

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/1rkssSCLbDrNjQY1Qavot2nLtJ43Ge1ndejrJ1EEd\\_4/copy?usp=sharing](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 into your own Google Drive account.



### 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 \*

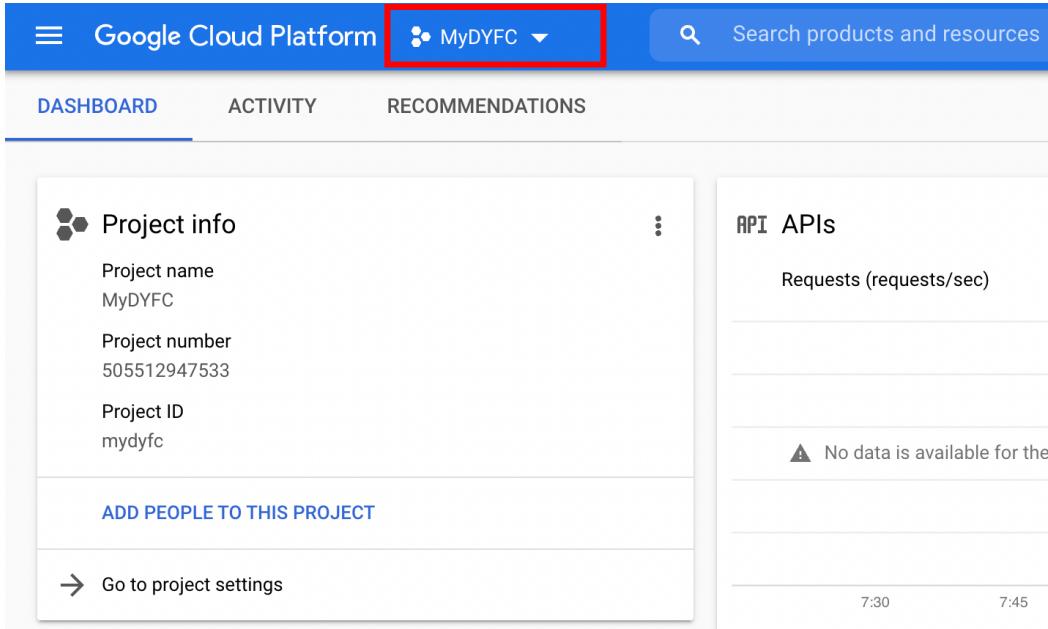
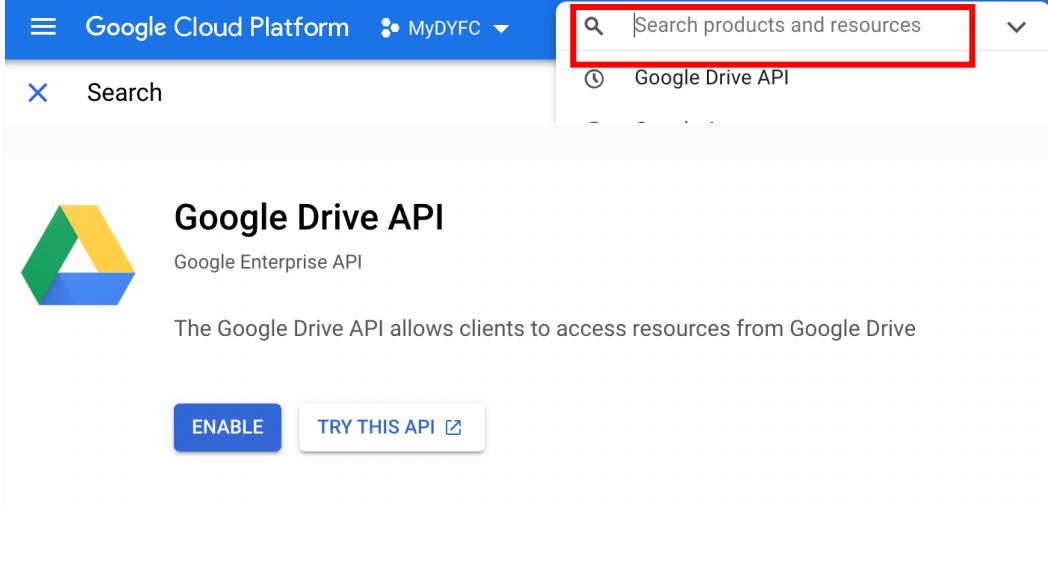
No organization

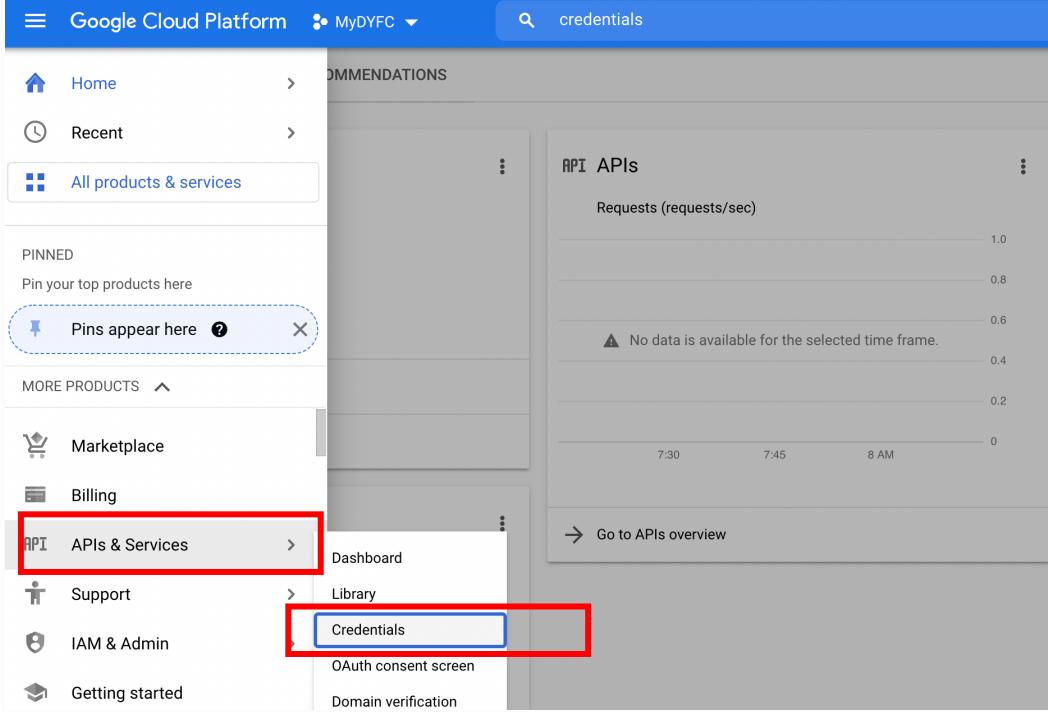
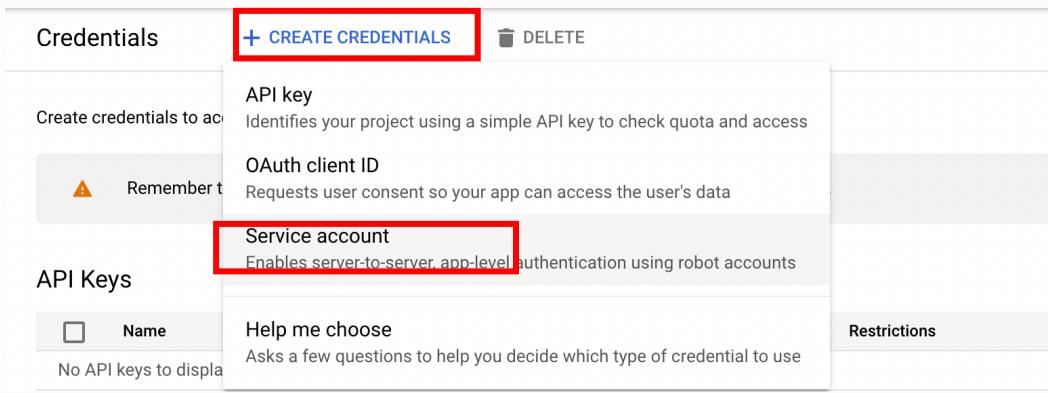
[BROWSE](#)

Parent organization or folder

[CREATE](#)

[CANCEL](#)

	Click <b>CREATE</b> . The console navigates to the Dashboard page and your project is created within a few minutes.
4.	<p>From the top bar, click the <b>triangle icon</b> (located next to the Google Cloud Platform)</p> <p>Select the project name that you have created.</p> 
5.	<p>In the <b>box</b> labelled “Search for APIs and Services”, search for “<b>Google Drive API</b>”.</p> <p>Click <b>ENABLE</b>.</p> 
6.	Repeat Step 5, search for <b>Google Sheet API</b> and enable the service.

	<p>In the box labelled “Search for APIs and Services”, search for “<b>Google Sheets API</b>” and enable it.</p>
7.	<p>Open the side navigation panel. Go to “<b>APIs &amp; Services &gt; Credentials</b>” .</p>  <p>The screenshot shows the Google Cloud Platform interface. The top navigation bar has "Google Cloud Platform" and "MyDYFC" dropdowns, and a search bar with "credentials". The left sidebar has "Home", "Recent", and "All products &amp; services" pinned. Below that is a "PINNED" section with "Pins appear here". Under "MORE PRODUCTS", "Marketplace", "Billing", and "API &amp; Services" are listed. "API &amp; Services" is highlighted with a red box. A dropdown menu for "API &amp; Services" shows "Dashboard", "Library", and "Credentials", with "Credentials" also highlighted with a red box. To the right is a chart titled "API APIs" showing requests over time.</p> <p>Choose “<b>+CREATE CREDENTIALS &gt; Service account key</b>”.</p>  <p>The screenshot shows the "Credentials" page. At the top, there's a "CREATE CREDENTIALS" button with a red box around it. Below it, there are two sections: "API key" and "OAuth client ID". Under "API Keys", there's a "Service account" button with a red box around it. The "Help me choose" section is also visible.</p> <p>Fill out the form as shown. Click “<b>CREATE</b> ” and “<b>DONE</b>”.</p>

	<p>Create service account</p> <p><b>1 Service account details</b></p> <p>Service account name * mydyfc</p> <p>Display name for this service account</p> <p>Service account ID * mydyfc @mydyfc.iam.gserviceaccount.com <b>X C</b></p> <p>Service account description chatbot</p> <p>Describe what this service account will do</p> <p><b>CREATE AND CONTINUE</b></p> <p><b>2 Grant this service account access to project (optional)</b></p> <p><b>3 Grant users access to this service account (optional)</b></p> <p><b>DONE</b>    <b>CANCEL</b></p> <p>You should return to the dashboard page.</p>																										
8.	<p>Press “<b>Manage service accounts</b>” on the right side of Service Accounts section.</p> <p>Credentials    <a href="#">+ CREATE CREDENTIALS</a>    <a href="#">DELETE</a></p> <p>Create credentials to access your enabled APIs. <a href="#">Learn more</a></p> <p><b>API Keys</b></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><b>OAuth 2.0 Client IDs</b></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><b>Service Accounts</b></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><a href="#"></a> <a href="#"></a></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	<a href="#"></a> <a href="#"></a>
<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	<a href="#"></a> <a href="#"></a>																									

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

Service accounts    + CREATE SERVICE ACCOUNT    DELETE    + MANAGE ACCESS

Service accounts for project "MyDYFC"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts.](#)

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies.](#)

Filter	Enter property name or value	Status	Name ↑	Description	Key ID	Key creation date	OAuth 2 Client	⋮
<input checked="" type="checkbox"/>	Email	Green checkmark	myDYFC	chatbot	No keys	10206783662	...	
<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

Upload existing key

Key creation date

Key expiration date

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

---

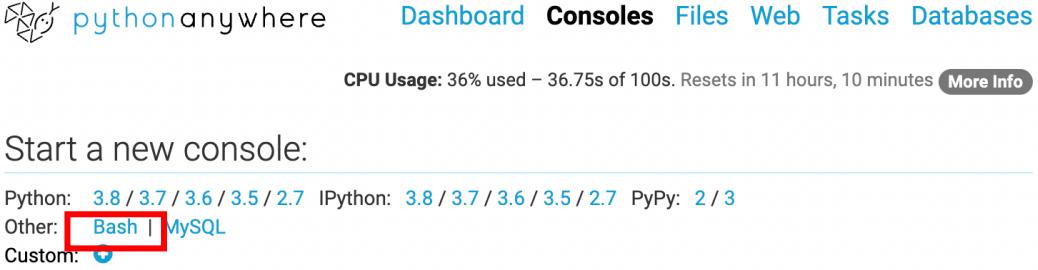
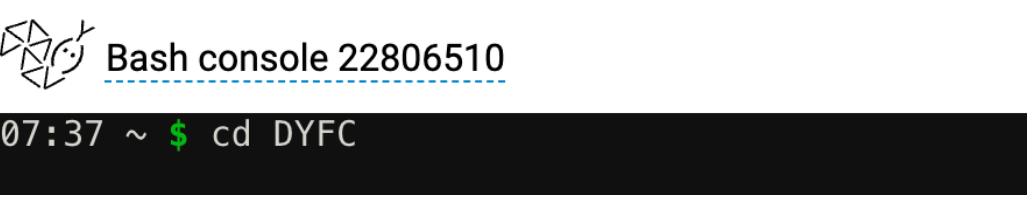
 <b>DYFC Trainer (you)</b> dyfctrainer@gmail.com	<i>Owner</i>
 <b>dyfc-587@eeee-qcgs.iam.gserviceaccount.com</b> dyfc-587@eeee-qcgs.iam.gserviceaccount.com	Editor ▾

[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 <b>secret-key-file.json</b> . 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 <a href="https://www.pythonanywhere.com">https://www.pythonanywhere.com</a> . Follow the onboarding instructions.
2.	Login to Pythonanywhere.com
3.	<p>From the navigator bar, select <b>Consoles</b>.</p> <p>Then Start a new console by choosing <b>Bash</b>.</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: <b>Bash</b>   MySQL Custom: </p>
4.	<p>a. Use the Bash Terminal console to create a new directory. Type the command below.</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 <a href="https://github.com/lpomoeabatatas/dyfc-webhook.git">https://github.com/lpomoeabatatas/dyfc-webhook.git</a> .</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.

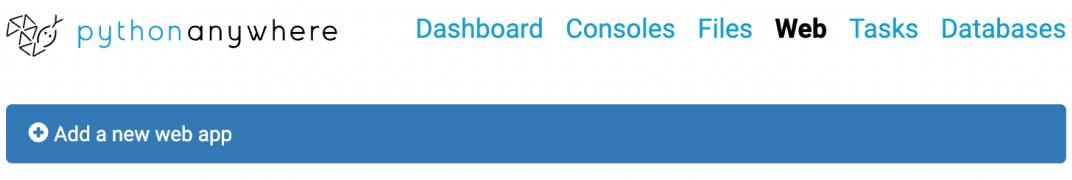
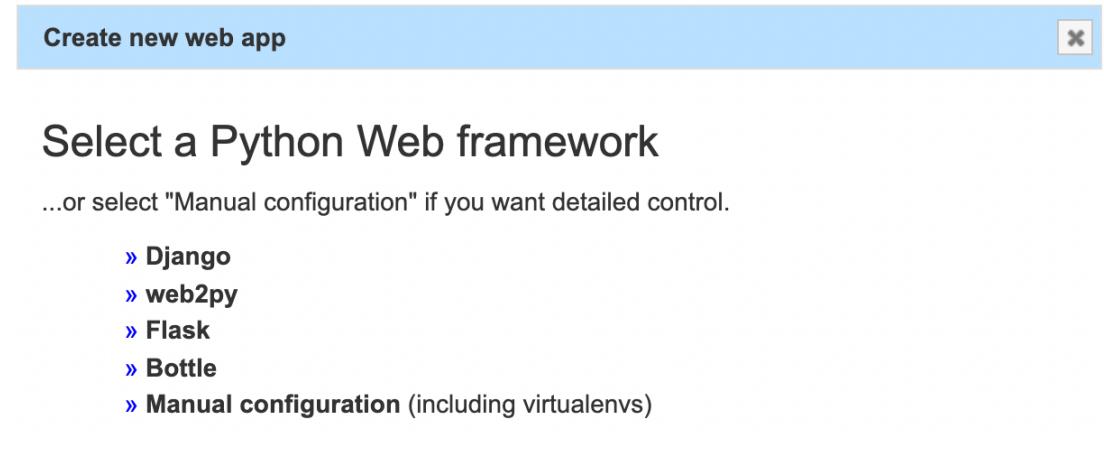
**Finally, install the dependency packages.**

```
$ 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 <b>Web</b> on the navigation bar. Choose <b>Add a new web app</b> and click <b>Next</b>.</p> 
2.	<p>Select <b>Manual Configuration</b> and click <b>Next</b>.</p>  <p><b>Select a Python Web framework</b> ...or select "Manual configuration" if you want detailed control.</p> <ul style="list-style-type: none"><li>» Django</li><li>» web2py</li><li>» Flask</li><li>» Bottle</li><li>» <b>Manual configuration</b> (including virtualenvs)</li></ul> <p>Choose Python 3.6 from the list and click <b>Next</b> twice.</p>
3.	<p>In the Configuration page, scroll to the <b>Code</b> 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> <span style="float: right;">↗ Go to directory</span></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 <b>rambutantree</b> with your account name)</p> <p>Code:</p> <p>What your site is running.</p> <table> <tr> <td>Source code:</td><td><a href="#">/home/rambutantree/DYFC/dyfc-webhook</a></td></tr> <tr> <td>Working directory:</td><td><a href="#">/home/rambutantree/DYFC/dyfc-webhook</a></td></tr> <tr> <td>WSGI configuration file:</td><td><a href="#">/var/www/rambutantree_pythonanywhere_com_wsgi.py</a></td></tr> <tr> <td>Python version:</td><td>3.6 </td></tr> </table>	Source code:	<a href="#">/home/rambutantree/DYFC/dyfc-webhook</a>	Working directory:	<a href="#">/home/rambutantree/DYFC/dyfc-webhook</a>	WSGI configuration file:	<a href="#">/var/www/rambutantree_pythonanywhere_com_wsgi.py</a>	Python version:	3.6 
Source code:	<a href="#">/home/rambutantree/DYFC/dyfc-webhook</a>								
Working directory:	<a href="#">/home/rambutantree/DYFC/dyfc-webhook</a>								
WSGI configuration file:	<a href="#">/var/www/rambutantree_pythonanywhere_com_wsgi.py</a>								
Python version:	3.6 								
4.	<p>Click on the WSGI configure file.</p> <p>Code:</p> <p>What your site is running.</p> <table> <tr> <td>Source code:</td><td><a href="#">/home/rambutantree/DYFC/dyfc-webhook</a></td></tr> <tr> <td>Working directory:</td><td><a href="#">/home/rambutantree/DYFC/dyfc-webhook</a></td></tr> <tr> <td>WSGI configuration file:</td><td><a href="#">/var/www/rambutantree_pythonanywhere_com_wsgi.py</a></td></tr> <tr> <td>Python version:</td><td>3.6 </td></tr> </table> <p>Search for the codes with</p> <p>For line 109, please change the path to your own account.</p> <pre> 108 import sys 109 path = '/home/rambutantree/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 <b>Save</b>.</p>	Source code:	<a href="#">/home/rambutantree/DYFC/dyfc-webhook</a>	Working directory:	<a href="#">/home/rambutantree/DYFC/dyfc-webhook</a>	WSGI configuration file:	<a href="#">/var/www/rambutantree_pythonanywhere_com_wsgi.py</a>	Python version:	3.6 
Source code:	<a href="#">/home/rambutantree/DYFC/dyfc-webhook</a>								
Working directory:	<a href="#">/home/rambutantree/DYFC/dyfc-webhook</a>								
WSGI configuration file:	<a href="#">/var/www/rambutantree_pythonanywhere_com_wsgi.py</a>								
Python version:	3.6 								
5.	<p>In the Configuration page, scroll to the <b>Virtual Environment</b> 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](#) 2022-01-01 04:50 6.5 KB

requirements.txt

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

secret-key-file.json

[Download](#) [Edit](#) 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/>).

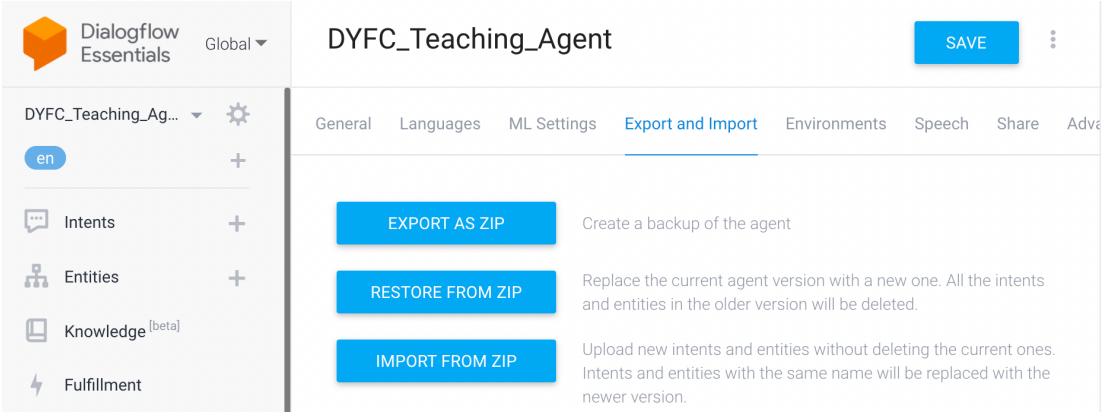
You should see a new page with the following contents.

← → ⌂ Not Secure | rambutan.tree.pythonanywhere.com

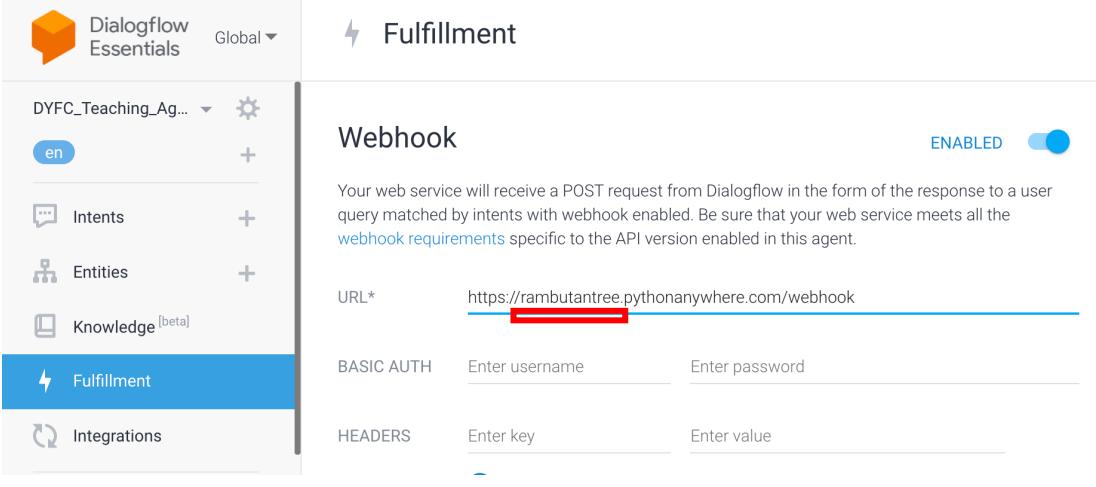
Flask Webhook 2021 Is Deployed successful. It is not free from bug yet

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><a href="https://drive.google.com/file/d/1HVCt9EaEW7dvKr0vJ71E8v6mBQYrz0kJ/view?usp=sharing">https://drive.google.com/file/d/1HVCt9EaEW7dvKr0vJ71E8v6mBQYrz0kJ/view?usp=sharing</a></p> <p>or <a href="https://tinyurl.com/agentdyfc">https://tinyurl.com/agentdyfc</a></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 <b>Gear</b> icon next to the agent name.</p> <p>Choose the <b>Export and Import</b> tab.</p> <p>Click <b>IMPORT FROM ZIP</b>.</p> 
4.	<p>Select the zip file for uploading.</p> <p>Then, type <b>IMPORT</b> and click <b>IMPORT</b>.</p>

	<h2>Upload agent</h2> <p>Upload a zip file of a previously exported agent.</p> <p><b>Important:</b> 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><b>SELECT FILE</b></p> </div> <p>DYFC_Teaching_Agent.zip</p> <p><b>IMPORT</b></p> <hr/> <div style="text-align: right;"> <p><b>IMPORT</b>    <b>CANCEL</b></p> </div>
5.	<p>From the side navigation panel, select the <b>Fulfilment module</b>.</p> <p>Update the field URL using the web services URL that you received from PythonAnywhere. Remember to include <b>/webhook</b> at the end of the URL.</p> <p><i>Example:</i></p> <p><a href="https://rambutantree.pythonanywhere.com/webhook">https://rambutantree.pythonanywhere.com/webhook</a></p> 

	 <p>The screenshot shows the Dialogflow Fulfillment interface. On the left, there's a sidebar with 'DYFC_Teaching_Ag...' (language: en), 'Intents', 'Entities', 'Knowledge [beta]', 'Fulfillment' (which is selected and highlighted in blue), and 'Integrations'. The main area is titled 'Webhook' with a status of 'ENABLED' and a blue toggle switch. It includes fields for 'URL*' (https://rambutantree.pythonanywhere.com/webhook), 'BASIC AUTH' (username and password inputs), and 'HEADERS' (key and value inputs). A note at the top states: '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>
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>