

Django: criando uma API REST simples

No prompt de comando, estando na pasta onde o projeto django deve ser criado, execute:

```
django-admin startproject univesp [enter]
```

```
django-admin startapp alunos [enter]
```

```
cd univesp [enter]
```

```
python manage.py runserver [enter]
```

* Abra o navegador e acesse <http://127.0.0.1:8000> – se tudo ok verá a página inicial do Django

No prompt de comandos, tecla Ctrl+C para finalizar o servidor Django. E ainda no prompt de comandos digite o comando a seguir, para criar as tabelas de controle do Django no banco de dados:

```
python manage.py migrate [enter]
```

Deixe o prompt de comandos aberto pois será utilizado em breve.

As pastas do projeto são:

- **univesp**: pasta base do projeto
- **univesp\univesp**: pasta de controles do projeto
- **univesp\alunos**: pasta do aplicativo alunos

A partir de agora inicia-se a codificação em si. Para isso pode-se abrir a pasta do projeto no VS Code, Sublime Text, PyCharm, etc e usar estes para as edições de arquivos a seguir.

Arquivo settings.py – localizado dentro da pasta de controles do projeto

Localize no arquivo settings.py a tag `INSTALLED_APPS` e inclua `'rest_framework'` e o nome do aplicativo, no caso, `'alunos'`:

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'rest_framework',  
    'alunos'  
]
```

Arquivo models.py – localizado dentro da pasta do aplicativo alunos

```
from django.db import models
```

```
class Aluno(models.Model):  
    matricula      = models.CharField(max_length=14, null=False, blank=False)  
    nome           = models.CharField(max_length=50, null=False, blank=False)  
    sobrenome      = models.CharField(max_length=50, null=False, blank=False)  
    data_cadastro  = models.DateField(auto_now_add=True)
```

Dentro da pasta do aplicativo alunos, crie uma subpasta com nome de **api** e dentro dela crie os arquivos **serializers.py** e **viewsets.py**.

Arquivo serializers.py -----

```
from rest_framework import serializers  
from alunos import models
```

```
class AlunoSerializer(serializers.ModelSerializer):  
    class Meta:  
        model = models.Aluno  
        fields = '__all__'
```

Arquivo viewsets.py -----

```
from rest_framework import viewsets  
from alunos.api import serializers  
from alunos import models
```

```
class AlunoViewSet(viewsets.ModelViewSet):  
    serializer_class = serializers.AlunoSerializer  
    queryset = models.Aluno.objects.all()
```

Arquivo urls.py – localizado dentro da **pasta de controles do projeto** (*dentro da pasta do aplicativo também tem esse arquivo, mas aqui estamos pegando da pasta de controle*)

```
from django.contrib import admin  
from django.urls import path
```

```
from rest_framework import routers
```

```
from alunos.api import viewsets as alunoviewsets
```

```
route = routers.DefaultRouter()  
route.register(r'alunos/', alunoviewsets.AlunoViewSet, basename='Alunos')
```

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include(route.urls))  
]
```

De volta ao prompt de comandos, na **pasta base do projeto**:

- digite o comando a seguir para preparar as migrações (alterações em models.py) para o banco de dados:

```
python manage.py makemigrations [enter]
```

- digite o comando a seguir para que as migrações sejam efetivamente gravadas no banco de dados e a tabela aluno criada [no banco de dados]:

```
python manage.py migrate [enter]
```

Ative novamente o servidor Django:

```
python manage.py runserver [enter]
```

Abra o browser e acesse <http://127.0.0.1/8000/alunos>

A página da api deverá ser exibida.

NOTA: Criei o projeto usando Django versão 3.1.7 e Python versão 3.9.0