변경

메뉴 보여줘

See how it works in [Google Assistant](#).

Agent

Domains

USER SAYS

COPY CURL

메뉴 보여줘

DEFAULT RESPONSE

피자 종류를 선택하세요 1. 불고기피자 2. 페퍼로니피자 3. 포테이토피자

CONTEXTS

RESET CONTEXTS

pizza_info_dialog_context

45eca394-80d3-4679-9df1-aeaa1e8a0fd8_id_dialog_context

KT

오후 5:04

100%

알파고 피자

알파고 피자입니다. 친구 추가 해 주셔서 감사합니다. 앞으로 다양한 소식과 혜택/정보를 메시지로 받으실 수 있습니다.

친구추가 일시(한국시간 기준): 2018년 07월 30일 16:56

수신거부: 홈>친구차단

오후 4:56

안녕

오후 5:03

알파고 피자

안녕!

오후 5:03

피자뭐있어?

오후 5:04

알파고 피자

피자 종류를 선택하세요

오후 5:04

불고기피자

페퍼로니피자

+

상담원과의 대화가 불가능한 프로...

```

1
2 @app.route( '/message', methods=[ 'POST' ])
3 def message():
4
5     # ...
6
7     #-----
8     # 사진 구함
9     #-----
10    answer, photo = get_photo(answer)
11    photo_width, photo_height = get_photo_size(photo)
12
13    #-----
14    # 사진 설정
15    #-----
16    if photo != '' and photo_width > 0 and photo_height > 0:
17        res[ 'message' ][ 'photo' ] = {
18            'url': photo,
19            'width': photo_width,
20            'height': photo_height
21        }
22

```

사진 처리

사진 크기 구함. 카카오톡 형식에 맞게 변경.

불고기피자가 뭐야?



See how it works in [Google Assistant](#).



Agent

Domains

USER SAYS

[COPY CURL](#)

불고기피자가 뭐야?



DEFAULT RESPONSE

<Photo><http://www.pizzamaru.co.kr/UpFile/...>
한국의 맛 불고기를 부드러운 치즈와 함께!

INTENT

[pizza_info](#)

ACTION

[pizza_info](#)

PARAMETER

VALUE

[pizza_type](#)

불고기피자



카카오톡 플러스친구 만들기

세상의 모든 비즈니스를 완성합니다.

카카오톡플러스친구



지금 무료로 만들어 새로운 비즈니스 홈을 경험하세요.

<https://center-pf.kakao.com> 접속

플러스친구 개설하기

카카오톡 친구들을 쉽게 만날수 있는 플러스친구를 만들어보세요. [?](#)

• 플러스친구 이름

20자 이내 (한글/영문/숫자)

• 검색용 아이디

15자 이내 (한글/영문 소문자/숫자), 변경불가

[노출화면 예시](#)

• 카테고리 1

카테고리 선택

카테고리 선택

카테고리 2

카테고리 선택

카테고리 선택

소개 메시지


플러스친구 홈에 노출될 소개 문구를 작성해주세요 (60자 이내)

프로필 사진



🔍 파일선택

- 권장 사이즈 640 x 640px
- jpg자원 (최대10MB)



알파고 피자

총 캐시 0원 >

메시지 이용권 0개 >

무료 발송 메시지 1,000건

다른 플러스친구 선택하기

홈

메시지 +

쿠폰 +

1:1채팅 +

스마트채팅

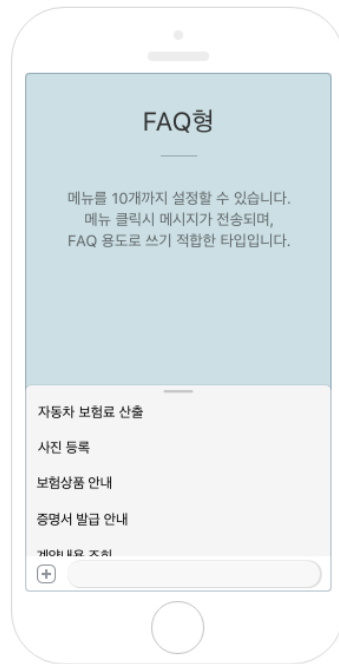
친구그룹 관리

홍보하기

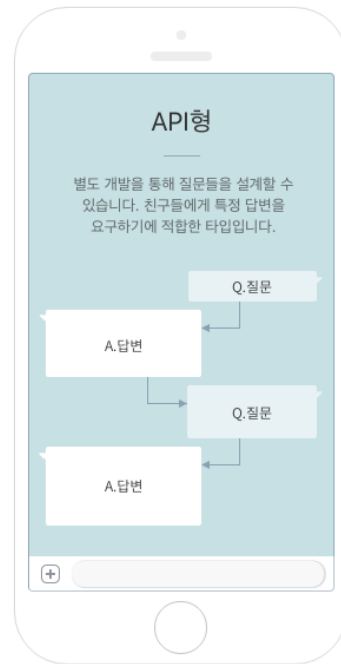
통계 +

관리 +

스마트채팅



설정하기



시작하기



알파고 피자

총 캐시 0원 >

메시지 이용권 0개 >

무료 발송 메시지 1,000건

다른 플러스친구 선택하기

홈

메시지 +

쿠폰 +

1:1채팅 +

스마트채팅

친구그룹 관리

API형

별도의 개발의 통해 특정 답변을 요구하는 형태의 질문들을 설계하는 타입입니다.

앱 등록

앱 이름 알파고 피자

앱 URL <https://dialogflow-kakao-chatbot.herokuapp.com>

API 테스트

앱 설명 알파고 피자

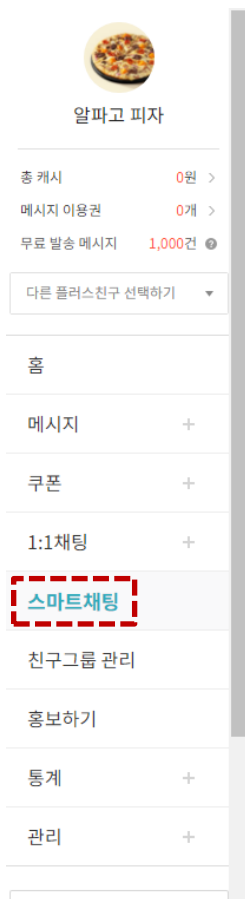
알림받을 전화번호

☒ 플러스친구 API의 개인정보 수집 및 이용에 동의합니다.

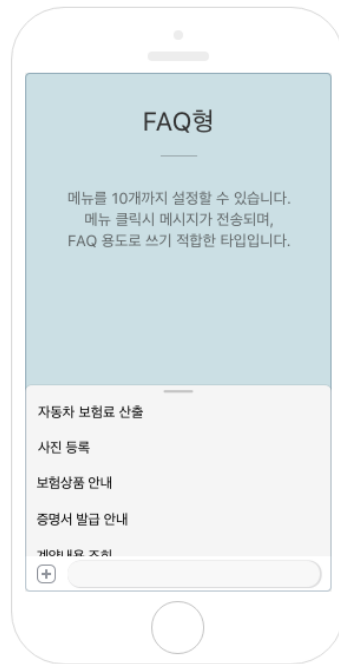
전화번호 대한민국(82)

카카오톡 이용중인 전화번호를 입력해주세요.

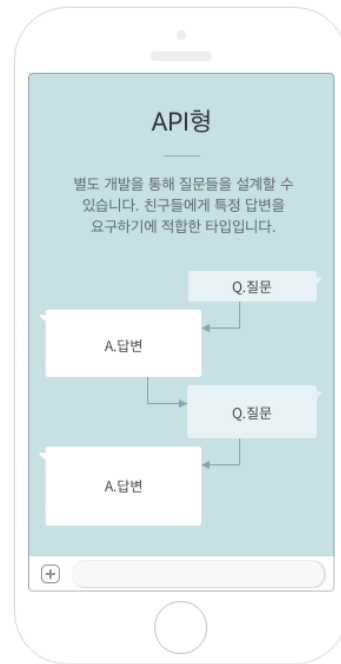
인증



스마트채팅



설정하기



시작하기

스마트채팅

친구그룹 관리

홍보하기

통계 +

관리 -

상세설정

관리자

캐시 관리

비즈정보 관리

히스토리

공지사항 >

고객센터 챗봇 바로가기 >

이용가이드 ↓

고객센터
1544-4293
(평일 10시 ~ 18시 30분)

플러스친구 홈

공개 설정 ?

홈 공개

ON

검색 허용

ON

국가설정

서비스 연결 ?

스토리채널

OFF

홈메뉴 설정 ?

기본

☐ 소식

☒ 정보

플러스친구 삭제

- 플러스친구 삭제는 마스터만 가능합니다.
- 마스터가 삭제를 요청하면 해당 플러스친구는 삭제 대기 상태로 전환되며 7일 후 완전 삭제됩니다.
- 삭제 대기 상태에서는 플러스친구가 비공개 처리되어 카카오톡에서 검색 및 홈 접근 등이 불가능합니다.
- 삭제 취소를 하더라도 원 상태로 복구되지 않는 정보를 꼭 확인 후 신중히 삭제를 진행해주세요.
- 플러스친구 관리자센터 탈퇴는 [내 계정정보](#)에서 하실 수 있습니다.

이제 마지막 작업

• pizza_order

SAVE



Training phrases ?

Search training phrases



” Add user expression

” 페퍼로니피자 먹을래

” 포테이토피자 구입하려고

” 불고기피자 보내줄래

” 페퍼로니피자 사고싶어

” 포테이토피자 주문할래

” 피자 먹을래

” 피자 보내줘

피자 주문 인텐트 추가

Traning phrases에 예시 문장들을 입력

• pizza_order

SAVE

Action and parameters ?

Enter action name

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?	PROMPTS ?
<input checked="" type="checkbox"/>	pizza_type	@pizza_type	\$pizza_type	<input type="checkbox"/>	피자 종류를 선택 하세요 1...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

+ New parameter

Action and parameters

예문에서 자동으로 엔터티 인식

• pizza_order

SAVE

Action and parameters ?

Enter action name

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?	PROMPTS ?
<input checked="" type="checkbox"/>	pizza_type	@pizza_type	\$pizza_type	<input type="checkbox"/>	피자 종류를 선택하세요 1...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

+ New parameter

REQUIRED 체크

슬롯 채우기. 문장에 엔터티가 없다면 PROMPTS 내용을 질문.

Prompts for "pizza_type" ×

NAME	ENTITY	VALUE
pizza_type	@pizza_type	\$pizza_type

PROMPTS

1

피자 종류를 선택하세요
1. 불고기피자
2. 페퍼로니피자
3. 포테이토피자

2

Enter a prompt variant

Close

PROMPTS

필요한 엔터티를 입력하도록 질문

• pizza_order

SAVE

Responses ?

DEFAULT +

Text response



1 주소를 입력하세요

2 Enter a text response variant



ADD RESPONSES

☐ Set this intent as end of conversation ?

Responses 설정

문장에 pizza_type 엔터티가 있으면 다음으로 주소 물어봄

연속된 대화를 하려면?

Search intents



 Default Fallback Intent

 Default Welcome Intent

 pizza_info

☐ pizza_order ▾

Add follow-up intent




Follow-up 인텐트 추가

피자 주문 후 이어서 주소 입력을 하도록 연결된 대화 흐름 작성


Search intents



 Default Fallback Intent

 Default Welcome Intent

 pizza_info

 pizza_order ▾

custom

fallback

yes

Custom 형식 선택

- pizza_order

SAVE



Contexts ?



Add input context

1

pizza_order-followup



Add output context



pizza_order 인텐트에 자동으로 output context 생성됨

output context는 현재 인텐트의 정보를 이후 인텐트로 전달. 1턴만 유지하도록 변경.

Intents

CREATE INTENT




Search intents





 Default Fallback Intent

 Default Welcome Intent

 pizza_info

 pizza_order ^

  pizza_order-followup

pizza_order와 연결된 새로운 인텐트

- pizza_order-followup

SAVE



Contexts ?



pizza_order-followup (X) Add input context

Add output context



새로운 인텐트에 자동으로 input context 생성됨

pizza_order-followup context가 존재해야 현재 인텐트를 실행

• pizza_order-followup

SAVE

Training phrases ?

Search training phrases



” Add user expression

” 주소

Filter

Action a @sys.unit-information

pizza_o @sys.given-name

@sys.any

REQUIRED

@sys.currency-name

VALUE

IS LIST ?

Training phrases에 아무 글자나 넣고 마우스 드래그

@sys.any 선택. 입력된 문장을 그대로 사용.

- pizza_order-followup

SAVE



Training phrases ?

Search training phrases



” Add user expression

” 주소

PARAMETER NAME	ENTITY	RESOLVED VALUE	
address	@sys.any	주소	x

PARAMETER NAME에 address 입력

- pizza_order-followup

SAVE

Action and parameters ?

pizza_order

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?
<input type="checkbox"/>	pizza_type	Enter entity	#pizza_order-followup.pizza_type	<input type="checkbox"/>
<input type="checkbox"/>	address	@sys.any	\$address	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

pizza_type 파라미터 추가

input context에서 받은 이전 인텐트 정보 사용. pizza_order 액션 이름 입력.

- pizza_order-followup

SAVE



Fulfillment ?



Enable webhook call for this intent



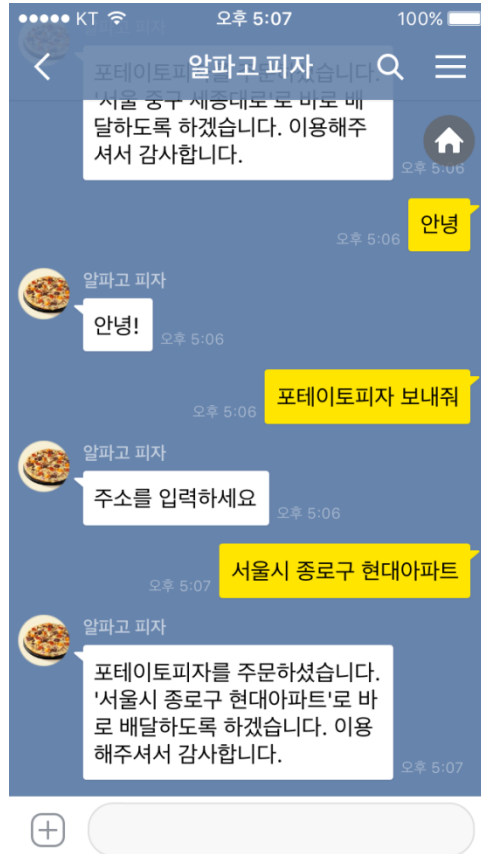
Enable webhook call for slot filling

Fulfillment 체크

Webhook에 피자 주문 추가

```
1
2 @app.route('/', methods=[ 'POST' ])
3 def webhook():
4
5     # 액션 구함
6     req = request.get_json(force=True)
7     action = req[ 'result' ][ 'action' ]
8
9     # 액션 처리
10    if action == 'pizza_info':
11        pizza_name = req[ 'result' ][ 'parameters' ][ 'pizza_type' ]
12        answer = process_pizza_info(pizza_name)
13    elif action == 'pizza_order':
14        pizza_name = req[ 'result' ][ 'parameters' ][ 'pizza_type' ]
15        address = req[ 'result' ][ 'parameters' ][ 'address' ]
16        answer = process_pizza_order(pizza_name, address)
17    else:
18        answer = 'error'
19
20    res = { 'speech': answer }
21
22    return jsonify(res)
23
```

```
1
2 def process_pizza_order(pizza_name, address):
3
4     answer = pizza_name.encode('UTF-8') + '를 주문하셨습니다.'
5     answer += " '" + address.encode('UTF-8') + "'로 바로 배달하도록 하겠습니다."
6     answer += ' 이용해주셔서 감사합니다.'
7
8     return answer
9
```



새로운 피자 추가해보기

1. 엔터티 추가
2. 인텐트 수정
3. 웹훅 코드 수정
4. Heroku 업로드

감사합니다



궁금한 점이 있으시면 이곳으로...