

Manual de Preparação do Ambiente Django para Criar uma API

Introdução

Este manual tem como objetivo guiar alunos iniciantes na preparação do ambiente para desenvolvimento de uma API utilizando o framework Django. Vamos abordar desde a instalação das ferramentas necessárias até a criação dos diretórios e arquivos iniciais do projeto.

Pré-requisitos

- Python instalado (versão 3.6 ou superior)
- Pip (gerenciador de pacotes do Python)
- Virtualenv (para criação de ambientes virtuais)

Passo a Passo

1. Instalando o Python

Certifique-se de que o Python está instalado em seu sistema. Você pode baixar a versão mais recente do Python no [site oficial](#).

2. Instalando o Pip

O Pip geralmente é instalado junto com o Python. Para verificar se o Pip está instalado, execute o seguinte comando no terminal:

```
pip --version
```

Se não estiver instalado, siga as instruções no [site oficial do Pip](#).

3. Instalando o Virtualenv

O Virtualenv é uma ferramenta para criar ambientes virtuais isolados. Para instalá-lo, execute:

```
pip install virtualenv
```

4. Criando um Ambiente Virtual

Crie um diretório para o seu projeto e navegue até ele:

```
mkdir meu_projeto_django  
cd meu_projeto_django
```

Crie um ambiente virtual:

```
virtualenv venv
```

Ative o ambiente virtual:

- No Windows:

```
venv\Scripts\activate
```

- No macOS/Linux:

```
source venv/bin/activate
```

5. Instalando o Django

Com o ambiente virtual ativado, instale o Django:

```
pip install django
```

6. Criando o Projeto Django

Crie um novo projeto Django:

```
django-admin startproject meu_projeto .
```

7. Estrutura de Diretórios

A estrutura inicial do projeto será semelhante a esta:

```
meu_projeto_django/  
├── manage.py  
├── meu_projeto/  
│   ├── __init__.py  
│   ├── settings.py  
│   ├── urls.py  
│   ├── asgi.py  
│   └── wsgi.py
```

8. Criando um Aplicativo Django

Dentro do diretório do projeto, crie um novo aplicativo:

```
python manage.py startapp meu_app
```

Aula 01: Configurando o Projeto Django

1. Registrando o Aplicativo no Projeto

Abra o arquivo `settings.py` localizado no diretório `meu_projeto` e adicione o nome do aplicativo na lista `INSTALLED_APPS`:

```
INSTALLED_APPS = [  
    ...  
    'meu_app',  
]
```

2. Configurando URLs do Projeto

Abra o arquivo `urls.py` no diretório `meu_projeto` e inclua a rota para o aplicativo:

```
from django.contrib import admin  
from django.urls import path, include  
  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('meu_app/', include('meu_app.urls')),  
]
```

3. Criando URLs do Aplicativo

No diretório `meu_app`, crie um arquivo `urls.py` e defina as rotas do aplicativo:

```
from django.urls import path  
from . import views  
  
urlpatterns = [  
    path('', views.index, name='index'),  
]
```

4. Criando a View Inicial

No diretório `meu_app`, abra o arquivo `views.py` e crie uma view simples:

```
from django.http import HttpResponse

def index(request):
    return HttpResponse("Olá, mundo! Este é o meu primeiro aplicativo Django.")
```

5. Executando o Servidor de Desenvolvimento

Para verificar se tudo está funcionando corretamente, execute o servidor de desenvolvimento:

```
python manage.py runserver
```

Acesse http://127.0.0.1:8000/meu_app/ no seu navegador e você deverá ver a mensagem "Olá, mundo! Este é o meu primeiro aplicativo Django."

Parabéns! Você configurou com sucesso seu primeiro projeto Django com um aplicativo básico.