restAPI using Flask

Laboratory Report

Author: Hector Alarcon Flores

Date: January 2020

Índex

1 Introduction

The idea behind this project is to create a functional rest API which enables clients to send API requests to manipulate data. For instance, we will be using a set of data which will be our start point (a .csv document)

2 Objectives

Essentially this project is a way of learning the basics of Flask, however it is important to emphasise the following objectives :

- To understand the framework Flask (Python)
- To make use of basic API methods : GET, PUSH, PUT, DELETE
- To be able to work with a .csv document (internally)

3 Decisions

In this section I will explain the decisions I have made during the realisation of this project and why I have done it that way.

3.1 Database

In this kind of introductory project to Flask I haven't used any external database to work with data. Instead, I used the framework **pandas** to manipulate the data. In fact, there is a specific class called *database.py* on which everything related to database management is handled. **Therefore, the original .csv is never modified.**

3.2 UI / frontend

As I mentioned before this project consists of making a rest API from scratch, so there is no frontend developed. As a matter of fact, every request returns always a **JSON** with the necessary information, even if the data is not found.

3.3 Methods

There are 7 methods, 4 are to handle basic request of a rest API plus 2 more to handle specific information manually and one index method.

- Index method shows up the actual database on screen as a JSON.
- 3 methods GET : first two with URL-hardcoded type and value, and the last one that allows to search for any kind of type and value via *'/<type>/value'*.

- Method PUT: product search only by *ID* in URL. It modifies everything except the *ID*.
- Method PUSH: add product, but in case there is a product with the same ID as the new product, this one will be overwritten.
- Method DELETE: product deleted offering an ID via URL.

Every method has been linked to a URL using the next method :

@app.route('route/to/link', methods = ['API','METHOD'])

4 Resources used

This section will cover all the resources used to make possible this project :

- framework **Flask** to create the whole rest API.
- framework **pandas and numpy**.
- *Visual Studio* as a text editor.
- *Insomnia* to simulate requests.
- *GitHub* as a repository manager.

The project will be found on :

`https://github.com/hectorAlarcon/basic_restAPI`