

Aluno: Gabriel Pires de Campos Rezende

Aluno: Giancarlo Moraes de Sousa

Descrição do framework Django

Introdução

Django, um dos frameworks de desenvolvimento web mais populares e robustos escritos em Python, é conhecido por sua filosofia de "baterias incluídas", o que significa que vem com uma série de funcionalidades integradas prontas para uso. Projetado para facilitar o desenvolvimento de aplicações web de forma rápida e limpa, Django segue o padrão de design arquitetônico Model-View-Template (MVT), uma variação do Model-View-Controller (MVC).

O framework em questão foi escolhido para desenvolvimento do projeto final da disciplina. Tal escolha foi feita baseada na proficiência da dupla na tecnologia e também pela agilidade trazida pelo framework quanto ao desenvolvimento de aplicações web. O Django além de oferecer um poderoso recurso de ORM (Object-Relational Mapping), também é utilizado como servidor para executar a aplicação web.

Princípios e Filosofia do Django

- **Princípio DRY (Don't Repeat Yourself):** Este princípio está no cerne do Django, incentivando a reutilização de código e evitando duplicações.
- **Desenvolvimento Rápido:** Django foi projetado para mover projetos do conceito à conclusão o mais rápido possível.
- **Componentes Desacoplados:** Cada parte do framework funciona independentemente da outra, o que proporciona uma grande flexibilidade.
- **Integração com Banco de Dados:** Django vem com um ORM embutido, que permite aos desenvolvedores interagir com o banco de dados usando Python, em vez de SQL.

Gerenciamento de Banco de Dados

O gerenciamento do banco de dados é feito pelo Django através das migrações, que gerenciam as mudanças no esquema do banco de dados - seja criando novas tabelas, modificando campos existentes, ou até mesmo removendo tabelas. Este sistema de migrações torna o processo de alteração do banco de dados muito mais gerenciável e menos propenso a erros.

Models

Os modelos no Django são classes Python que definem a estrutura dos dados (campos e comportamentos) e também representam tabelas no banco de dados. Cada modelo é mapeado para uma tabela, e cada campo do modelo é uma coluna na tabela. Desse modo, a definição de tabelas a nível de banco de dados e classes de negócio ficam unificadas em um só lugar, minimizando alterações em caso de adição de campos, restrição em campos, entre outros.

Realização de Queries

O ORM do Django permite a realização de queries de maneira intuitiva e poderosa. A API de consultas do Django abstrai a necessidade de escrever consultas SQL cruas, permitindo aos desenvolvedores realizar operações de banco de dados complexas de maneira simples e eficaz. Através do ORM é possível especificar quais campos devem ser retornados, qual a ordenação, retornar consultas como lista de objetos, dicionários, entre outras flexibilidades trazidas.

Templates e Contexto

- **Templates:** São arquivos HTML que permitem a inserção de dados dinâmicos e lógica de programação. Eles são responsáveis pela camada de apresentação na arquitetura MVT do Django.
- **Contexto:** As views no Django passam informações para os templates através de um "contexto". O contexto é um dicionário em Python que mapeia nomes de variáveis para objetos Python. Quando uma view renderiza um template, ela passa este contexto para o template. O sistema de templates do Django então processa o arquivo HTML, substituindo as variáveis de template pelos valores correspondentes do contexto.

Integração com Front-end e o Uso de Templates

O framework oferece a possibilidade de templates HTML serem renderizados como resposta de uma requisição em uma determinada view. Através dessa funcionalidade, uma view pode renderizar um template repassando consigo informações de contexto que podem ser acessadas no HTML como variáveis com funções embutidas que auxiliam na manipulação desses dados no front-end.

Contudo, considerando a crescente demanda por aplicações que utilizam arquitetura de microsserviços, o framework também permite a construção de APIs RESTful, removendo de si a responsabilidade de renderizar o front-end das aplicações.

Conclusão

Em resumo, Django oferece uma solução abrangente e integrada para o desenvolvimento web, cobrindo desde o gerenciamento de banco de dados até a renderização de interfaces de usuário no navegador. Com seu princípio de "baterias incluídas", o Django se destaca como uma ferramenta poderosa para desenvolvedores que

desejam construir aplicações web sofisticadas e eficientes com um mínimo de esforço e tempo.