Criando uma sala de estudos virtual com compartilhamento e edição de documentos em tempo real, videochamada e chat utilizando o Django, Sockets e Threads.

Resumo de conteúdo

Estudo de caso número 1: Introdução ao framework Django por meio da criação de uma aplicação de player de música.

Observações:

O desenvolvimento e execução de todas as partes do projeto serão feitas em sistemas *nix. Linhas que começam com ‘#’ são apenas comentários que explicam o que um determinado comando faz.

O que é Django?

Django é um framework utilizado para desenvolver aplicações web com eficiência, segurança e performance utilizando a linguagem Python. O Django fornece uma API completa para manipulação de bancos de dados, criação de modelos, configuração e gerênciamento de servidores etc.

Criação de projetos com o Django

Passo 1: Criando ambientes virtuais

Antes de iniciar qualquer projeto utilizando o framework Django, é recomendado criar antes um ambiente virtual para o gerênciamento de pacotes. Um ambiente virtual permite que pacotes sejam instalados e gerênciados para um único projeto em específico, assim, evita-se o risco de conflitos com pacotes da instalação de sistema da linguagem Python.

O primeiro passo é criar um diretório para o nosso projeto. Para isso, abra uma janela de terminal e digite:

mkdir music_player

e depois acesse o novo diretório criado. Para criar um ambiente virtual dentro do diretório do projeto, basta digitar:

python -m venv <nome_do_ambiente_virtual>

Por exemplo, vamos criar um projeto denominado music_player. O comando mostrado acima terá a seguinte aparência:

python -m venv music_player

É interessante utilizar uma convensão de nomenclatura para ambientes virtuais, colocando ‘env’ ao final do nome do ambiente. Isso deixará claro que um diretório em específico é um ambiente virtual. Eis a aplicação dessa nomenclatura no exemplo anterior:

```
python -m venv music_player_env
```

e com isso temos o nosso ambiente virtual criado.

Passo 2: Ativando ambientes virtuais

Apenas criar um ambiente virtual não é o suficiente. Para utilizar o seu ambiente virtual, antes é preciso ativá-lo. Você pode ativar um ambiente virtual executando o arquivo ‘activate’. Na mesma janela de terminal do primeiro passo, digite:

```
source music_player_env/bin/activate
```

Esse comando fará com que o seu ambiente virtual seja ativado. Como confirmação de que tudo ocorreu bem, você deverá ver o nome do seu ambiente virtual no canto superior à esquerda, dentro de parêntesis. Eis um exemplo completo que inclui os passos 1 e 2.

```
# Cria o diretório do projeto
gabriel-home@gabrielhome:~$ mkdir music_player

# Acessa o diretório criado
gabriel-home@gabrielhome:~$ cd music_player

# Cria o ambiente virtual music_player_env
gabriel-home@gabrielhome:~/music_player$ python -m venv music_player_env

# Ativa o ambiente virtual music_player_env
gabriel-home@gabrielhome:~/music_player$ source music_player_env/bin/activate

# Resultado
(music_player_venv) gabriel-home@gabrielhome:~/music_player$
```

Caso o resultado obtido não tenha sido o mesmo, exclua o seu projeto digitando:

```
# Remove o diretório criado para o projeto
gabriel-home@gabrielhome:~$ cd .. && rm -r music_player
```

e retorne ao passo inicial. Verifique se está digitando todos os comandos da forma correta.

Passo 3: Instalando o Django em um ambiente virtual

Com o seu ambiente virtual ativo(confira se ele está entre parêntesis à esquerda do seu terminal), basta utilizar o gerênciador de pacotes da linguagem Python, o PIP, para instalar o Django no seu ambiente virtual:

 # Instala o django no ambiente virtual
(music_player_venv) gabriel-home@gabrielhome:~$ pip install django

Criando o projeto music_player

Após instalar o Django, com o seu ambiente virtual ativo, digite o seguinte comando para criar o projeto music_player:

 # Cria o projeto music_player_proj
(music_player_venv) gabriel-home@gabrielhome:~$ django-admin.py startproject music_player_project .

OBS: Utilizar o '.' fará com que o Django crie uma estrutura de diretórios mais simples e fácil de gerênciar. O que é útil quando for configurar URLs de aplicações ou implantar o seu projeto em um servidor. Caso contrário, o seu projeto terá uma estrutura de diretórios mais complexa para manipulação. Aqui mais uma vez utilizamos uma convensão, 'proj', para indicar que nesse diretório está contido os arquivos de configuração do projeto.

Após a execução desse comando, você deverá visualizar os seguintes arquivos e diretórios:

manage.py music_player_env music_player_proj

Criando o banco de dados do projeto

Com o projeto iniciado, o próximo passo será criar o banco de dados do projeto. O Django fornece uma API muito completa para a manipulação de um banco de dados SQL. O Django armazena todos os dados do projeto em um único arquivo SQL. Para criar o banco de dados, digite:

 # Cria o banco de dados do projeto
(music_player_venv) gabriel-home@gabrielhome:~$ python manage.py migrate

Após executar esse comando, você deverá ser capaz de ver o arquivo do banco de dados do projeto com a extensão .sqlite3:

db.sqlite3 manage.py music_player_env music_player_proj

O arquivo *manage.py* é utilizado para repassar comandos específicos para o Django, que é responsável pela execução dos mesmos. Nesse caso, o comando passado como argumento para *manage.py* foi *migrate.* A operação de alterar um banco de dados é denominada "migrar o banco de dados". Como foi a primeira vez que esse comando foi executado, a única coisa que o Django fez foi atualizar o banco de dados para estar de acordo o estado atual do projeto.

Criando a aplicação music_player_app

Um projeto Django é constituído por um conjunto de aplicações que, quando executadas juntas, formam a estrutura de todo o projeto. Para criar a aplicação *music_player_app*, digite o comando a seguir:

 # Cria a aplicação music_player_app
(music_player_venv) gabriel-home@gabrielhome:~$ python manage.py startapp music_player_app

Após executar esse comando, a estrutura de arquivos do seu projeto se parecerá com:

db.sqlite3 manage.py music_player_app music_player_env music_player_proj

Inserindo a aplicação music_player_app no projeto

O próximo passo é adicionar a nova aplicação no arquivo de configuração do django, que está contido no seguinte path:

music_player_proj/settings.py

Após abrir este arquivo, procure pela lista denominada INSTALLED_APPS e adicione o nome da sua aplicação, nesse caso, *music_player_app*. Eis um exemplo:

```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'music_player_app',
]
```

Pronto, temos o projeto configurado e com a aplicação instalada. O próximo passo será criar uma conta no site de administração do Django e começar a criar os primeiros modelos(dados do projeto). Tudo isso ficará para o próximo resumo.

Sugestões e erratas enviar para:
santosgabriel.rochamarcos@gmail.com

Autor: Marcos Gabriel Santos Rocha