Steps overview:

Part A: Prepare Google Sheet

Part B: Set Up Google Service Account

Part C: Install reference codes in Pythonanywhere

Part D: Configuring Flask Application Webhook in Pythonanywhere

Part E: Importing Dialogflow Agent

*Last updated: 17 May 2022*

Part A: Setting Up Google Sheet

|  |  |
| --- | --- |
| Google Sheet can be used a data source for a Dialogflow agent. It can also be used to store information that was collected by an agent. Within the same document, we can create multiple sheets (tabs), with each sheet containing one particular category of information. | |
| 1. | Visit the link below to access the Google Sheet Template.  <https://docs.google.com/spreadsheets/d/1rkssSClbDrNjQY1Qavot2nLtJ43Ge1ndej-rJ1EEd_4/copy?usp=sharing>  or  <https://tinyurl.com/gsdyfc>  You might be required to sign into your Google account.  Request permission to access the document. |
| 2. | Make a copy of the Google Sheet into your own Google Drive account.    Rename the sheet as **DYFC-GSheet-Backend**  *Note: The name of the sheet is very important. The webhook program assumes that this is the name of the spreadsheet.* |

Part B: Creating a Service Account

|  |  |
| --- | --- |
| A service account is a special type of Google account intended to represent a non-human user that needs to authenticate and be authorized to access data in Google APIs . Since it’s a separate account, by default it does not have access to any spreadsheet until you share it with this account. Just like any other Google account. | |
| 1. | Goto Google Cloud console at <https://console.cloud.google.com> |
| 2. | From the top bar, click the triangle next the Google Cloud Platform.    In the pop up screen, click **NEW PROJECT** |
| 3. | In the pop up screen, replace the **project name** with something that is more descriptive.    Click **CREATE.** The console navigates to the Dashboard page and your project is created within a few minutes. |
| 4. | From the top bar, click the **triangle icon** (located next to the Google Cloud Platform)  Select the project name that you have created. |
| 5. | In the **box** labelled “Search for APIs and Services”, search for “**Google Drive API**”.  Click **ENABLE.** |
| 6. | Repeat Step 5, search for **Google Sheet API** and enable the service.  In the box labelled “Search for APIs and Services”, search for “**Google Sheets API”** and enable it. |
| 7. | Open the side navigation panel. Go to “**APIs & Services > Credentials**” .    Choose “+**CREATE CREDENTIALS > Service account key**”.    Fill out the form as shown. Click “**CREATE** ” and “**DONE**”.    You should return to the dashboard page. |
| 8. | Press “**Manage service accounts**” on the right side of Service Accounts section.    Press on **⋮** near recently created service account and select “**Manage keys”**    Click on “**ADD KEY > Create new key**”.    Select **JSON key type** and press “**CREATE**’.    You will automatically download a JSON file with credentials. It may look like this:    Remember the path to the downloaded credentials file. Also, in the next step you’ll need the value of client\_email from this file. |
| 9. | Very important! Go to your spreadsheet and share it (editor rights) with a client\_email from the step above. Just like you do with any other Google account.    If you don’t do this, you’ll get a gspread.exceptions.SpreadsheetNotFound exception when trying to access this spreadsheet from your application or a script. |
| 10. | Rename the credential JSON file to **secret-key-file.json** .  Make sure you store the credentials file in a safe place. You will need it in a later step. |

Part C: Setting Up Pythonanywhere.com

|  |  |
| --- | --- |
| PythonAnywhere is an online integrated development environment (IDE) and web hosting service (Platform as a service) based on the Python programming language. it provides in-browser access to server-based Python and Bash command-line interfaces, along with a code editor with syntax highlighting. Program files can be transferred to and from the service using the user's browser. | |
| 1. | Create an account with <https://www.pythonanywhere.com>. Follow the onboarding instructions. |
| 2. | Login to Pythonanywhere.com |
| 3. | From the navigator bar, select **Consoles** .  Then Start a new console by choosing **Bash** . |
| 4. | a. Use the Bash Terminal console to create a new directory. Type the command below.  $ mkdir DYFC    b. Change to the new directory  $ cd DYFC    c. Clone the repository .  $ git clone <https://github.com/Ipomoeabatatas/dyfc-webhook.git> .  *Note that this process will take a while to complete.*    A new sub-directory (d**yfc-webhook**) will be created. Change directory to this new sub-directory.  $ cd dyfc-webhook |
| 5. | Use the Bash Terminal console to create a new virtual environment (dyfc).  $ mkvirtualenv --python=/usr/bin/python3.6 dyfc    Note that your command prompt will prefix with **(dyfc)** |
| 6. | **Finally, install the dependency packages.**  $ pip install -r requirements.txt  *Note that this process will take a while to complete.* |

Part D: Setting Up A Flask Application Webhook

|  |  |
| --- | --- |
| PythonAnywhere is an online integrated development environment (IDE) and web hosting service (Platform as a service) based on the Python programming language. it provides in-browser access to server-based Python and Bash command-line interfaces, along with a code editor with syntax highlighting. Program files can be transferred to and from the service using the user's browser. | |
| 1. | Return to Pythonanywhere Dashboard.  Select **Web** on the navigation bar.  Choose **Add a new web app** and click **Next.** |
| 2. | Select **Manual Configuration** and click **Next**.    Choose Python 3.6 from the list and click **Next** twice. |
| 3. | In the Configuration page, scroll to the **Code** section.  Before:    Update the path to the source code and working directory to where the codes are saved.  After: (for reference : replace rambutantree with your account name) |
| 4. | Click on the WSGI configure file.    Search for the codes with  For line 109, please change the path to your own account.    Click **Save.** |
| 5. | In the Configuration page, scroll to the **Virtual Environment** section.  Before:    Update the path to the virtual environment.  After: (for reference only, replace rambutantree with your account name) |
| 6. | Return to the top of the Configuration page.  Click on **Reload.** |
| 7. | In the console, select **Files.**  On the directory navigation sidebar, change to subdirectory **DYFC/dyfc-webhook**.  Except for the three files listed on the right, you can delete the rest of subfolder folders and files.    Click on **Upload a File**.  Replace the file **secret-key-file.json** with the service key that you have created in a prior step. It is important to retain the same naming convention. |
| 8. | Test that the Flask App has been properly setup.  From the Dashboard, select **Web.**    Click on the given url (e.g. <http://rambutantree.pythonanywhere.com/>).  You should see a new page with the following contents.    Note down the URL. You will need it to setup Fulfillment in Dialogflow. |

Part E: Setting Up Dialogflow Agent

|  |  |
| --- | --- |
| 1. | Visit the link below and request for access to a preconfigured Diaglowflow Agent.  <https://drive.google.com/file/d/1HVCt9EaEW7dvKr0vJ71E8v6mBQYrz0kJ/view?usp=sharing>  or <https://tinyurl.com/agentdyfc>  Download the zip file. Do NOT uncompressed it. |
| 2. | Create a new agent in Dialogflow. |
| 3. | From the side navigation panel, select the **Gear** icon next to the agent name.  Choose the **Export and Import** tab.  Click **IMPORT FROM ZIP**. |
| 4. | Select the zip file for uploading.  Then, type **IMPORT** and click **IMPORT**. |
| 5. | From the side navigation panel, select the **Fulfilment module**.  Update the field URL using the web services URL that you received from PythonAnywhere.  Remember to include **/webhook** at the end of the URL.  *Example:*  *https://rambutantree.pythonanywhere.com/webhook* |
| 6. | Test the agent against the following requests.  User Request: Are you able connect to the webhook?  Chatbot Reply: You have made a successful connection to the webhook on 2022-01-02 17:07:35  User Request : Tell me the pickup time for Bedok.  Chatbot Reply: The bus will depart from Bedok at Time 1:56 PM  User Request : Please arrange for someone call me at 98769876. My name is Peter  Chatbot Reply: Hi Peter, sorry I can't help you now. But, someone will call you back at 98769876. Talk to you soon. We've got your information in the spreadsheet.  Check if your Google Sheet (sheet CallbackRequest) is able to log the information that was provided in the last request. |