HEROES & VILLIANS

Introduction

The heart of all great applications is data. At the end of the day, our job as software/web developers is to store, organize, and present data in a way that solves problems for clients and enhances their quality of life and businesses. Data becomes much more interesting when it grows and becomes more complex. The interaction of data then fuels more meaningful applications!

This project will be a crowning achievement of your knowledge of Django REST Framework so far. You will build a single app for your Django REST API, but with more complex data and opportunities for more advanced querying to satisfy the needs of requests being made to your API!

Technologies

Django REST Framework, Postman

Learning Objective

Increase skill with building and testing Web APIs using intermediate level Django REST Framework knowledge

💡 You will be using a collection of Postman requests that WE give you and building the API endpoints to make the requests work!

Resources

PowerPoints

- Intermediate Django REST Framework

Relevant Projects

- Django REST Framework Tutorial
- Django REST Products API

Other Resources

- Django REST Framework documentation: https://www.django-rest-framework.org/

Tasks

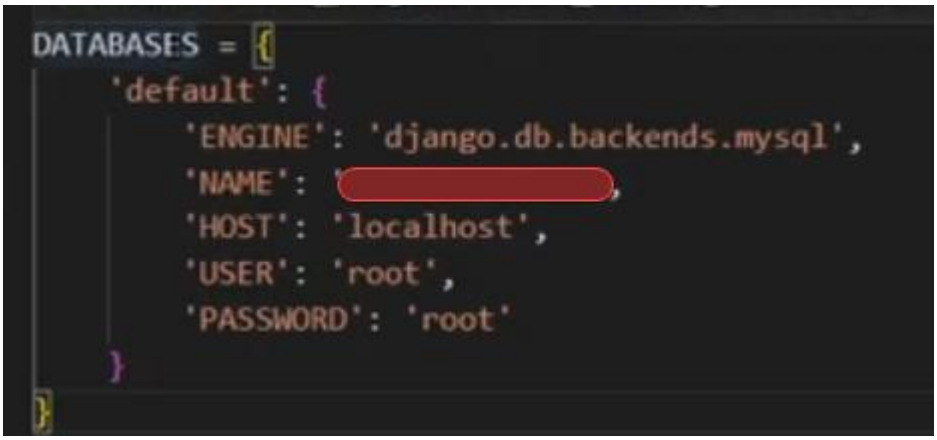
Build a Django REST API based on the provided user stories that meets the requirements of the provided Postman tests.

Setup Steps

1. Make a GitHub Repository `(** with Python gitignore **)`
2. Clone down repository to local computer
3. Open folder in VS code and create/activate a local venv
    a. "pipenv install"
    b. "pipenv shell"
4. Select the correct Python interpreter for the project

5. Install all necessary packages
    a. “pipenv install django”
    b. “pipenv install djangorestframework”
    c. Windows - “pipenv install mysqlclient”
    d. Mac - “pipenv install mysql-connector-python==8.0.26”
6. Create an initial Django project
    a. “django-admin startproject heroes_villains_project .”
7. Create a `local_settings.py` file and import it into your settings.py file to prevent your settings from being pushed to the public repository.
8. Cut & Paste DATABASES and SECRET_KEY from settings.py to local_settings.py. Change DATABASES to reflect the appropriate database NAME, ENGINE, USERNAME, PASSWORD, etc.

Windows:



Mac:

9. Push project to GitHub repo.

10. Create database in MySQL Workbench
11. Execute migrations commands
    a. “python manage.py migrate”
12. Create a new app for the supers, and an app for “super_types”
    a. “python manage.py startapp supers”
    b. “python manage.py startapp super_types”
13. Download the Postman collection (available on your online portal) and import the collection into Postman (File > Import, then upload the file into the window that pops up).
14. Postman will create a folder in your list of collections. This folder will contain all of the test requests that you will be building your API to successfully pass.

End Result

Your API will have all features as required by the user stories, and all of the requests provided in the Postman collection will have passed with success codes!