



CONFIGURANDO UM PROJETO DJANGO REST FRAMEWORK (DRF)

Aprenda um passo a passo para configurar um projeto Django Rest Framework e começar a desenvolver APIs REST.

Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



Deixe que nossa IA faça o trabalho pesado

 Syntax Highlight

 Adicione Banners Promocionais

 Edite em Markdown em Tempo Real

 Infográficos feitos por IA

TESTE AGORA 

Salve salve Pythonista!

Neste artigo, abordaremos a configuração de um projeto Django Rest Framework (DRF), incluindo a configuração do arquivo `requirements.txt` e do próprio Django Rest Framework.

Entender como configurar um projeto com DRF é essencial para quem deseja construir APIs robustas e escaláveis utilizando o Django, que é um dos frameworks mais populares para desenvolvimento web em Python.

Configurando o Django Rest Framework corretamente garantirá que seu projeto esteja pronto para adicionar, gerenciar e expor dados por meio de APIs RESTful.

Passo 1: Criando e configurando um novo projeto Django

Primeiro, precisamos instalar o Django e criar um novo projeto.

Certifique-se de ter o Python e o `pip` instalados em seu sistema.

Crie um novo ambiente virtual e instale as dependências.

Se ainda não sabe como criar ambientes virtuais para seus projetos Python, **PRECISA** ler esse artigo agora: [Python e Virtualenv: como programar em Ambientes Virtuais](#)

Com um ambiente virtual criado e ativado:

```
pip install django
django-admin startproject meu_projeto
cd meu_projeto
python manage.py startapp minha_app
```

Isso irá instalar o Django, iniciar um novo projeto com `startproject` e criar um novo App Django chamado `minha_app`.

Passo 2: Instalar o Django Rest Framework

Em seu recente criado ambiente virtual, instale o Django Rest Framework utilizando `pip`.

Adicione-o ao seu projeto Django configurando seu arquivo `requirements.txt`.

```
pip install djangorestframework
```

Crie um arquivo `requirements.txt` na raiz do seu projeto e adicione as dependências necessárias.

```
django
djangorestframework
```

Você pode instalar todas as dependências do `requirements.txt` com o comando:

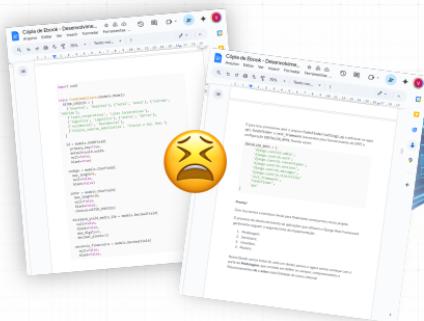
```
pip install -r requirements.txt
```

Projeto Django na prática: O **DevBook** é um projeto real construído com Django. Uma plataforma que gera ebooks técnicos com IA, processamento assíncrono e exportação em PDF. Quer ver como ficou? Dá uma olhada!



Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



Deixe que nossa IA faça o trabalho pesado

Syntax Highlight

Adicione Banners Promocionais

Edite em Markdown em Tempo Real

Infográficos feitos por IA

TESTE AGORA! PRIMEIRO CAPÍTULO 100% GRÁTIS

Passo 3: Configurando o Django Rest Framework

Abra o arquivo `settings.py` do seu projeto Django e adicione '`rest_framework`' à lista de aplicativos instalados:

```
INSTALLED_APPS = [
    ...
    'rest_framework',
    'minha_app',
]
```

Pronto, com isso você estará pronto para iniciar o **desenvolvimento de APIs REST com Django REST Framework!**

Fique ligado pois nos próximos artigos vamos falar sobre a configuração de rotas da sua API, como criar `Viewsets` para expôr dados, como criar `Serializers` para converter seus dados de objetos Python para representação JSON e vice-versa e muito mais!

Conclusão

Neste artigo, vimos como configurar um projeto Django Rest Framework.

Configuramos o arquivo `requirements.txt`, instalamos e configuramos o Django Rest Framework para criarmos APIs REST.

Com essas configurações, você está preparado para desenvolver APIs RESTful robustas utilizando Django e DRF!

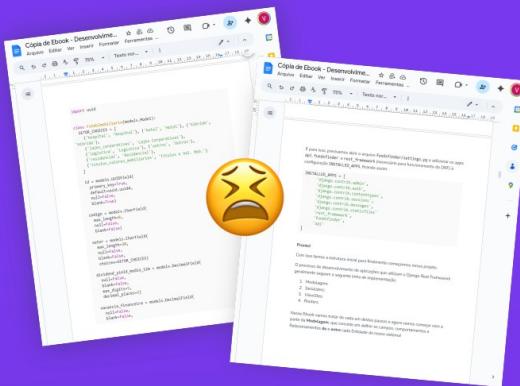
Fique ligado nos próximos artigos!

Até lá 



Crie Ebooks técnicos em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



Arquitetura de Software Moderna

A arquitetura de software alvo é profissional contendo o e-mail e produções de software para arquiteturas modernas. Oferece recursos como: códigos de software gerados automaticamente.

```
import python
import python

class Arquitetura_de_Software_Moderna:
    ...
    def share(self):
        pass
    ...
    return "Arquitetura de Mod", "arquitetura_mod"
}

def __init__(self):
    if user.username == "admin":
        self.user = "admin"
    else:
        self.user = "user"

    self.__dict__.update(locals())
    del self.__dict__["self"]

    # Criação de uma nova classe
    self.__dict__[user] = self.__class__(**self.__dict__)
    self.__dict__[user].__dict__.update(locals())
    self.__dict__[user].__dict__.pop(user)

    # Criação de uma nova classe
    self.__dict__[user] = self.__class__(**self.__dict__)
    self.__dict__[user].__dict__.update(locals())
    self.__dict__[user].__dict__.pop(user)
```

AI-generated system

A arquitetura com propósito é a arquitetura moderna. Seus componentes principais incluem o sistema centralizado, os sistemas periféricos e os sistemas de suporte. O sistema centralizado é responsável por gerenciar todos os recursos do sistema, enquanto os sistemas periféricos fornecem suporte para o sistema centralizado. Os sistemas de suporte fornecem suporte para o sistema centralizado.

Clean layout

O layout é limpo e organizado, facilitando a leitura e compreensão do código gerado.



</> Syntax Highlight

Infográficos feitos por IA

Adicione Banners Promocionais

Deixe que nossa IA faça o trabalho pesado

Edite em Markdown em Tempo Real

TESTE AGORA



PRIMEIRO CAPÍTULO 100% GRÁTIS