**Passo 1: Instalar Dependências**

Primeiro, você precisa instalar o Django e o Django Rest Framework. Vou assumir que você já tem o Python e o pip instalados.

1. Crie e ative um ambiente virtual (opcional, mas recomendado):

Um ambiente virtual é uma ferramenta que ajuda a criar um espaço isolado para um projeto Python, onde você pode instalar pacotes e dependências específicos sem afetar outros projetos ou a instalação global do Python no seu sistema.

Terminal

python -m venv myenv

1. Instale o Django e o Django Rest Framework:

Terminal

pip install django djangorestframework

**Passo 2: Criar um Novo Projeto Django**

1. Crie um novo projeto Django:

Terminal

django-admin startproject myproject

cd myproject

1. Crie um novo aplicativo dentro do projeto:

Terminal

python manage.py startapp myapp

**Passo 3: Configurar o Projeto para Usar o Django Rest Framework**

1. Adicione rest\_framework e o aplicativo recém-criado (myapp) ao INSTALLED\_APPS no arquivo myproject/settings.py:

python

INSTALLED\_APPS = [

...

'rest\_framework',

'myapp',

]

**Passo 4: Criar um Modelo e Serializador**

1. Defina um modelo no arquivo myapp/models.py. Por exemplo, um modelo simples de Item:

python

from django.db import models

class Item(models.Model):

name = models.CharField(max\_length=100)

description = models.TextField()

def \_\_str\_\_(self):

return self.name

1. Crie um serializador para o modelo Item em myapp/serializers.py:

python

from rest\_framework import serializers

from .models import Item

class ItemSerializer(serializers.ModelSerializer):

class Meta:

model = Item

fields = '\_\_all\_\_'

**Passo 5: Criar uma View e Rota para a API**

1. Crie uma view para a API em myapp/views.py:

python

from rest\_framework import viewsets

from .models import Item

from .serializers import ItemSerializer

class ItemViewSet(viewsets.ModelViewSet):

queryset = Item.objects.all()

serializer\_class = ItemSerializer

1. Adicione as rotas para a API em myapp/urls.py:

python

from django.urls import path, include

from rest\_framework.routers import DefaultRouter

from .views import ItemViewSet

router = DefaultRouter()

router.register(r'items', ItemViewSet)

urlpatterns = [

path('', include(router.urls)),

]

1. Inclua as rotas do myapp no arquivo myproject/urls.py:

python

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('api/', include('myapp.urls')),

]

**Passo 6: Migrar o Banco de Dados e Executar o Servidor**

1. Execute as migrações para criar as tabelas no banco de dados:

Terminal

python manage.py makemigrations

python manage.py migrate

1. Crie um superusuário para acessar o admin do Django (opcional):

Terminal

python manage.py createsuperuser

1. Inicie o servidor de desenvolvimento:

Terminal

python manage.py runserver

**Testar a API**

Agora, você pode acessar a API no navegador em http://127.0.0.1:8000/api/items/. Você pode adicionar, editar e visualizar itens usando as rotas fornecidas pelo ItemViewSet.