

Steps overview:

Part A: Setting Google Sheet

Part B: Setting Google Service Account

Part C: Setting Pythonanywhere.com

Part D: Setting Flask Application Webhook

Part E: Importing Dialogflow Agent

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/1rkssSCLbDrNjQY1Qavot2nLtJ43Ge1ndejrJ1EEd_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.



Copy document

Would you like to make a copy of **DYFC-GSheet-Backend(student version)**?



[Make a copy](#)

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.

New Project

⚠ You have 5 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *

MyDYFC

?

Project ID: mydyfc. It cannot be changed later. [EDIT](#)

Location *

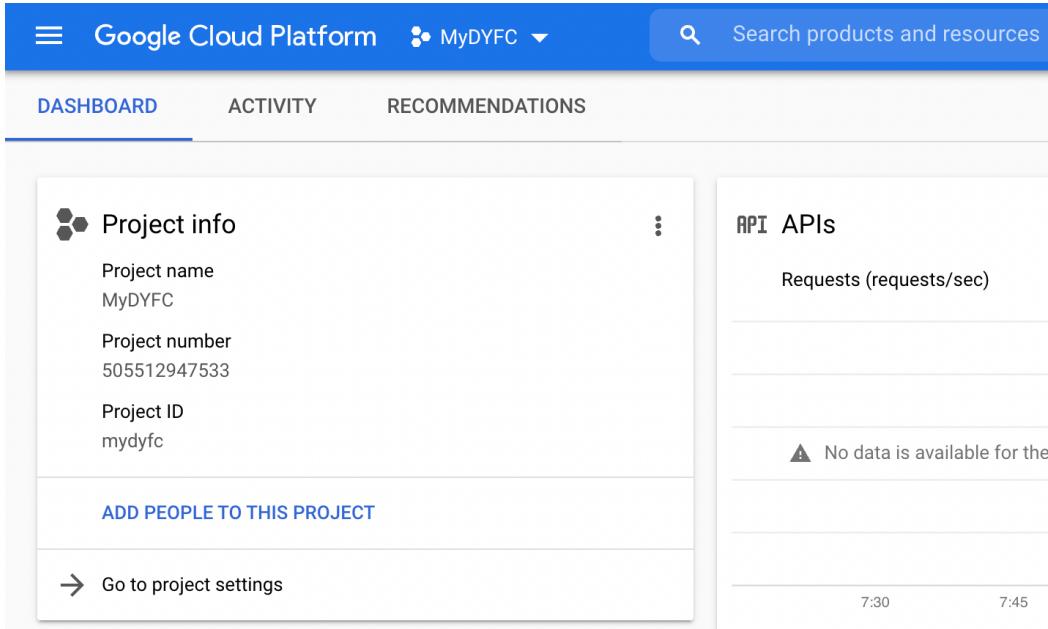
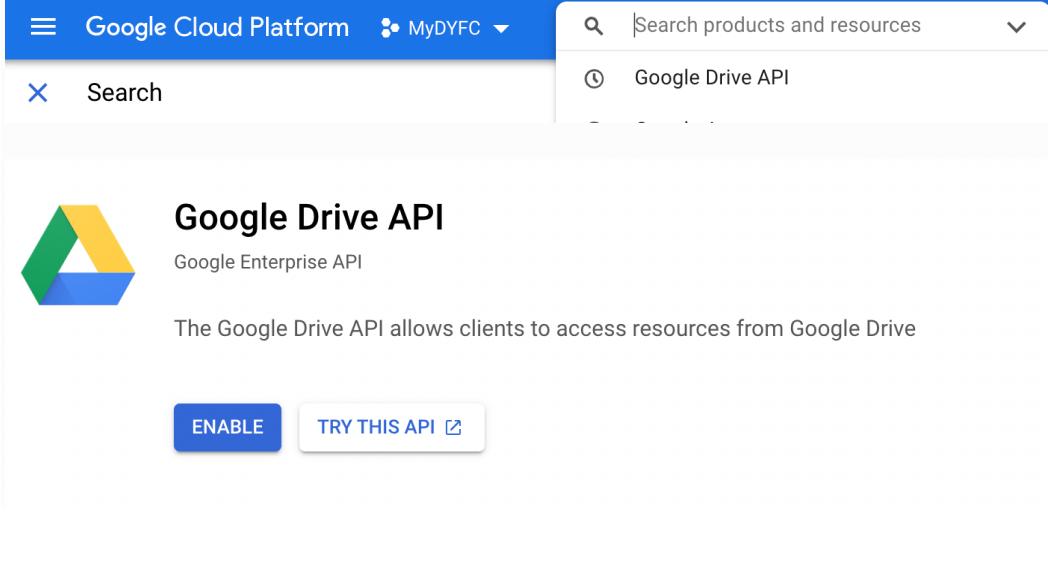
No organization

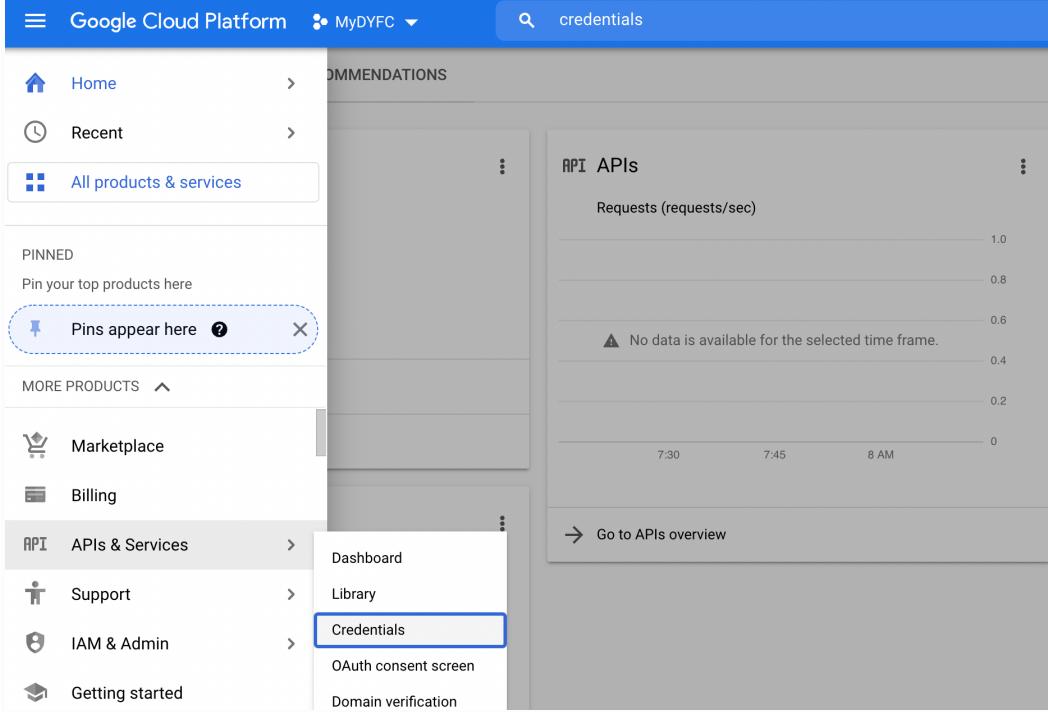
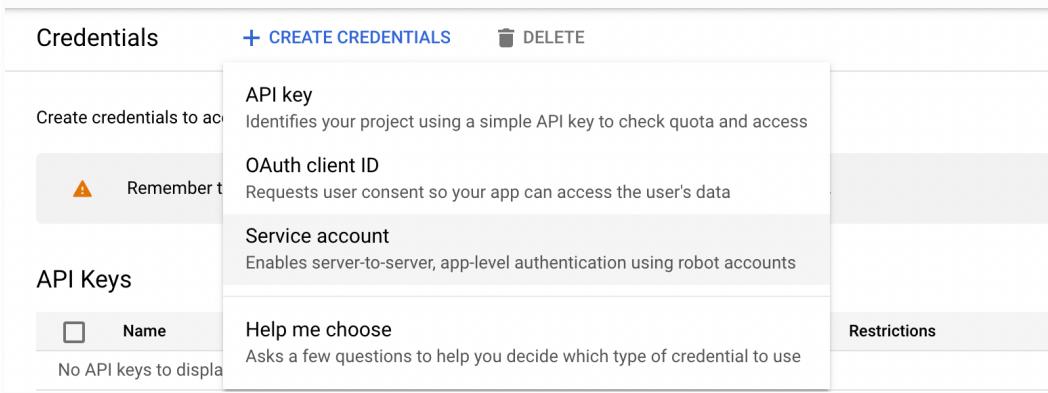
[BROWSE](#)

Parent organization or folder

[CREATE](#)

[CANCEL](#)

	Click CREATE . The console navigates to the Dashboard page and your project is created within a few minutes.
4.	<p>From the top bar, click the triangle icon (located next to the Google Cloud Platform)</p> <p>Select the project name that you have created.</p>  <p>The screenshot shows the Google Cloud Platform dashboard. On the left, the 'Project info' section displays the project name 'MyDYFC', project number '505512947533', and project ID 'mydyfc'. It also includes a link to 'ADD PEOPLE TO THIS PROJECT' and a button to 'Go to project settings'. On the right, the 'APIs' section shows a chart for 'Requests (requests/sec)' with a note: 'No data is available for the selected time range'. The x-axis shows time points at 7:30 and 7:45.</p>
5.	<p>In the box labelled “Search for APIs and Services”, search for “Google Drive API”.</p> <p>Click ENABLE.</p>  <p>The screenshot shows the search results for the Google Drive API. The search bar contains 'Google Drive API'. Below the search bar, the 'Google Drive API' card is displayed, featuring its logo, the text 'Google Enterprise API', and a description: 'The Google Drive API allows clients to access resources from Google Drive'. At the bottom of the card are two buttons: 'ENABLE' and 'TRY THIS API'.</p>
6.	Repeat the prior step by enabling Google Sheet API .

	<p>In the box labelled “Search for APIs and Services”, search for “Google Sheets API” and enable it.</p>
7.	<p>Open the side navigation panel. Go to “APIs & Services > Credentials” .</p>  <p>Choose “CREATE CREDENTIALS > Service account key”.</p>  <p>Fill out the form as shown. Click “Create” and “Done”.</p>

	<h2>Create service account</h2> <p>1 Service account details</p> <p>Service account name * <input type="text" value="mydyfc"/></p> <p>Display name for this service account</p> <p>Service account ID * <input type="text" value="mydyfc"/> @mydyfc.iam.gserviceaccount.com X C</p> <p>Service account description <input type="text" value="chatbot"/></p> <p>Describe what this service account will do</p> <p>CREATE AND CONTINUE</p> <p>2 Grant this service account access to project (optional)</p> <p>3 Grant users access to this service account (optional)</p> <p>DONE CANCEL</p> <p>You should return to the dashboard page.</p>																										
8.	<p>Press “Manage service accounts” above Service Accounts section.</p> <p>Credentials + CREATE CREDENTIALS DELETE</p> <p>Create credentials to access your enabled APIs. Learn more</p> <p>⚠ Remember to configure the OAuth consent screen with information about your application. CONFIGURE CONSENT SCREEN</p> <p>API Keys</p> <table border="1"> <thead> <tr> <th><input type="checkbox"/> Name</th> <th>Creation date ▼</th> <th>Restrictions</th> <th>Key</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td colspan="5">No API keys to display</td> </tr> </tbody> </table> <p>OAuth 2.0 Client IDs</p> <table border="1"> <thead> <tr> <th><input type="checkbox"/> Name</th> <th>Creation date ▼</th> <th>Type</th> <th>Client ID</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td colspan="5">No OAuth clients to display</td> </tr> </tbody> </table> <p>Service Accounts</p> <table border="1"> <thead> <tr> <th><input type="checkbox"/> Email</th> <th>Name ↑</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> mydyfc@mydyfc.iam.gserviceaccount.com</td> <td>myDYFC</td> <td>edit trash</td> </tr> </tbody> </table>	<input type="checkbox"/> Name	Creation date ▼	Restrictions	Key	Actions	No API keys to display					<input type="checkbox"/> Name	Creation date ▼	Type	Client ID	Actions	No OAuth clients to display					<input type="checkbox"/> Email	Name ↑	Actions	<input type="checkbox"/> mydyfc@mydyfc.iam.gserviceaccount.com	myDYFC	edit trash
<input type="checkbox"/> Name	Creation date ▼	Restrictions	Key	Actions																							
No API keys to display																											
<input type="checkbox"/> Name	Creation date ▼	Type	Client ID	Actions																							
No OAuth clients to display																											
<input type="checkbox"/> Email	Name ↑	Actions																									
<input type="checkbox"/> mydyfc@mydyfc.iam.gserviceaccount.com	myDYFC	edit trash																									

Press on : near recently created service account and select “Manage keys”

The screenshot shows the Google Cloud Service Accounts page for a project named "MyDYFC". The page includes a header with "Service accounts", "+ CREATE SERVICE ACCOUNT", "DELETE", and "+ MANAGE ACCESS". Below the header, it says "Service accounts for project "MyDYFC"" and provides a brief description of what service accounts are. It also mentions organization policies for securing service accounts. A table lists service accounts, with one account named "myDYFC" selected. A context menu is open next to this account, with "Manage keys" highlighted.

Filter	Enter property name or value	Status	Name ↑	Description	Key ID	Key creation date	OAuth 2 Client	Actions
<input checked="" type="checkbox"/>	Email	Green checkmark	myDYFC	chatbot	No		102067836621	⋮
<input checked="" type="checkbox"/>	Key	Green checkmark	mydyfc@mydyfc.iam.gserviceaccount.com					⋮

Manage details
Manage permissions
Manage keys
View metrics

Click on “ADD KEY > Create new key”.

Keys

⚠ Service account keys could pose a security risk if compromised. We recommend you avoid downloading them. You can learn more about the best way to authenticate service accounts on [Identity Federation](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#).

[Learn more about setting organization policies for service accounts](#)

ADD KEY ▾

Create new key

Key creation date

Key expiration date

Upload existing key

Select **JSON key type** and press “Create”.

Create private key for "myDYFC"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

JSON

Recommended

P12

For backward compatibility with code using the P12 format

CANCEL

CREATE

You will automatically download a JSON file with credentials. It may look like this:

```
{  
  "type": "service_account",  
  "project_id": "api-project-XXX",  
  "private_key_id": "2cd ... ba4",  
  "private_key": "-----BEGIN PRIVATE KEY-----\nNrDyLw ... jINQh/9\n-----END PRIVATE KEY-----\n",  
  "client_email": "473000000000-yoursisdifferent@developer.gserviceaccount.com",  
  "client_id": "473 ... hd.apps.googleusercontent.com",  
  ...  
}
```

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.

 **Share with people and groups** 

Add people and groups

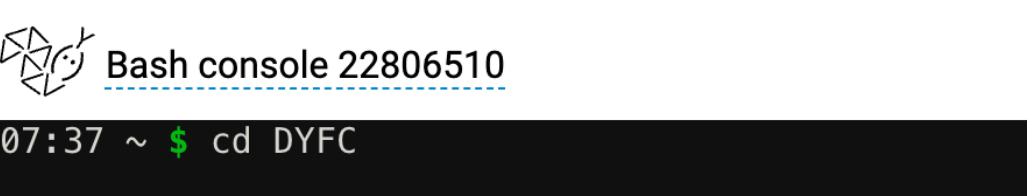
 DYFC Trainer (you) dyfctrainer@gmail.com	<i>Owner</i>
 dyfc-587@eeee-qcgs.iam.gserviceaccount.com dyfc-587@eeee-qcgs.iam.gserviceaccount.com	<i>Editor</i> ▾

[Send feedback to Google](#) 

	If you don't do this, you'll get a <code>gspread.exceptions.SpreadsheetNotFound</code> 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.	<p>From the navigator bar, select Consoles .</p> <p>Then Start a new console by choosing Bash .</p>  <p>Start a new console:</p> <p>Python: 3.8 / 3.7 / 3.6 / 3.5 / 2.7 IPython: 3.8 / 3.7 / 3.6 / 3.5 / 2.7 PyPy: 2 / 3 Other: Bash MySQL Custom: +</p>
4.	<p>a. Use the Bash Terminal console to create a new directory</p> <pre>\$ mkdir DYFC</pre>  <p>b. Change to the new directory</p> <pre>\$ cd DYFC</pre>  <p>c. Clone the repository .</p> <pre>\$ git clone https://github.com/lpomoeabatatas/dyfc-webhook.git .</pre> <p><i>Note that this process will take a while to complete.</i></p>



Bash console 22806510

```
07:38 ~/DYFC2 $ git clone https://github.com/Ipomoeabatatas/dyfc-webhook.git
```

A new sub-directory (dyfc-webhook) will be created. Change directory to this new sub-directory.

```
$ cd dyfc-webhook
```



Bash console 22806510

```
07:40 ~/DYFC2 $ cd dyfc-webhook
```

5.

Use the Bash Terminal console to create a new virtual environment (dyfc).

```
$ mkvirtualenv --python=/usr/bin/python3.6 dyfc
```



Bash console 22806437

```
07:17 ~/DYFC/dyfc-webhook $ mkvirtualenv --python=/usr/bin/python3.6 dyfc
```

Note that your command prompt will prefix with (**dyfc**)



Bash console 22806510

```
(dyfc2) 07:48 ~/DYFC2/dyfc-webhook (main)$
```

6.

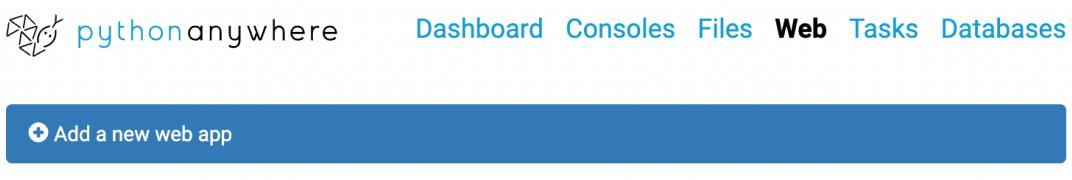
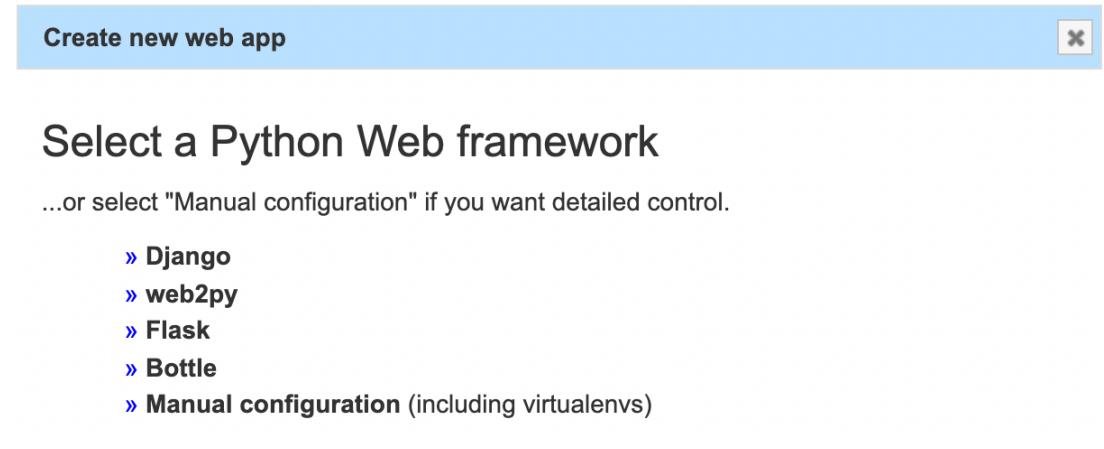
Finally, install the dependency packages.

```
$ 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.	<p>Return to Pythonanywhere Dashboard. Select Web on the navigation bar. Choose Add a new web app and click Next.</p> 
2.	<p>Select Manual Configuration and click Next.</p>  <p>Select a Python Web framework ...or select "Manual configuration" if you want detailed control.</p> <ul style="list-style-type: none">» Django» web2py» Flask» Bottle» Manual configuration (including virtualenvs) <p>Choose Python 3.6 from the list and click Next twice.</p>
3.	<p>In the Configuration page, scroll to the Code section.</p> <p>Before:</p> <p>Code:</p> <p>What your site is running.</p> <p>Source code: <i>Enter the path to your web app source code</i></p> <p>Working directory: <i>/home/rambutantree/</i> ↗ Go to directory</p> <p>WSGI configuration file: <i>/var/www/rambutantree_pythonanywhere_com_wsgi.py</i></p> <p>Python version: 3.6 </p>

	<p>Update the path to the source code and working directory to where the codes are saved.</p> <p>After: (for reference : replace rambutanree with your account name))</p> <p>Code:</p> <hr/> <p>What your site is running.</p> <table> <tbody> <tr> <td>Source code:</td> <td>/home/rambutanree/DYFC/dyfc-webhook</td> </tr> <tr> <td>Working directory:</td> <td>/home/rambutanree/DYFC/dyfc-webhook</td> </tr> <tr> <td>WSGI configuration file:</td> <td>/var/www/rambutanree_pythonanywhere_com_wsgi.py</td> </tr> <tr> <td>Python version:</td> <td>3.6 </td> </tr> </tbody> </table>	Source code:	/home/rambutanree/DYFC/dyfc-webhook	Working directory:	/home/rambutanree/DYFC/dyfc-webhook	WSGI configuration file:	/var/www/rambutanree_pythonanywhere_com_wsgi.py	Python version:	3.6 
Source code:	/home/rambutanree/DYFC/dyfc-webhook								
Working directory:	/home/rambutanree/DYFC/dyfc-webhook								
WSGI configuration file:	/var/www/rambutanree_pythonanywhere_com_wsgi.py								
Python version:	3.6 								
4.	<p>Click on the WSGI configure file.</p> <p>Code:</p> <hr/> <p>What your site is running.</p> <table> <tbody> <tr> <td>Source code:</td> <td>/home/rambutanree/DYFC/dyfc-webhook</td> </tr> <tr> <td>Working directory:</td> <td>/home/rambutanree/DYFC/dyfc-webhook</td> </tr> <tr> <td>WSGI configuration file:</td> <td>/var/www/rambutanree_pythonanywhere_com_wsgi.py</td> </tr> <tr> <td>Python version:</td> <td>3.6 </td> </tr> </tbody> </table> <p>Edit with the following codes.</p> <p>For line 109, please change the path to your own account.</p> <pre> 108 import sys 109 path = '/home/rambutanree/DYFC/dyfc-webhook' 110 111 if path not in sys.path: 112 sys.path.append(path) 113 114 from index import app as application # noqa ... </pre> <p>Click Save.</p>	Source code:	/home/rambutanree/DYFC/dyfc-webhook	Working directory:	/home/rambutanree/DYFC/dyfc-webhook	WSGI configuration file:	/var/www/rambutanree_pythonanywhere_com_wsgi.py	Python version:	3.6 
Source code:	/home/rambutanree/DYFC/dyfc-webhook								
Working directory:	/home/rambutanree/DYFC/dyfc-webhook								
WSGI configuration file:	/var/www/rambutanree_pythonanywhere_com_wsgi.py								
Python version:	3.6 								
5.	<p>In the Configuration page, scroll to the Virtual Environment section.</p> <p>Before:</p>								

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

[Enter path to a virtualenv, if desired](#)

Update the path to the virtual environment.

After: (for reference only, replace rambutanree with your account name)

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

[/home/rambutanree/.virtualenvs/dyfc](#)

6.

Return to the top of the Configuration page.

Click on **Reload**.

Configuration for [rambutanree.pythonanywhere.com](#)

Reload:

 [Reload rambutanree.pythonanywhere.com](#)

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.



pythonanywhere

Dashboard Consoles Files Web

/home/rambutantree/DYFC/ dyfc-webhook

[Open Bash console here](#)

39% full – 199.0 MB of your 51

Directories

Files

Enter new directory name

New directory

Enter new file name, eg hello.py

__pycache__/



index.py

[Download](#) [Edit](#) [Delete](#) 2022-01-01 04:50 6.5 KB

requirements.txt

[Download](#) [Edit](#) [Delete](#) 2022-01-01 03:55 458 bytes

secret-key-file.json

[Download](#) [Edit](#) [Delete](#) 2022-01-01 04:49 2.2 KB

[Upload a file](#)

100MB maximum size

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**.



pythonanywhere

Dashboard Consoles Files Web

rambutantree.pythonanywh...

Configuration for
rambutantree.pythonanywhere.com

[Add a new web app](#)

Click on the given url (e.g. <http://rambutantree.pythonanywhere.com/>).

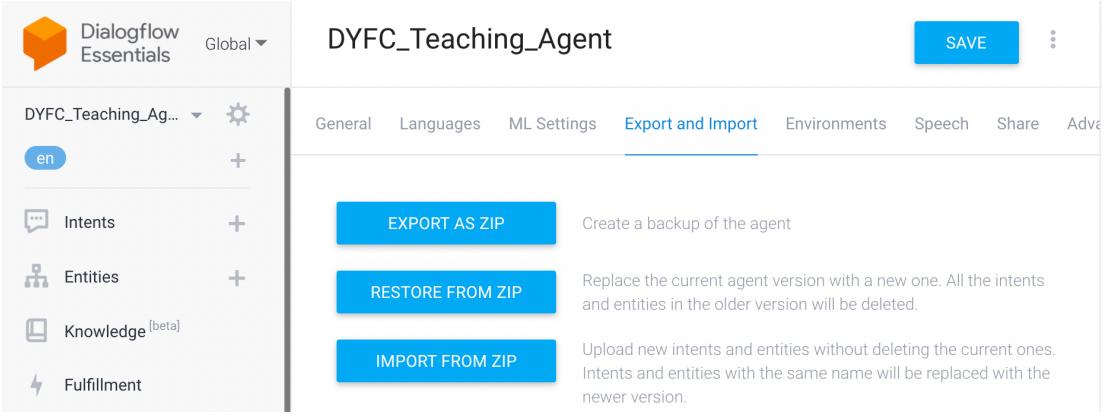
You should see a new page with the following contents.

← → ⌂ Not Secure | rambutantree.pythonanywhere.com

Flask Webhook 2021 Is Deployed successful. It is not free from bug yet

Note down the URL. You will need it to setup Fulfillment in Dialogflow.

Part E: Setting Up Dialogflow Agent

1.	<p>Visit the link below and request for access to a preconfigured Dialogflow Agent.</p> <p>https://drive.google.com/file/d/1HVCt9EaEW7dvKr0vJ71E8v6mBQYrz0kJ/view?usp=sharing</p> <p>or https://tinyurl.com/agentdyfc</p> <p>Download the zip file. Do NOT uncompressed it.</p>
2.	<p>Create a new agent in Dialogflow.</p>
3.	<p>From the side navigation panel, select the Gear icon next to the agent name.</p> <p>Choose the Export and Import tab.</p> <p>Click IMPORT FROM ZIP.</p> 
4.	<p>Select the zip file for uploading.</p> <p>Then, type IMPORT and click IMPORT.</p>

	<h2>Upload agent</h2> <p>Upload a zip file of a previously exported agent.</p> <p>Important: Intents and entities that you upload will replace existing intents and entities with the same name.</p> <div style="border: 1px dashed #ccc; padding: 10px; text-align: center;"> <p>Drop files here to attach them or</p> <p>SELECT FILE</p> </div> <p>DYFC_Teaching_Agent.zip</p> <p>IMPORT</p> <hr/> <div style="text-align: right;"> <p>IMPORT CANCEL</p> </div>
5.	<p>From the side navigation panel, select the Fulfilment module.</p> <p>Update the field URL using the web services URL that you received from PythonAnywhere. Remember to include /webhook at the end of the URL.</p> <p><i>Example:</i></p> <p>https://rambutantree.pythonanywhere.com/webhook</p> 

	<p>Fulfillment</p> <p>Webhook ENABLED</p> <p>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.</p> <p>URL* <input type="text" value="https://rambutantree.pythonanywhere.com/webhook"/></p> <p>BASIC AUTH <input type="text" value="Enter username"/> <input type="text" value="Enter password"/></p> <p>HEADERS <input type="text" value="Enter key"/> <input type="text" value="Enter value"/></p>
6.	<p>Test the agent against the following requests.</p> <p>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</p> <p>User Request : Tell me the pickup time for Bedok. Chatbot Reply: The bus will depart from Bedok at Time 1:56 PM</p> <p>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.</p> <p>Check if your Google Sheet (sheet CallbackRequest) is able to log the information that was provided in the last request.</p>