ShopBot

for earphones and headphones

User Guide



|  |  |
| --- | --- |
| **Team Members** | |
| Shashank Nigam | A0198469A |
| Lin Wenqi | A0198435R |
| Ng Mei Ying | A0198546L |
| Edmund Leow Kwong Wei  **TEAM MEMBERS**  Edmund Leow Kwong Wei  Ng Mei Ying  Wilson Lum Kok Keong A0198478A  M A S T E R O F TECH N O L O G Y  P R O J E C T R E P O RT    T E A M M E M B E R S  Edmund Leow Kwong Wei - 1  Ng Mei Ying - 2  Wilson Lum Kok Keong - 3  M A S T E R O F TECH N O L O G Y  P R O J E C T R E P O RT    T E A M M E M B E R S  Edmund Leow Kwong Wei - 1  Ng Mei Ying - 2  Wilson Lum Kok Keong - 3 | A0198458H |

Contents

[**Getting Started** 1](#_Toc36378179)

[**Full installation and deployment** 1](#_Toc36378180)

[System Requirements and Dependencies 1](#_Toc36378181)

[Pre-requisites 1](#_Toc36378182)

[Importing ShopBot into DialogFlow 1](#_Toc36378183)

[Connecting Kommunicate with DialogFlow 2](#_Toc36378184)

[Starting Rasa NLU server 3](#_Toc36378185)

[Starting Python Flask for Webhooks 3](#_Toc36378186)

[Launching the Website 5](#_Toc36378187)

[Troubleshooting 6](#_Toc36378188)

[**Test Scenarios** 7](#_Toc36378189)

[Welcome 7](#_Toc36378190)

[Get Product Recommendations 7](#_Toc36378191)

[Get Pricing and Buying information 8](#_Toc36378192)

[Get answers to headphone-related stuff (FAQ) 9](#_Toc36378193)

# **Getting Started**

## **Full installation and deployment**

Follow this section if installing and deploying to your own DialogFlow account and local web server.

### System Requirements and Dependencies

* Python or Anaconda, and Python libraries as specified in requirements.txt
* A modern web browser. Recommended Google Chrome version 76 and above.

### Pre-requisites

* Google DialogFlow account. Sign up at <https://console.dialogflow.com/>
* Kommunicate account. Sign up at <https://www.kommunicate.io/>
* Download ngrok.exe or ngrok binary to your machine from <https://ngrok.com/download>
* Clone or download project source code from GitHub repository from <https://github.com/eleow/shopBot>, navigate to the root folder of the repo, and install python project dependencies
* In a new Anaconda environment (eg shopbot) with **Python 3.6**,

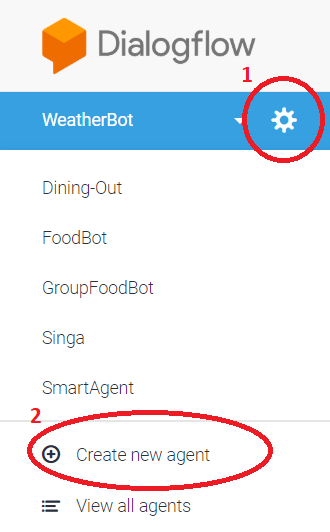
pip install -r requirements.txt

* Alternatively, use the script below. This will automatically create a new environment shopbot and install all dependencies (Note: Tested only on Windows 10 version 1909)

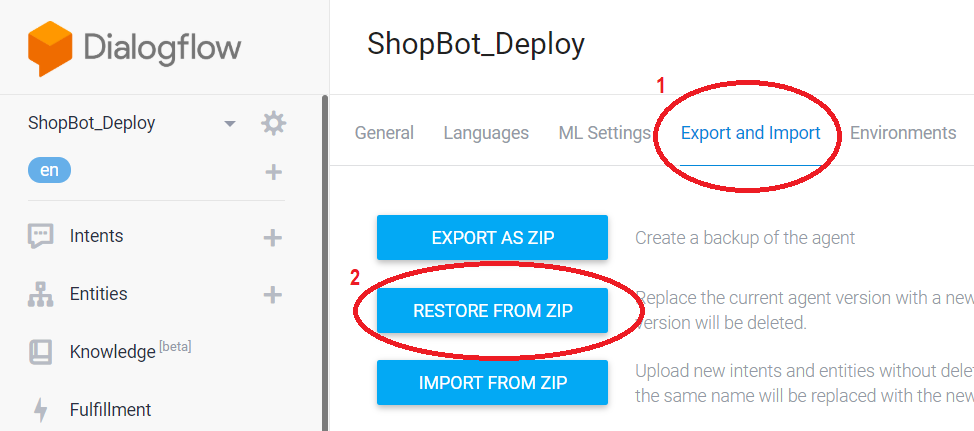
conda env create -f environment.yml

### Importing ShopBot into DialogFlow

* Login to DialogFlow console at <https://console.dialogflow.com/>
* Create a new agent called “ShopBot” or whatever name you desire by
  + Clicking the cog button
  + Clicking the “Create new agent” button

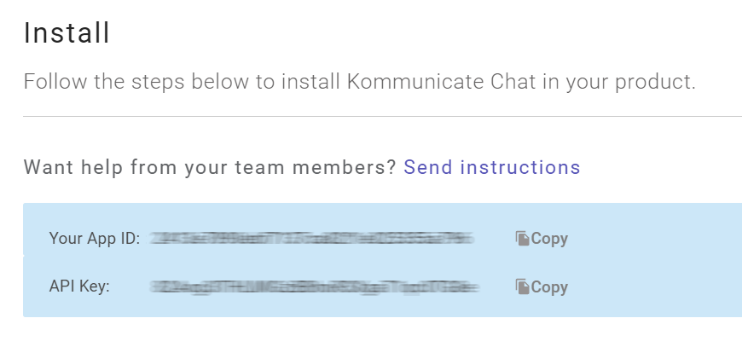


* Import the zip file (ShopBot\_Deploy.zip) from the local copy of the GitHub repository at \SystemCode\DialogFlow by clicking the “Export and Import” button on DialogFlow, and then clicking “**RESTORE FROM ZIP**”

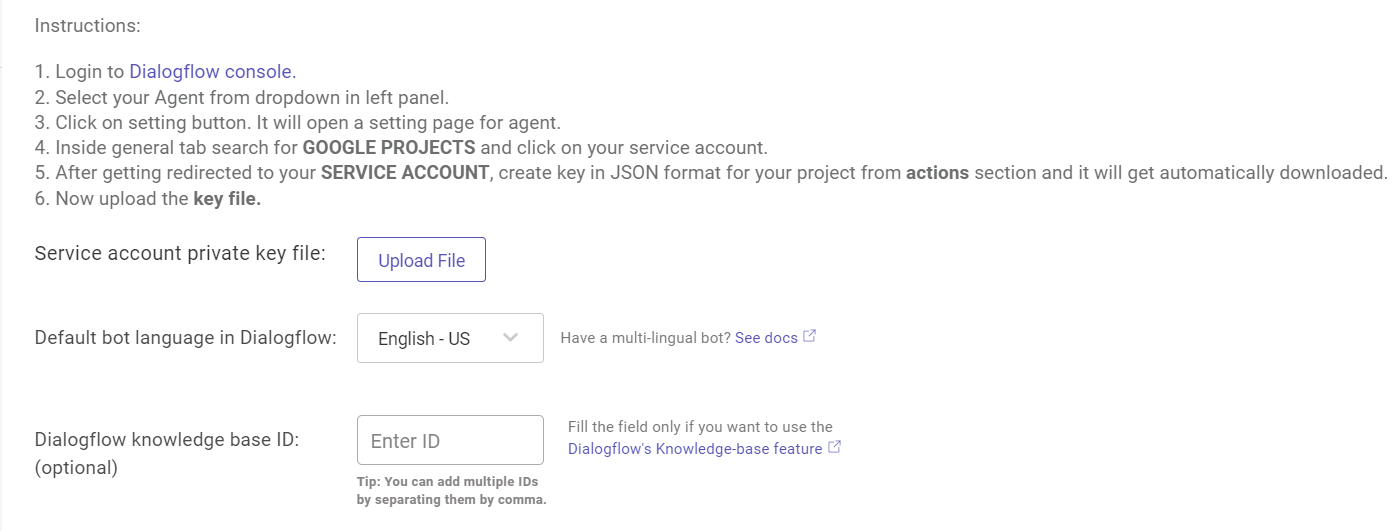


### Connecting Kommunicate with DialogFlow

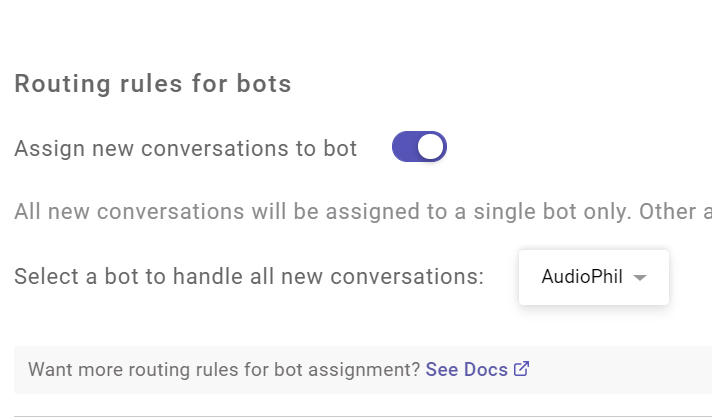
* Login to [Kommunicate](https://www.kommunicate.io/)
* In [Install Page](https://dashboard.kommunicate.io/settings/install), note down your **App ID**. This will be used later



* Integrate Bot with DialogFlow in [Bot Integration Page](https://dashboard.kommunicate.io/bots/bot-integrations), by following the instructions below



* Important: Go to [Conversation Rules page](https://dashboard.kommunicate.io/settings/conversation-rules) (Settings 🡪 Conversation Rules 🡪 Routing rules for bots), and enable “Assign new conversations to bot. Select the bot that you have created.

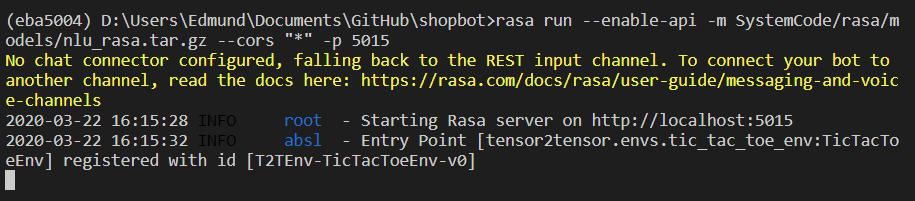


### Starting Rasa NLU server

* Navigate to the root of your local copy of the GitHub repository eg D:\github\shopBot
* Run Rasa NLU server by the following script in console:

rasa run --enable-api -m SystemCode/rasa/models/nlu\_rasa.tar.gz --cors "\*" -p 5015

* It might take some time to start up. But finally, you should see the following in the console

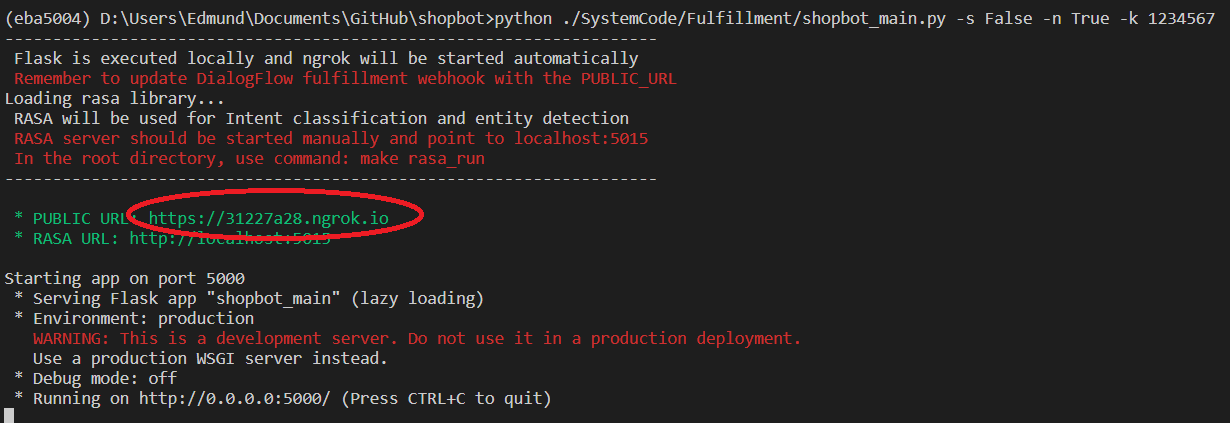


### Starting Python Flask for Webhooks

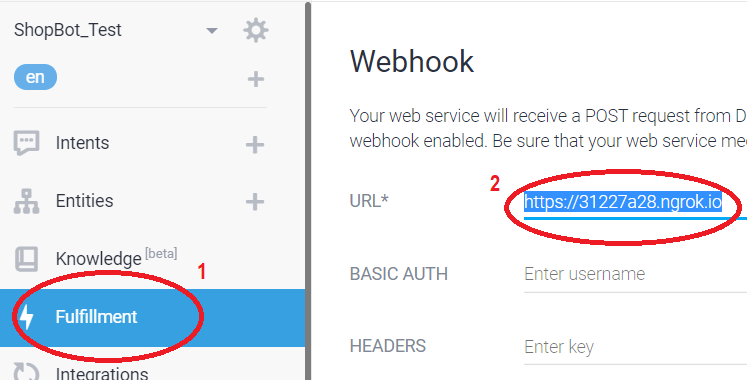
* Navigate to the root of your local copy of the GitHub repository eg D:\github\shopBot
* Run shopbot\_main.py by the following script in console, using the **Kommunicate AppID** from earlier instead of <AppID>:

python ./SystemCode/Fulfillment/shopbot\_main.py -s False -n True -k <AppID>

* It might take some time to start up. But finally, you should see the following in the console



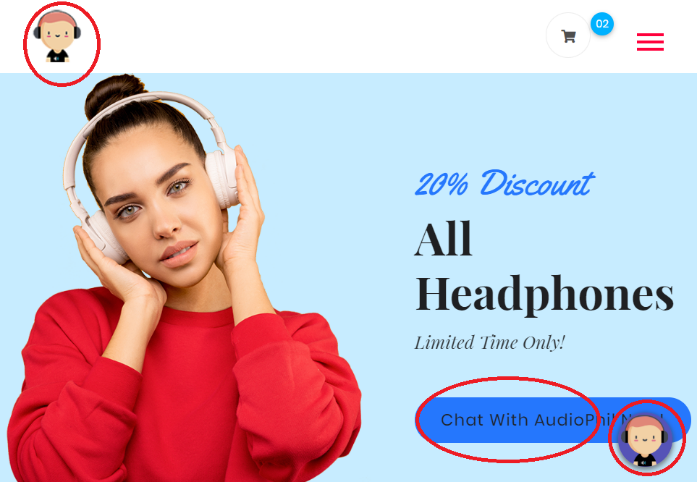
* Backend server is now running, and ngrok has been automatically started as well. Note down the public url for ngrok (Public url will change everytime you restart the backend)
* Now, we need to update DialogFlow Webhook URL with this. In DialogFlow, navigate to **Fulfilment** and configure the webhook to point to the URL that was captured earlier, and then **remember to click Save.**

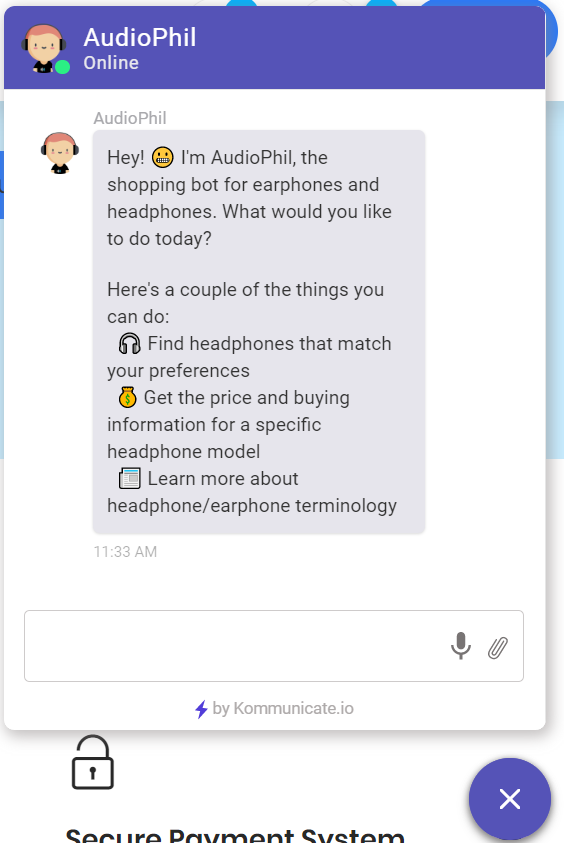


* The ShopBot is now running on DialogFlow and is connected to the local Python Flask web server through ngrok, and Kommunicate is connected to DialogFlow.

### Launching the Website

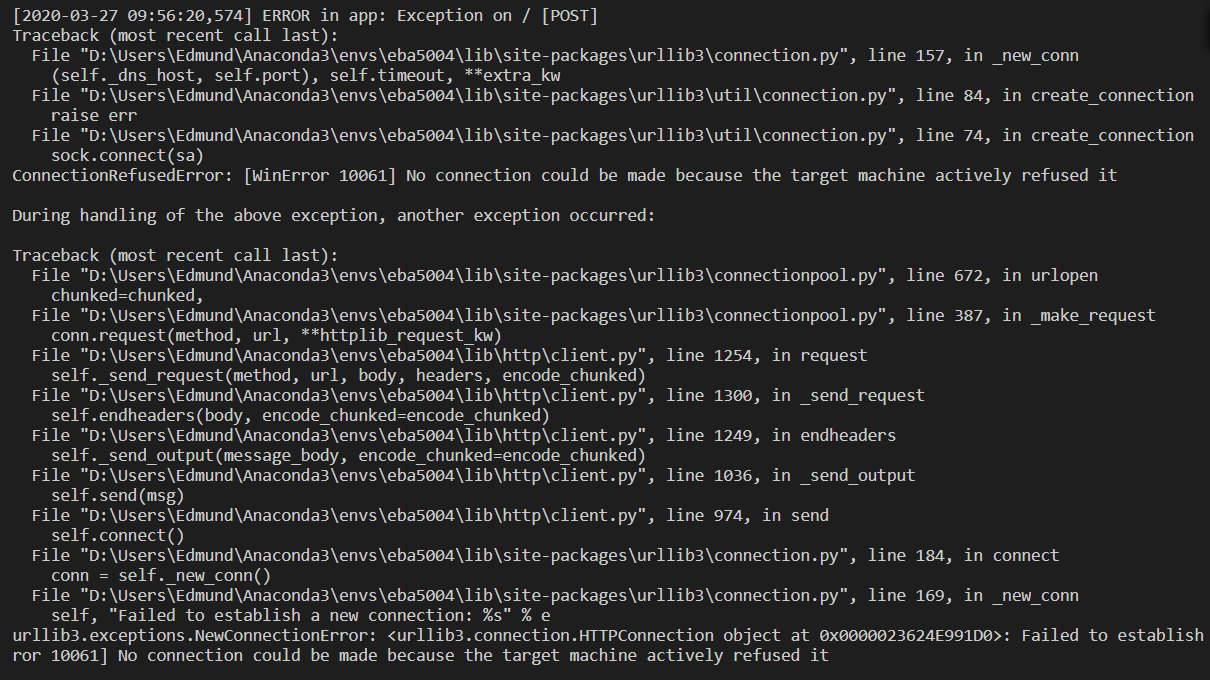
* Ensure that (i) ngrok, (ii) rasa NLU server, (iii) flask webhook are running from the steps above.
* Finally, you can launch the website using PUBLIC\_URL or using <http://localhost:5000>
* Click on the AudioPhil icon  at the bottom right of the page (or any of the other items circled in red below), and start chatting!
* Note that the rest of the website will NOT work as this is just a proof-of-concept of how the chatbot could be integrated into an actual e-commerce website selling headphones.





## **Troubleshooting**

### [WinError 10061] No connection could be made because the target machine actively refused it



Looks like you might have forgotten to run Rasa NLU server! This error can occur if Rasa NLU server is not running or is running on a different port from what is configured. Refer to Starting Rasa NLU server

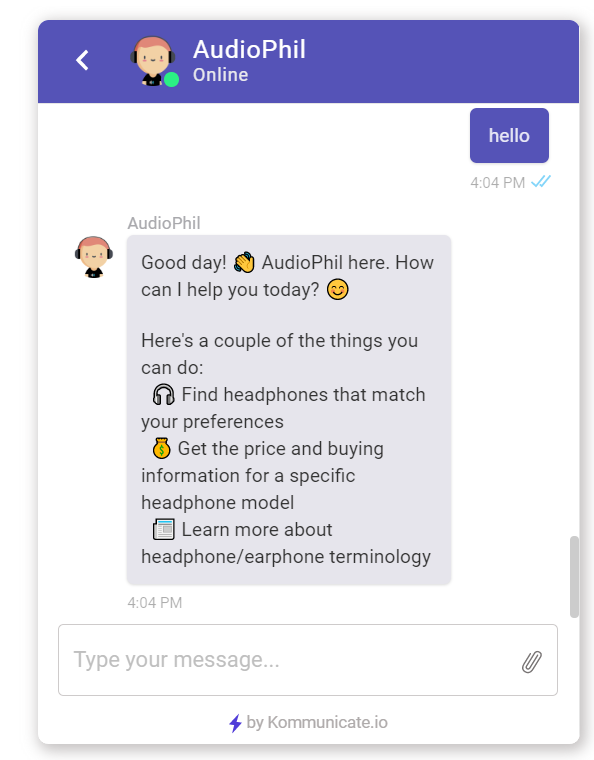
## **Test Scenarios**

The following intents are available

* Welcome
* Get product recommendations
* Get pricing and buying information for a specific headphone model
* Get answers to headphone-related stuff (FAQ)

### Welcome

On launching AudioPhil, the chatbot will automatically welcome you.



### Get Product Recommendations

If the user asks for product recommendation, we will return products based on his preferences.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample User Query** | **Sample Response** | **Entities** | **Description** |
| “can you find sony headphones?” |  | product\_type= [“headphones”]  ent\_brand=["sony"] | Product type and brand was entered by user and chatbot asked for connectivity type |
| “I want both” |  | wired="both" | Connectivity type was entered by user and chatbot asked for confirmation of selection |
| “that’s right” |  |  | Results based on user’s preferences were returned, as a carousel |

### Get Pricing and Buying information

If the user provides a specific headphone model, we will provide the product information (price, features and link to buy it), if it is found.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample User Query** | **Sample Response** | **Entities** | **Description** |
| “what is the price of AKG 452?” |  | brand= "AKG"  model="452" | Product brand and model were provided by the user, and result was returned. Product was either found only in Treoo or was cheaper in Treoo, therefore, the product price and link were given from Treoo. |
| “price xg13 mini” |  | brand=""  model= "xg13" | Same as above, but this time, only product model was provided. And product was found to be cheaper from Amazon SG |
| “hey details of air” |  | brand=""  model="air" | In this case, there were multiple brands that had this particular model. Therefore, we asked the user for the brand. Since the possible brands are not many (<5), we showed suggestion chips. |
|  | brand="crazybaby"  model="air" | Then the user selected “crazybaby”, and results were returned |
| “info on sony” |  |  | In this case, user only entered a brand. Since possible models are too many (>5), we did not show any suggestion chips |

### Get answers to headphone-related stuff (FAQ)

You can ask the chatbot specific terms related to headphones. We will provide an explanation of the term, as well as the source of the information.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample User Query** | **Sample Response** | **Entities** | **Description** |
| “what is Bluetooth” |  | query="bluetooth" | Search term was found in our list of glossary. Description and source of information is provided to the user |
| “what is alkjopuwer” |  | query=”alkjopuwer” | In this case, search term was not found in our list. Therefore, we tell the user that we do not know what it is. |
| “what is power” |  | query=”power” | In this case, while an exact match was not found, we managed to match it to the closest term “POWER SUPPLY”, by comparing word vector similarity. |