

Task 5: Create a demo app using Flask

Introduction

Flask is a Python framework that allows you to create webpages and create APIs to manipulate data using those webpages. It is similar in concept to Django, but where Django is more defined by nature, Flask is by design more simple and user-extensible. In Django there is more functionality built-in and guidelines that should be followed, whereas in Flask, it is up to the user to add the functionality that they desire.

Description of the Task

My task is to create a demo app using Flask. My definition and interpretation of a demo app is to create an app that is useful, but not production-ready, a Proof of Concept. A demo app should demonstrate the workings and functions of the technology, and show the usecases that the technology is most used for.

As such, I will be creating a page that users can input their name and email address into.

Then, one of the key uses of Flask, providing methods to use HTTP requests, such as GET, POST, etc. will be used. In this case, we will be using the POST request, to forward the information input by the user.

Next, on a different API endpoint, /data, we will use the GET request, to obtain the data which was input by the user and display it.

We will be storing the input data into a simple JSON file, for demonstration purposes.

And of course, we will have HTML pages with some styling to render a User Interface for all these operations.

Requirements

Flask

Instructions:

As this is a demo app, it is not platform dependent. As such, the only prerequisite is to install the required library: Flask, and to make sure you are working from the directory that the code is in.

The easiest way to install the required libraries is to use a virtual environment. The way to create and use a virtual environment (after ensuring you are in the directory that the code is located) is to run the commands: `python -m virtualenv venv`, `source venv/bin/activate` and `pip install -r requirements.txt`.

Then, just run the application by using the command `python app.py`

Next, open up a web browser, and navigate to the address that will be printed in the terminal, usually it will be `http://localhost:5000` or `http://127.0.0.1:5000`, both will work.

Enter the details that the app requires, and press the button that will initiate the POST request.

To view the data that you have input, press the button **View Data** , which will automatically refer you to the `/data` endpoint, which will use the GET request to obtain and then display the data.

The data is being stored using JSON in a file named `data.json`, Flask has a module named `jsonify`, which allows you to use all things JSON.

Work done by:

Aseem Aniket Athale

athaleaseem@gmail.com