Python REST API with Django REST Framework

Description

It is data analysis API which accepts csv files and calculates different statistics of each column of this file. API works only with numerical data!

API functional

- accepts csv files
- calculates number of rows and columns
- calculates statistics of each column:
  1. minimum value
  2. maximum value
  3. mean
  4. 10th percentile
  5. 90th percentile
  6. missing values %
- stores all the data in database, so you can access the data about file even after server restart

API endpoints

- *upload_file/* - file upload via POST method
- *all_files/* - returns data about all files in database
- *file/{id}* - returns data about file with given id
- *all_details/* - returns details about all files in database (number of rows and columns, file id, and columns details)
- *file_details/{id}* - returns details files with given id
- *columns_details/* - returns details about all columns in database (column id, column order id, column name, statistics of this column)
- *column/{id}* - returns details about acolumn with given id
- *statistics/{file_id}/{statistic}* - returns list of given statistic in file with given id (example of usage *statistic/1/mean*)

API testsing tool

testing_tool.py is a python script that allows to interact with API via command line This script has a primitive console user interface which helps user to interact with API

Frameworks and important libraries

- django
- django-rest-framework
- pandas

Before installation

Make sure that you have Python installed on your PC

If you dont have it intalled, please install it: (https://www.python.org/)

Installation

1. Create a directory where api project will be stored

    1. Open Command Prompt
    2. Choose directory where you want to save the project

        ```
        example
        cd Desktop
        ```

    3. Make new directory here

        ```
        example
        mkdir RestAPI
        ```

    4. Navigate to this folder

        ```
        cd RestAPI
        ```

2. Create and activate a virtual environment for the project

    1. When you are inside the folder you have created earlier use this command

        ```
        python3 -m venv env
        ```

    2. Activate virtual environment

        ```
        env\Scripts\activate
        ```

    3. Upgrade your pip

        ```
        python -m pip install --upgrade pip
        ```

    4. Install required packages

        ```
        pip install django djangorestframework pandas requests
        ```

3. Extract folders

    1. extract `rest_api` to the folder you have created before (RestAPI)
    2. extract `rest_api_testing_tool` to the folder you have created before (RestAPI)

4. Setting up and staring the server

    1. Navigate to the folder with your project

        ```
        example
        cd Desktop\RestAPI
        ```

    2. Then navigate to rest_api folder

        ```
        cd rest_api
        ```

    3. Make migrations

```
    python manage.py makemigrations
    ```

4. Apply migrations

    ```
    python manage.py migrate
    ```

5. Synchronize your database

    ```
    python manage.py migrate --run-syncdb
    ```

6. Run the server

    ```
    python manage.py runserver
    ```

Before using testing tool

- Make sure that your server is running
- It is impossible to use this tool when server is not working

Testing tool guide

1. Open another Command Prompt and activate your virtual environment

    1. Navigate to the folder with your project

        ```
        example
        cd Desktop\RestAPI
        ```

    2. Activate virtual environment

        ```
        env\Scripts\activate
        ```

    3. Then navigate to rest_api_testing_tool folder

        ```
        cd rest_api_testing_tool
        ```

    4. Run script testing_tool.py

        ```
         python testing_tool.py
        ```

    5. Follow the instructions from script

Test files with data

Inside `example_files` folder there are some files with data. You can use them to test API.

Postman as an alternative way to test API

You can also use Postman to test API. This is much more easy way than testing API via command line. You will need to install a Postman desktop version to use this tool with your local server. After installation register or log in. Then create new workspace and click on `collections` button. In your workspace click `import` button and choose option `link` then use this link: https://www.getpostman.com/collections/f59ba619cc4c2487d8a3 Now you are able to test api in a very comfortable way.