INSTITUTO FEDERAL DO PIAUÍ - Campus Teresina Central
Curso Técnico em Desenvolvimento de Sistemas
Programação Web Backend / Turma 366/386 / 2024.2

Atividade de Programação Web Backend - API com Django/DRF

API-GEEK

Objetivo:
Desenvolver uma API REST com Django/DRF que permita realizar operações básicas para um sistema de gerenciamento de vendas de artigos voltados para jovens de 13 a 19 anos.

Os produtos são itens relacionados a séries, mangás/animes, games e cultura geek. A aplicação deve encapsular a lógica de negócio e banco de dados em uma classe de serviço, utilizando as técnicas abordadas em aula.

Requisitos da Atividade:

1. **Estrutura do Projeto:**
    - Utilize o framework **Django/DRF**.
    - Implemente a organização de aplicações/módulos.
    - Crie um banco de dados SQL simples utilizando o Models/**Migrations** para persistência dos dados.
    - Crie Models, Views, URLS, Serializers e demais itens.
2. **Modelos:**
    - **Produto:** Representa um item que está a venda na API
    - **Categoria:** Agrupamento de produto, ou seja, a que ele está vinculado, se é todo tipo Série, Game, Mangá
    - **Franquia/Título**
    - **Venda**
3. **Funcionalidades CRUD:**
    - **Criar Produto (Create):**
        - Endpoint para cadastrar um novo produto, contendo os seguintes atributos:
            - Nome do produto.
            - Descrição.
            - Preço.
            - Quantidade em estoque.
            - Categoria/Tipo (Séries, Mangás, Games, Filmes, etc.).
            - Franquia/Título (e.g., "Naruto", "The Witcher", "Marvel").
        - Valide os dados inseridos (e.g., preço não pode ser negativo).
    - **Listar Produtos (Read):**
        - Endpoint para retornar todos os produtos cadastrados.
        - Permita a filtragem de produtos por nome, preço, categoria ou franquia.
    - **Atualizar Produto (Update):**
        - Endpoint para atualizar as informações de um produto específico com base em seu ID.
    - **Atualizar Estoque Produto (Update):**
        - Endpoint para informar venda (quantidade) ou reposição de estoque (quantidade).
    - **Excluir Produto (Delete):**

        - Endpoint para remover um produto com base em seu ID.
    - **Vender**
        - Cria uma Venda com nome do cliente, Produto escolhido, quantidade, valor da venda, data/hora venda. Não deve permitir venda de produtos se o estoque não for suficiente
        - (Opcional) tente fazer um venda com vários produtos
    - **Repor Produto por ID**
        - Aumenta o estoque de um determinado produto.
4. **Relacionamento:**
    - O Atributo Categoria deve ser outro Model
    - O Atributo Franquia/Título deve ser também um Model, permitindo assim que vários produtos possam ver vinculados a uma franquia/título como Naruto.
5. **Regras de Negócio:**
    - Produtos só podem ser excluídos se estiverem fora de estoque.
    - Ao atualizar a quantidade em estoque de um produto, a API deve impedir a inserção de valores negativos.
    - Não é possível vender se não tiver Estoque. Reposição incrementa a quantidade
6. **Admin:**
    - Faça um painel administrativo com django-admin para gestão dos projetos pela Web
7. **Autenticação:**
    - Ative autenticação com Token JWT
8. **Busca, Filtro, Ordenação, Paginação**
    - Adicione esses recursos em sua API REST
9. **Docs Swagger:**
    - Ative Swagger no seu projeto DRF com o pacote: **drf-yasg**
10. **Insomnia/Postman/Equivalente:**
    - Faça suas consulta via cliente HTTP

Avançando (Opcional):

- Perfil de Usuário: tenha dois perfis de usuário, um que é o vendedor, que pode cadastrar e vender produtos. E outro cliente, que pode realizar apenas compras de produtos

Entrega:

- O código deve ser entregue até a data combinada no formato de link para um repositório Git ou arquivo ZIP contendo o projeto pelo Classroom
- Deploy Rende.com (Opcional)

Critérios de Avaliação:

- Organização do código.
- Implementação correta do CRUD.
- Uso adequado das técnicas