



TUTORIAL I

DJANGO REST FRAMEWORK



INTRODUÇÃO: CONFIGURANDO AMBIENTE

- `mkdir meuprojeto`
- `cd meuprojeto`
- `python3 -m venv env`
- `source env/bin/activate`
- `pip install django djangorestframework pygments`
- `django-admin startproject tutorial`
- `cd tutorial`
- `python manage.py startapp snippets`

PASSO I

- Modificar o settings.py para incluir as APPs : snippets e rest_framework
- Lá no settings incluir:

```
INSTALLED_APPS = [  
    ...  
    'rest_framework',  
    'snippets.apps.SnippetsConfig',  
]
```

PASSO II: CRIAR UM MODEL EM MODELS.PY

```
from django.db import models
from pygments.lexers import get_all_lexers
from pygments.styles import get_all_styles

LEXERS = [item for item in get_all_lexers() if item[1]]
LANGUAGE_CHOICES = sorted([(item[1][0], item[0]) for item in LEXERS])
STYLE_CHOICES = sorted([(item, item) for item in get_all_styles()])

class Snippet(models.Model):
    created = models.DateTimeField(auto_now_add=True)
    title = models.CharField(max_length=100, blank=True, default='')
    code = models.TextField()
    linenos = models.BooleanField(default=False)
    language = models.CharField(choices=LANGUAGE_CHOICES, default='python', max_length=100)
    style = models.CharField(choices=STYLE_CHOICES, default='friendly', max_length=100)

    class Meta:
        ordering = ['created']
```

PASSOII: CONTINUANDO

- `python manage.py makemigrations snippets`
- `python manage.py migrate`

PASSO III : CRIANDO SERIALIZERS

- É preciso dizer “como” vamos converter o modelo para JSON, como cria um novo dado, como atualiza um existente
- Isso é feito, nessa API, definindo classes chamadas de SERIALIZERS
- Existem várias opções (<https://www.django-rest-framework.org/api-guide/serializers/>)
- A mais fácil delas é usar o ModelSerializer
- O ModelSerializer define basicamente que os campos em JSON são os mesmo do model que nós criamos.
- Ele define também que o comportamento de criação/atualização para um dado no formato JSON é mosmo de uma instância de um model.

PASSO III : CRIANDO SERIALIZERS....

- Crie um arquivo (no mesmo nível do models.py) chamado serializers.py

```
from rest_framework import serializers
from .models import Snippet
class SnippetSerializer(serializers.ModelSerializer):
    class Meta:
        model = Snippet
        fields = ['id', 'title', 'code', 'linenos', 'language', 'style']
```

PASSO IV : CRIAR AS VIEWS

- Editar o arquivo views.py

```
from django.http import HttpResponse, JsonResponse
from django.views.decorators.csrf import csrf_exempt
from rest_framework.parsers import JSONParser
from .models import Snippet
from .serializers import SnippetSerializer
```


PASSO IV : CRIAR AS VIEWS

- Editar o arquivo views.py

@csrf_exempt # Aqui dizemos que não precisa autenticar com um token...apenas para efeito didático.

def snippet_list(request): #nome da view

""" Aqui nós queremos ou Listar todos os objetos snippets, ou criar um novo objeto snippet. """

if request.method == 'GET': # Se o pedido for do tipo GET

snippets = Snippet.objects.all()

serializer = SnippetSerializer(snippets, many=True)

return JsonResponse(serializer.data, safe=False)

elif request.method == 'POST': # Se for do tipo POST

data = JSONParser().parse(request)

serializer = SnippetSerializer(data=data)

if serializer.is_valid():

serializer.save()

return JsonResponse(serializer.data, status=201)

return JsonResponse(serializer.errors, status=400)

PASSO IV : CRIAR AS VIEWS

- Precisamos também de viws que retornem ou alterem um objeto

```
@csrf_exempt
def snippet_detail(request, pk):
    """ Retrieve, update or delete a code snippet. """
    try:
        snippet = Snippet.objects.get(pk=pk)
    except Snippet.DoesNotExist:
        return HttpResponse(status=404)

    if request.method == 'GET':
        serializer = SnippetSerializer(snippet)
        return JsonResponse(serializer.data)
    elif request.method == 'PUT':
        data = JSONParser().parse(request)
        serializer = SnippetSerializer(snippet, data=data)
        if serializer.is_valid():
            serializer.save()
```

PASSO IV : CRIAR AS VIEWS

- Precisamos também de viws que retornem ou alterem um objeto

#Continuando

```
elif request.method == 'PUT':  
    data = JSONParser().parse(request)  
    serializer = SnippetSerializer(snippet, data=data)  
    if serializer.is_valid():  
        serializer.save()  
        return JsonResponse(serializer.data)  
    return JsonResponse(serializer.errors, status=400)  
elif request.method == 'DELETE':  
    snippet.delete()  
    return HttpResponse(status=204)
```

PASSO V : CONECTANDO AS VIEWS (URLS)

- Precisamos criar o snippets/urls.py

```
from django.urls import path
from snippets import views
```

```
urlpatterns = [
    path('snippets/', views.snippet_list),
    path('snippets/<int:pk>', views.snippet_detail),
]
```

- Edite criar o tutorial/urls.py

```
from django.urls import path, include
```

```
urlpatterns = [
    path("", include('snippets.urls')),
]
```

AGORA É SÓ RODAR!

- Teste seu projeto usando postman
 - <http://127.0.0.1:8000/snippets/> (lista todos os snippets)
 - <http://127.0.0.1:8000/snippets/2/> (lista o snippet “2” se ele existir!!)
 - Tente fazer um POST VOCE MESMO!!!