

# NUS-ISS Chatbot User Guide

31-May-2019 | Version 1.0

---

**Team Name:** ai.Orz

Dai Yirui, Dong Meirong, Gu Lijian, Guo Feng, Wong Yoke Keong, Zhang Le

Prepared for NUS-ISS Master of Technology in Intelligent Systems ISY5001 Cognitive Systems Group Project

# Table of Contents

	1
Table of Contents	2
1.0 System Overview	3
2.0 Requirements	3
2.1 Recommended Browsers	3
2.2 Dependencies	3
3.0 Deployment	4
1. Prerequisites	4
2. Start Dialogflow	4
3. Start backend script	5
4. Start the ngrok tunnel	7
4.0 Walkthrough	8
5.0 Business Scenarios - Sample Input and System Output	9
5.1 Scenario 1	9
5.2 Scenario 2	10
5.3 Scenario 3	11

## 1.0 System Overview

In this project, we aim to design and train a chatbot based on Google powered Dialogflow to automatically handle users' general enquiries about Institute of System Sciences (ISS) at National University of Singapore (NUS). Some more specific queries will be covered by webhook API. The main objectives are to increase the efficiency of information retrieval and to provide more human-like conversational experiences.

## 2.0 Requirements

This project uses Dialogflow, a set of trained contexts and an external API that retrieves data from JSON format file to facilitate conversational flow.

### 2.1 Recommended Browsers

ISS chatbot demonstration supports the following web browsers:

- Google Chrome Version 74 and above

### 2.2 Dependencies

- Dialogflow
- Python 3.7
- Python framework: Flask
- Python library: requests, json
- ngrok

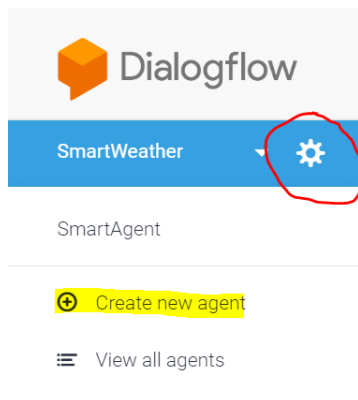
## 3.0 Deployment

### 1. Prerequisites

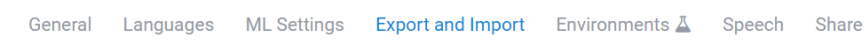
- Got to Google Dialogflow and create an account. Link: <https://console.dialogflow.com/api-client/#/login>
- Download project source code from github repository: <https://github.com/ddgone2far/IRS-CS-2019-05-04-IS1PT-GRP-ai.Orz-ISSChatBot>
- Download ngrok.exe or ngrok binary to your machine from ngrok.com
- Install python 3.7, Flask, requests and json

### 2. Start Dialogflow

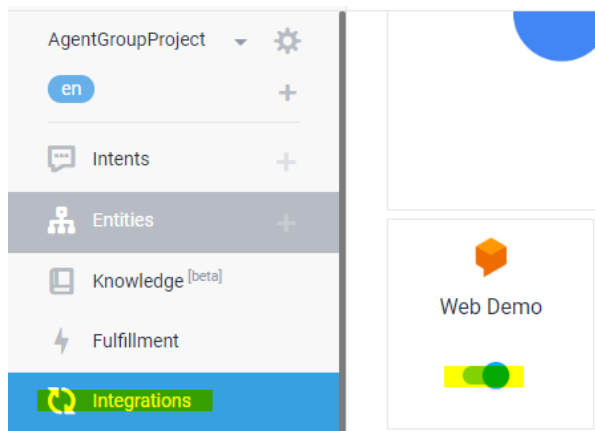
- Create a new agent called "ISS Chatbot" (yellow section)



- Click the button in red circle
- Import the zip file "AgentGroupProject.zip" downloaded



- Check and enable the Dialogflow agent integration.



- Check and copy the integration URL

```
<iframe
  allow="microphone;"
  width="350"
  height="430"
  src="https://console.dialogflow.com/api-client/demo/embedded/71fd30ed-1ffd-4c7a-8869-b90896f51143"
>
</iframe>
```

### 3. Start backend script

The backend scripts are in the folder "ISSChatbot".

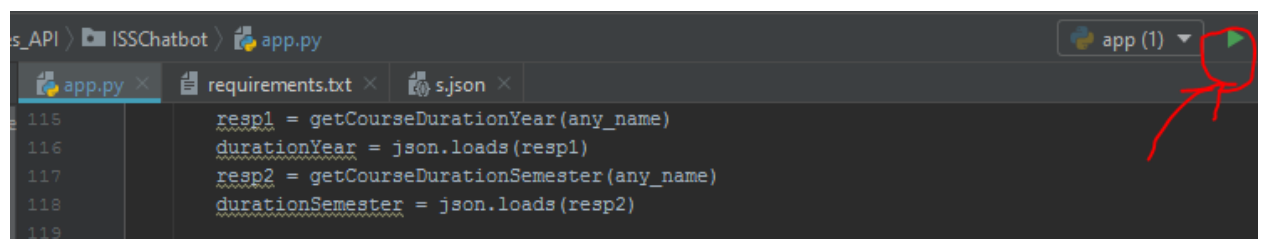
Replace the iframe Dialogflow integration URL in /ISSChatbot/templates/index.html with above Agent ID.



```
1 <!doctype html>
2 <html lang="en">
3 <title>AI.orz</title>
4 <style>
5   .resp-iframe { position: absolute; top: 0; left: 0; width: 100%; height: 100%; border: 0; }
6 </style>
7 <iframe class="resp-iframe"
8   allow="microphone;"
9   src="https://console.dialogflow.com/api-client/demo/embedded/71fd30ed-1ffd-4c7a-8869-b90896f51143"
10 >
```

If PyCharm is used:

- Follow the PyCharm instruction to install required libraries (by clicking the recommended install button)
- Click the Run Button (or right click on the app.py and select run)



```
115 resp1 = getCourseDurationYear(any_name)
116 durationYear = json.loads(resp1)
117 resp2 = getCourseDurationSemester(any_name)
118 durationSemester = json.loads(resp2)
119
```

```
app (1) x
D:\pycharm\py-workspace\venv\Scripts\python.exe D:/pycharm/py-workspace/AgentZhangLe_Programmes_API/ISSChatbot/app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 117-035-302
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

If cmd is used:

- cd /ISSChatbot (Go to the app.py and requirements.txt directory)
- pip install -r requirements.txt (Install dependency and library) python app.py

```
C:\> python app.py
D:\pycharm\py-workspace\AgentZhangLe_Programmes_API\ISSChatbot>python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 102-801-000
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

## 4. Start the ngrok tunnel

- Run ngrok tunnel
- For Mac/Linux: ngrok http 5000
- For Windows: ngrok.exe http 5000
- Copy the https....\*.ngrok.io URL and paste it in Dialogflow fulfillment section of your agent
- Under “Fulfillment” >> Enable “Webhook”
- Click save

Screenshot from ngrok

```

ngrok by @inconsthreveable (Ctrl)

Session Status      online
Session Expires     6 hours, 31 minutes
Version             2.2.8
Region             United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://67d4e01e.ngrok.io -> localhost:5000
Forwarding           https://67d4e01e.ngrok.io -> localhost:5000

Connections         ttl    opn    rt1    rt5    p50    p90
                   12     0      0.00   0.00   0.03   0.45

HTTP Requests
-----

```

Screenshot from Dialogflow

### ⚡ Fulfillment

#### Webhook

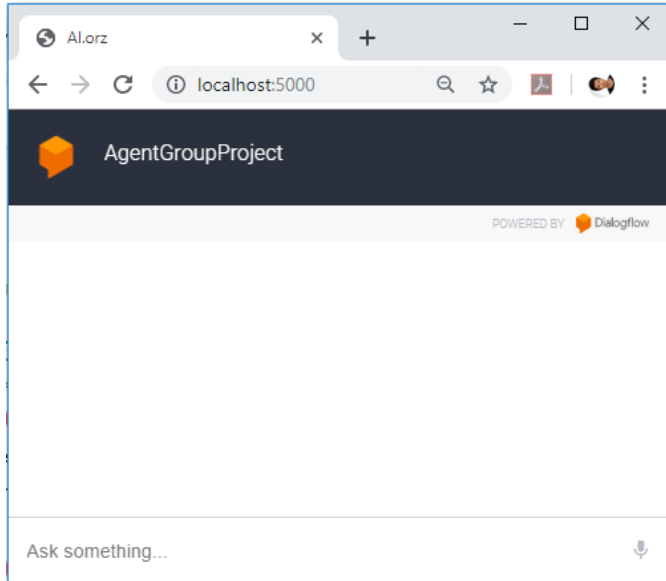
ENABLED ☒

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.

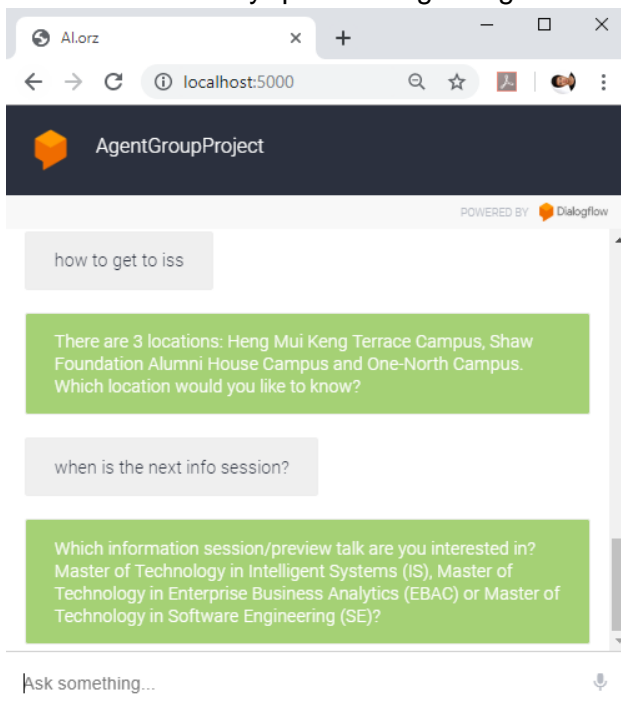
URL*	<input type="text" value="https://67d4e01e.ngrok.io"/>	
BASIC AUTH	<input type="text" value="Enter username"/>	<input type="text" value="Enter password"/>
HEADERS	<input type="text" value="Enter key"/>	<input type="text" value="Enter value"/>
	<input type="text" value="Enter key"/>	<input type="text" value="Enter value"/>

## 4.0 Walkthrough

Open chrome and go to <http://localhost:5000/> (or you can access the same service anywhere via url exposed by ngrok)



Feel free to ask any question regarding NUS ISS (e.g. Programs, Courses etc)



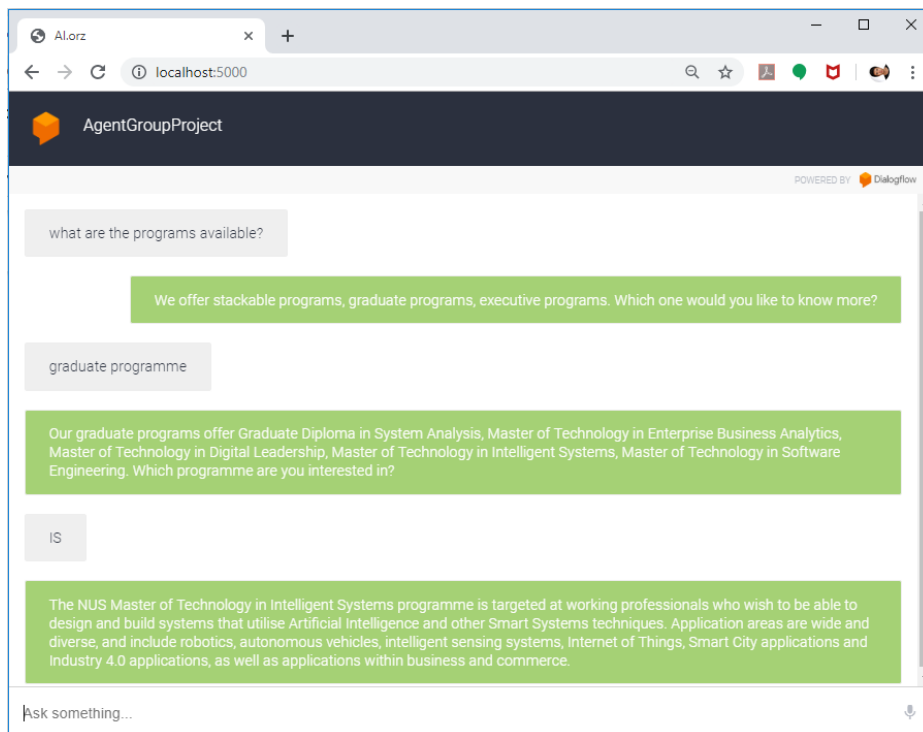


## 5.0 Business Scenarios - Sample Input and System Output

### 5.1 Scenario 1

<p><b>SCENARIO 1</b> User asks about the programs information of NUS ISS.</p>	<p>Intents:</p> <ul style="list-style-type: none"> <li>iss.programs <ul style="list-style-type: none"> <li>iss.programs.graduate <ul style="list-style-type: none"> <li>iss.programs.graduate.DL</li> <li>iss.programs.graduate.GDipSA</li> <li>iss.programs.graduate.EBAC</li> <li>iss.programs.graduate - fallback</li> </ul> </li> <li>Contexts: <ul style="list-style-type: none"> <li>issprogramsgraduate-followup</li> <li>iss.programs.graduate.IS</li> <li>iss.programs.graduate.SE</li> </ul> </li> </ul> </li> <li>iss.programs.executive</li> <li>iss.programs.stackable</li> <li>iss.programs - no</li> </ul>
---	---

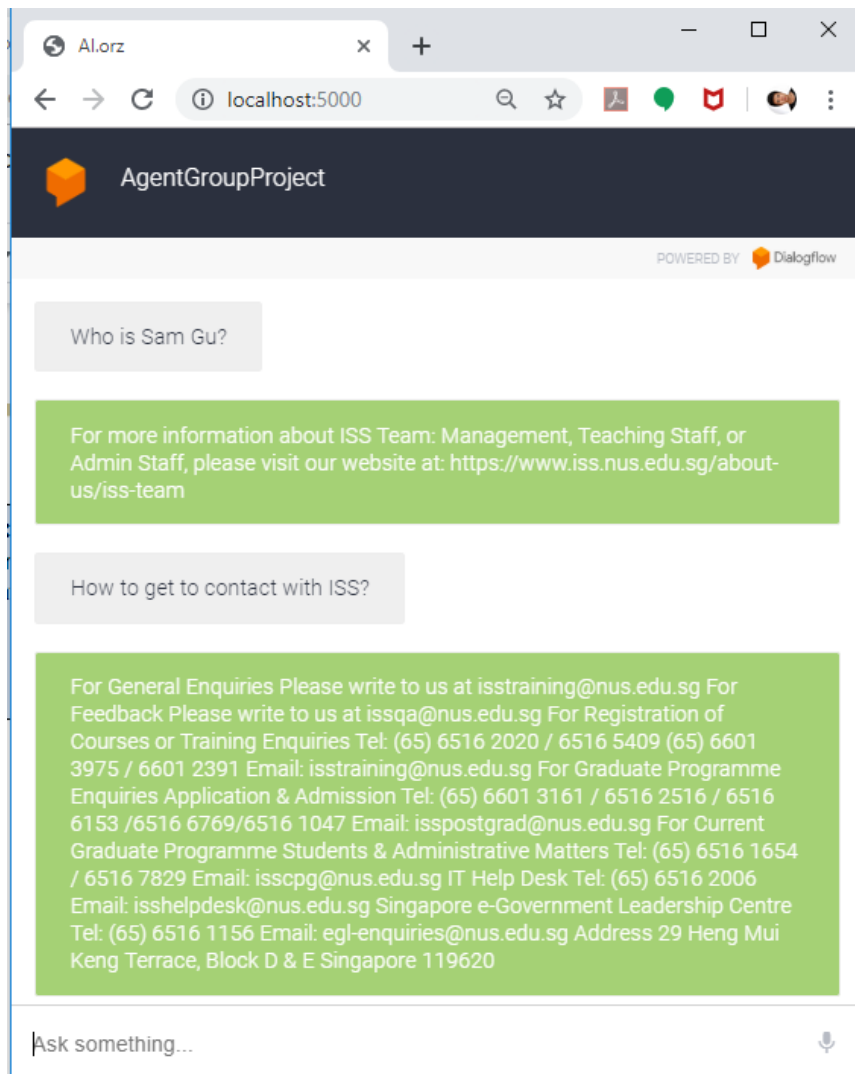
Sample Screenshot:



## 5.2 Scenario 2

<p><b>SCENARIO 2</b> User asks about a specific name (Could be teaching or management staff), or general ISS contacts.</p>	<p>Intents:</p> <ul style="list-style-type: none"> <li>• lss.people</li> <li>• iss.contactInfo</li> </ul>
--	---

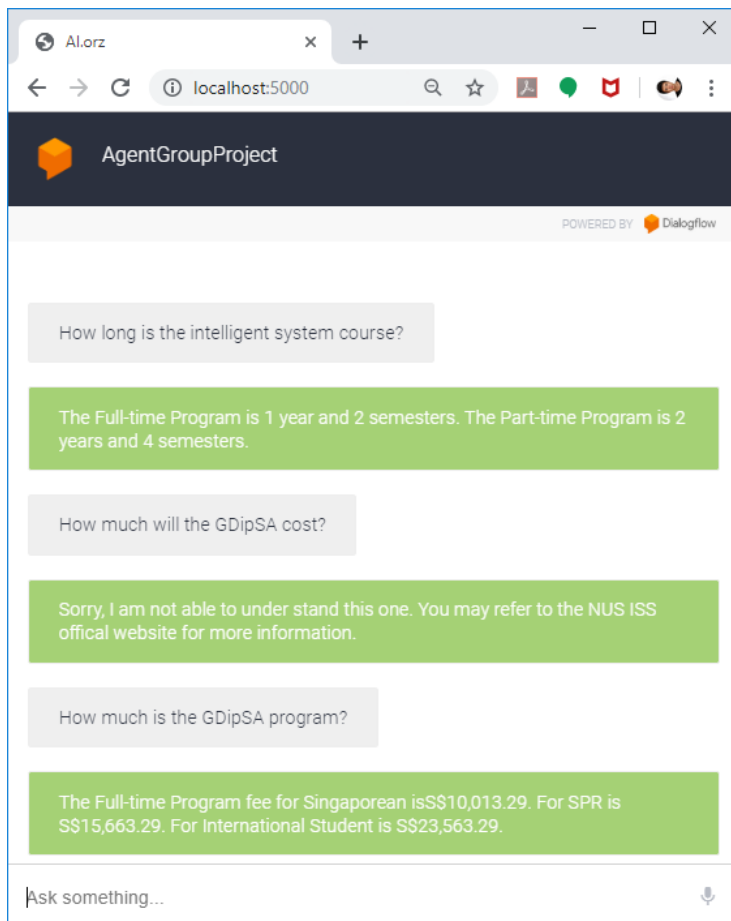
### Sample Screenshot



## 5.3 Scenario 3

<p><b>SCENARIO 3</b> User asks about details (duration, fees) of a specific course.</p>	<p>Intents:</p> <ul style="list-style-type: none"> <li>• ISSCourseDuration</li> <li>• ISSCourseFee</li> <li>• ISSCourseIntent</li> </ul>
---	--

Sample Screenshot:



Backend system print out after receiving HTTP request:

```
app (1) x
[31/May/2019 22:26:17] "POST / HTTP/1.1" 200 -
{"responseId": "48b71f09-5c5e-42c2-889b-a006f0b5b7a0-e6604cc1", "queryResult": {"queryText": "How long is the intelligent system course?", "parameters": {"courseName": "intel
{"responseId": "26911df0-d93d-4486-b13a-3ec94bdc5abf-e6604cc1", "queryResult": {"queryText": "How much will the GDipSA cost?", "parameters": {"courseName": "How much will the
127.0.0.1 - - [31/May/2019 22:26:17] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [31/May/2019 22:27:09] "POST / HTTP/1.1" 200 -
{"responseId": "00cc4ab9-183c-4cd6-8d8d-661adaffce01-e6604cc1", "queryResult": {"queryText": "How much is the GDipSA program?", "parameters": {"courseName": "GDipSA"}, "allRe
```