@fpftech.educacional

FPFtech
Escola Tecnológica

FPFtech
Escola Tecnológica

Serializers

```
from rest_framework import serializers

from escola import models


class ClienteSerializer(serializers.ModelSerializer):
    id = serializers.CharField(read_only=True) <- somente GET
    name = serializers.CharField(max_length=70)
    age = serializers.IntegerField(min_value=18, max_value=100)<- min=18 e
                                                                  max=100

    class Meta:
        model = models.Client
        fields = ['id', 'name', 'age', 'created_at']<- só vai retornar esses
                                                       campos...

                                                       '__all__' = retorna todos
```

validando campos:
id, name, age

Serializers

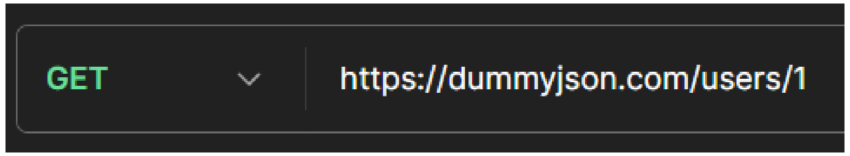
```
from rest_framework import serializers
from escola.models import Sale
```

```
class SaleSerializer(serializers.ModelSerializer):
    product = ProductSerializer(read_only=True)
    product_id = serializers.PrimaryKeyRelatedField(queryset=Product.objects.all(), source='product', write_only=True)
    client = ClientSerializer(read_only=True)
    client_id = serializers.PrimaryKeyRelatedField(queryset=Client.objects.all(), source='client', write_only=True)
    employee = EmployeeSerializer(read_only=True)
    employee_id = serializers.PrimaryKeyRelatedField(queryset=Employee.objects.all(), source='employee', write_only=True)

    class Meta:
        model = Sale
        fields = ['id', 'product_id', 'product', 'client_id', 'client', 'employee_id', 'employee', 'nrf']
```

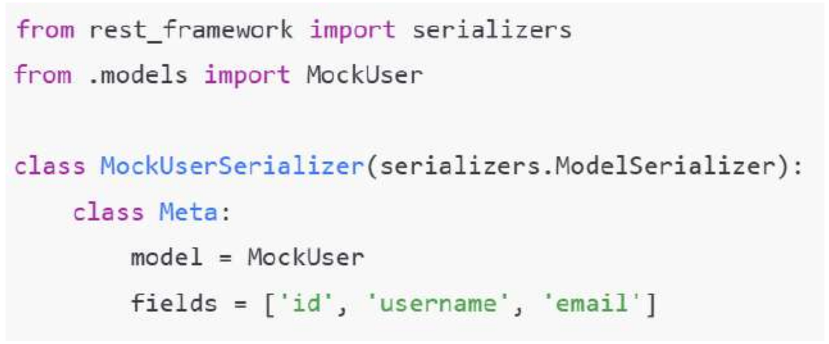
GET, em vez de mostrar somente o id_produto, ele mostra <- todos os Fields do Serializer Produt, obj em JSON/XML

<- POST, não precisa passar todo o obj produto, somente dê o id que ele vai procurar e atrelar o pk

Ciclo de Requests



GET https://dummyjson.com/users/1



```python
from rest_framework import serializers
from .models import MockUser

class MockUserSerializer(serializers.ModelSerializer):
    class Meta:
        model = MockUser
        fields = ['id', 'username', 'email']
```

request

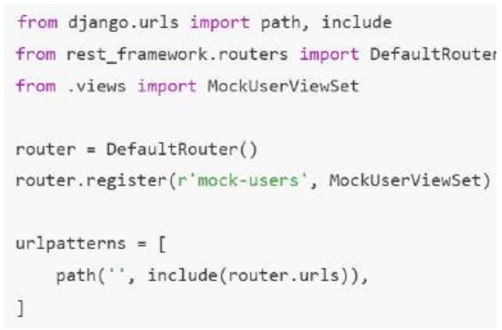
endpoint
urls

viewsets

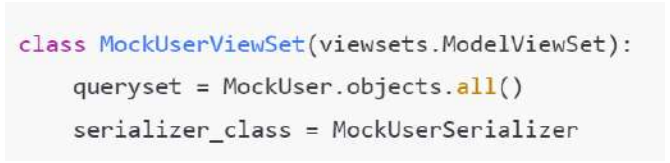
Serializers
JSON=Django

models

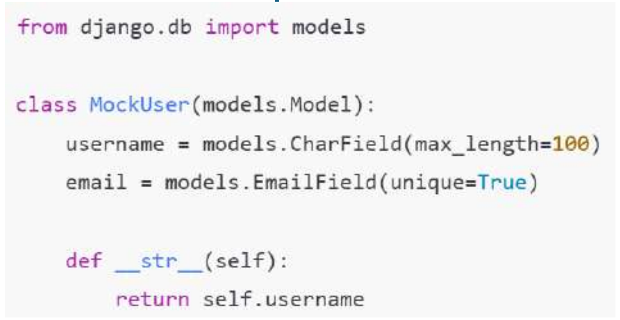
database



```python
from django.urls import path, include
from rest_framework.routers import DefaultRouter
from .views import MockUserViewSet

router = DefaultRouter()
router.register(r'mock-users', MockUserViewSet)

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



```python
class MockUserViewSet(viewsets.ModelViewSet):
    queryset = MockUser.objects.all()
    serializer_class = MockUserSerializer
```

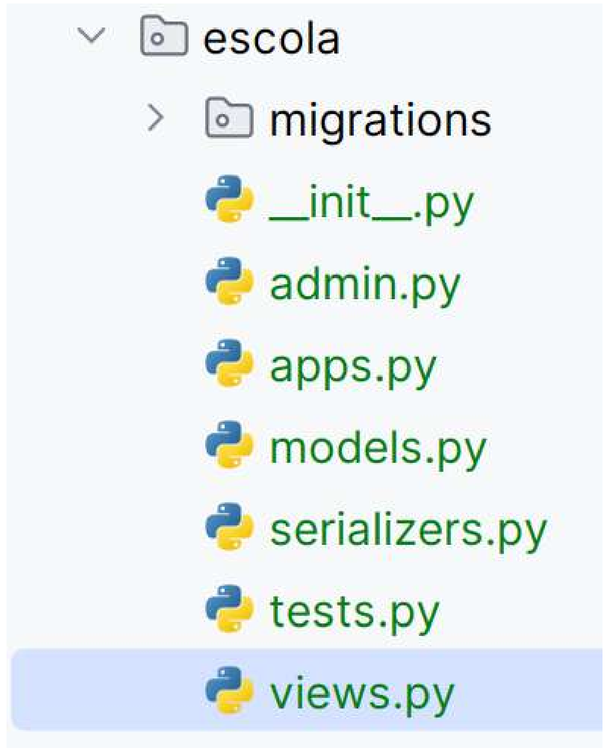


```python
from django.db import models

class MockUser(models.Model):
    username = models.CharField(max_length=100)
    email = models.EmailField(unique=True)

    def __str__(self):
        return self.username
```

Viewsets

Viewsets = criar classes responsáveis pelos nossos endpoints.

1) Vá para o arquivo views.py na sua aplicação.



escola
migrations
__init__.py
admin.py
apps.py
models.py
serializers.py
tests.py
views.py

2) Lá criaremos os endpoints que possuem 2 aspectos importantes:

queryset <- você definirá como será feito a busca
serializer_class <-você vai apontar pro serializer já criado

Viewsets

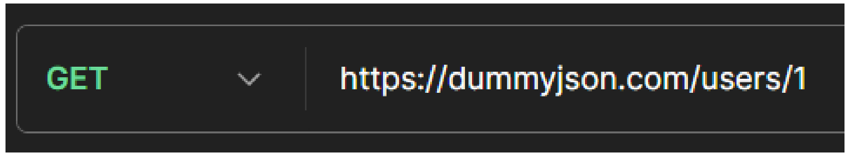
FPFtech
Escola Tecnológica

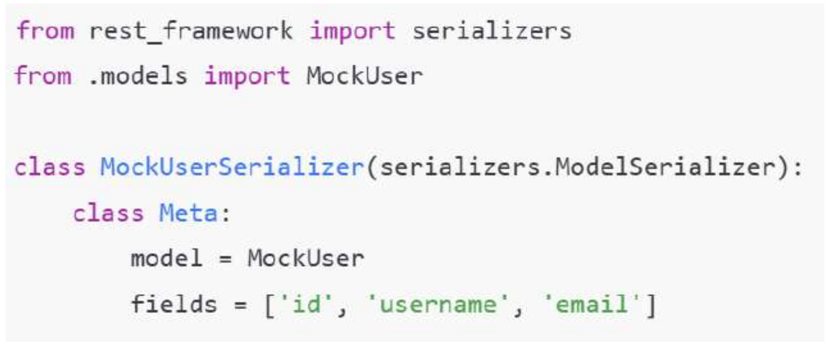
```
from rest_framework import viewsets
from escola.models import Client <- importamos os modelos necessários
from escola.serializers import ClientSerializer<- importamos os serialializers


class ClientViewSet(viewsets.ModelViewSet):
    queryset = Client.objects.all() <- queryset vai buscar todos os registros ‘all’
    serializer_class = ClientSerializer <- apontar pro serializer
```

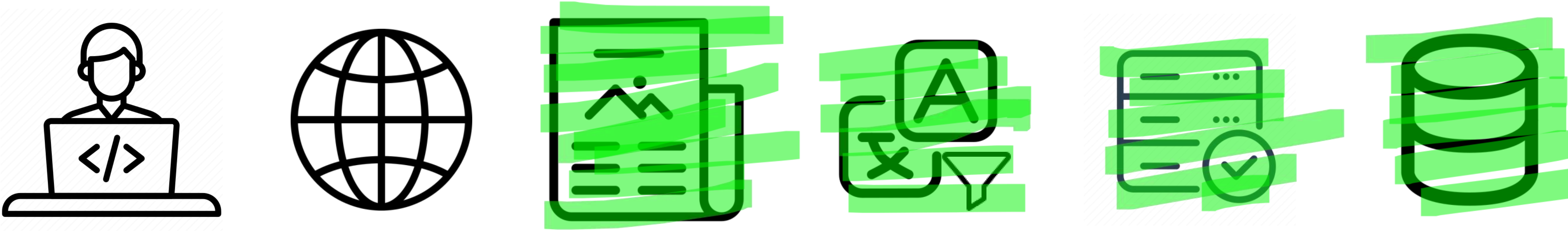
Fazer isso pros outros

Ciclo de Requests

FPFtech
Escola Tecnológica


GET https://dummyjson.com/users/1


```python
from rest_framework import serializers
from .models import MockUser

class MockUserSerializer(serializers.ModelSerializer):
    class Meta:
        model = MockUser
        fields = ['id', 'username', 'email']
```



request

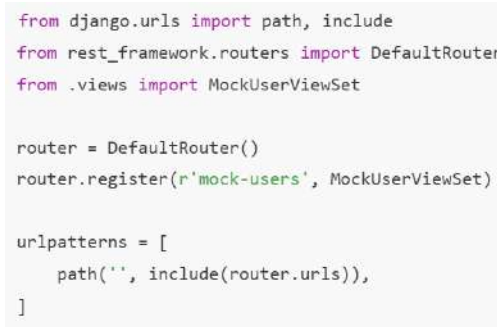
endpoint
urls

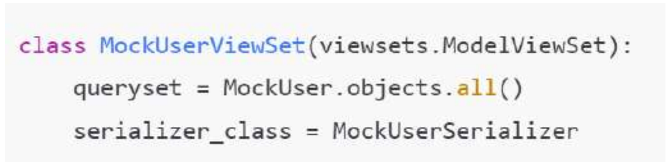
viewsets

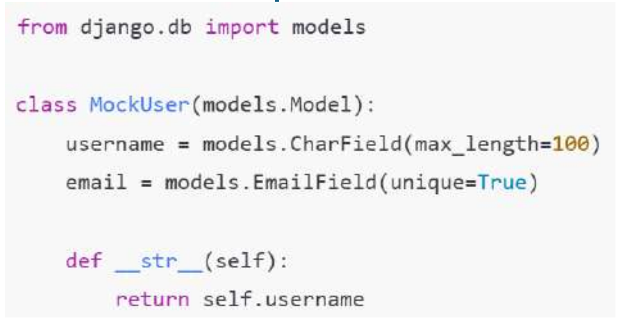
Serializers
JSON=Django

models

database


```python
from django.urls import path, include
from rest_framework.routers import DefaultRouter
from .views import MockUserViewSet

router = DefaultRouter()
router.register(r'mock-users', MockUserViewSet)

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


```python
class MockUserViewSet(viewsets.ModelViewSet):
    queryset = MockUser.objects.all()
    serializer_class = MockUserSerializer
```


```python
from django.db import models

class MockUser(models.Model):
    username = models.CharField(max_length=100)
    email = models.EmailField(unique=True)

    def __str__(self):
        return self.username
```

FPFtech
Escola Tecnológica

Obrigado!

FPFtech
Escola Tecnológica

@fpftech.educacional