Project Documentation: Car Model CRUD API Backend

Project Overview

The Car Model CRUD API Backend is a Python Tornado project designed to provide a comprehensive API for managing car models. This documentation outlines the key features, setup instructions, and usage guidelines for the project.

Purpose

The purpose of this project is to offer a flexible and user-friendly API for CRUD operations on car models. Whether you're building an e-commerce platform, a car enthusiast website, or any application that requires car model management, this API serves as a reliable backend solution.

Setup and Installation

Prerequisites

Before you begin, ensure you have the following installed:

Python (version 3.1.11)

Pip package manager

SQLite or your preferred database system

Git (optional but recommended)

Installation Steps

Clone the repository from GitHub:

git clone https://github.com/Liya-23/for-metalogix.git

Navigate to the project directory:

cd for-metalogix

Install project dependencies:

pip install tornado

Configure the database connection:

Open main.py and locate the database configuration section.

Configure the database URL according to your setup.

Run database migration:

python migration.py

Start the server:

python main.py

Access the API and user interface:

Open your web browser and navigate to http://localhost:2040/

Database Models

The core data structure of the project revolves around the CarModel model, representing various attributes of car models. The model's structure is defined in the main.py file. To extend or modify the model, edit the CarModel class and run the migration script.

Authentication Logic

User registration and login functionality are implemented for secure access to the API. When users register, their passwords are securely hashed before storage. The authentication logic is contained within the main.py file.

CRUD Endpoints

The API provides the following endpoints for CRUD operations:

POST /findex/post:

Add a new car model to the database.

Request parameters: car specs, color, etc.

PATCH /findex/patch:

Update details of a specific car model.

Request parameters: fields to update.

DELETE /findex/delete:

Delete a specific car model.

GET /findex/get:

Retrieve a list of all car models.

User Interface

The user interface provides an interactive way to interact with the API. It consists of HTML files (register.html, login.html, index.html) that communicate with the API endpoints. The UI allows users to register, log in, view, add, edit, and delete car models.

To access the user interface, open your web browser and visit http://localhost:2040/ after starting the server.

Unit Tests

Comprehensive unit tests ensure the reliability and accuracy of API endpoints and authentication logic. Run the tests using the following command:

pytest main.py

Conclusion

The Car Model CRUD API Backend project offers a robust API for managing car models, complete with user authentication, CRUD functionality, a basic user interface, and thorough unit tests. By following the setup instructions and exploring the provided features, you can effectively integrate this backend solution into your projects.

For inquiries, feedback, or support, please contact:

Email: liyab1303@gmail.com

Email: liyabona.mtintsilana@codetelligence.org.za

I welcome your contributions and suggestions to further enhance this project!