Documentation - The Dev Challenge

author: Guilherme Santos Souza

Access the application: https://the-dev-challenge.herokuapp.com/

Settings

Language, libraries and frameworks

Ruby

- Ruby v2.5.0
- Ruby on Rails v5.2.6

Javascript

- Jquery v3.6.0

System

- Ubuntu v18.04

Database

- Sqlite3 - Development
- Postgres - Production

Project Structure

This project was made using the Ruby on Rails framework. Consequently, the MVC pattern was adopted to exploit the framework resources.

Models

One model was created which represents the purchases (each line of the file). The data was persisted in a Sqlite3 database, with the following fields:

- purchaser_name: string
- item_description: string
- item_price: float
- purchase_count: integer
- merchant_address: string
- merchant_name: string

The database setting and migrations are saved in the folder "db". The model class is defined in the folder "app/models".

Views and Controllers

Views and controllers are both responsible for client interaction. Two controllers were created, one to receive the first request plus respond with the home page HTML and another to process the request from the form. Each controller has a set of views to respond to the client. One and the other are located in the "app" folder.

The controller **home** only receives the GET request from the route "/". The corresponding view is an *HTML* page and its JS and CSS files are saved in the folder "lib/assets".

The controller **purchase** receives a POST request from the route "/purchases/post_file" with the ".tab" file, processes the text, and determines the total gross income. In this case, the response is a ".json' file that contains de gross income values.

Development Environment

To start the development database, run the following code on the terminal:

```
rails db:migrate RAILS_ENV=development
```

In order to start the development server, run the following server:

```
rails s -e development
```

Tests

The test files are saved in de folders "/test". Each model and controller was tested and also the user uploading file flow. In addition, the example file plus two alterations were saved in the "/test/fixtures" to be used during tests.

To run the tests, it must start the database in the test environment by the following code:

```
rails db:migrate RAILS_ENV=test
```

And then, run the following code:

```
rails t
```

Production Environment

In order to start a local production server, it must start the database in the environment by the following code:

```
rails db:migrate RAILS_ENV=production
```

And then, run the following:

```
rails s -e production
```

Heroku

To deploy the application using Heroku, follow the instructions of the link below:

https://devcenter.heroku.com/articles/getting-started-with-rails5

Run the following command before using `git push heroku master` to change the Heroku's stack:

```
heroku stack:set heroku-18
```

Then start the database with the following command:

```
heroku run rails db:migrate
```

Configurations

General settings are saved in the "/config" folder. The server settings are in the "puma.rb" file.