

# Back-End Frameworks

Aula 9 - 19/05/2023



[https://bit.ly/23\\_1\\_backend\\_m](https://bit.ly/23_1_backend_m)



## O que temos para **hoje**?

- Criação de API;
- Trabalho avaliativo;



# Criação de API

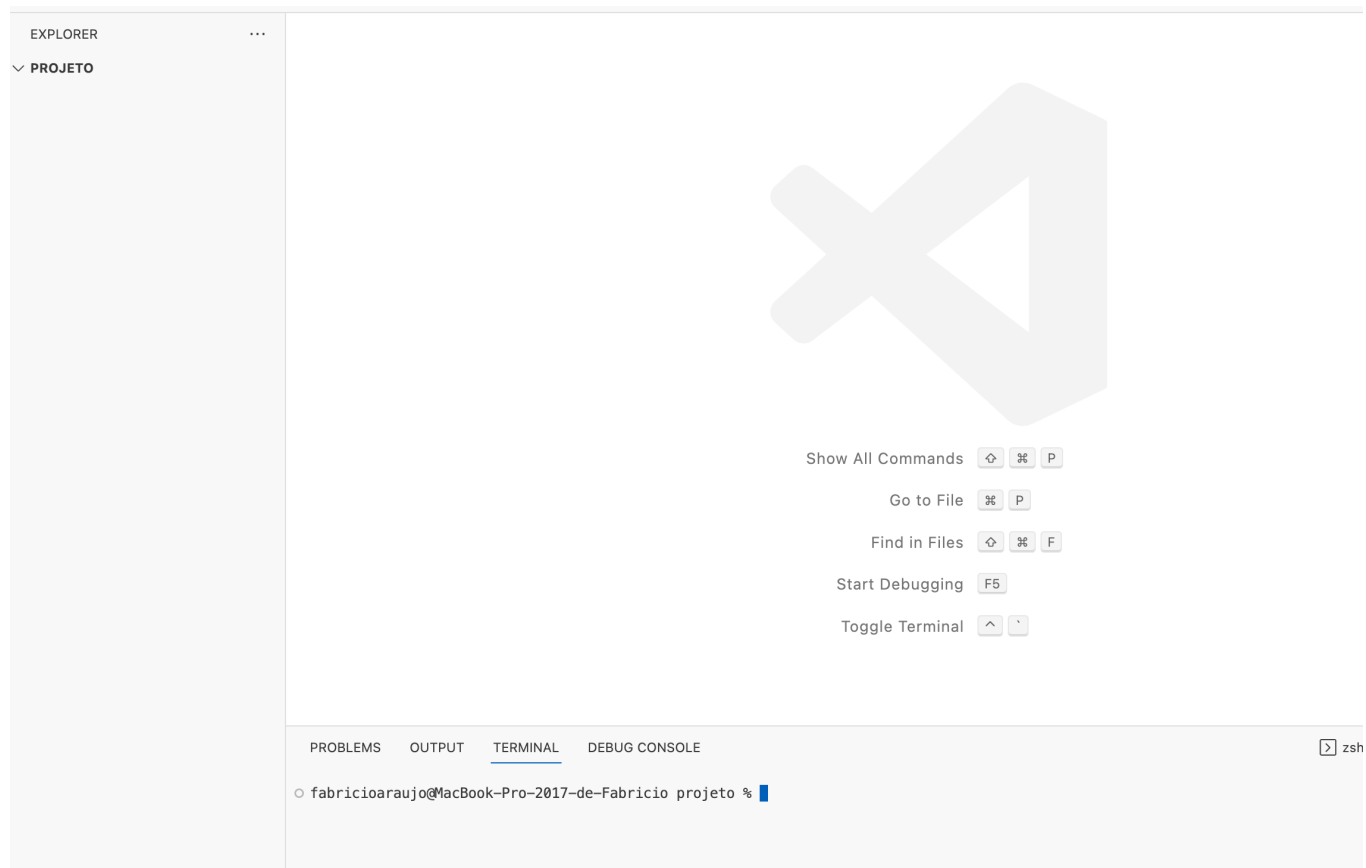


## Criação de API

- Criar uma pasta “projeto” e abrir esta pasta no Visual Studio;



## Criação de API





## Criação de API

- Digitar no terminal:  
`py -m pip install django`
- Depois:  
`py -m django --version`



## Criação de API

```
● fabricioaraujo@MacBook-Pro-2017-de-Fabricio projeto % python3 -m pip install django
Requirement already satisfied: django in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (4.1.3)
Requirement already satisfied: asgiref<4,>=3.5.2 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from django) (3.5.2)
Requirement already satisfied: sqlparse>=0.2.2 in /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages (from django) (0.4.2)
WARNING: You are using pip version 21.2.4; however, version 23.1.2 is available.
You should consider upgrading via the '/Library/Frameworks/Python.framework/Versions/3.10/bin/python3 -m pip install --upgrade pip' command.
● fabricioaraujo@MacBook-Pro-2017-de-Fabricio projeto % python3 -m django version
4.1.3
○ fabricioaraujo@MacBook-Pro-2017-de-Fabricio projeto %
```



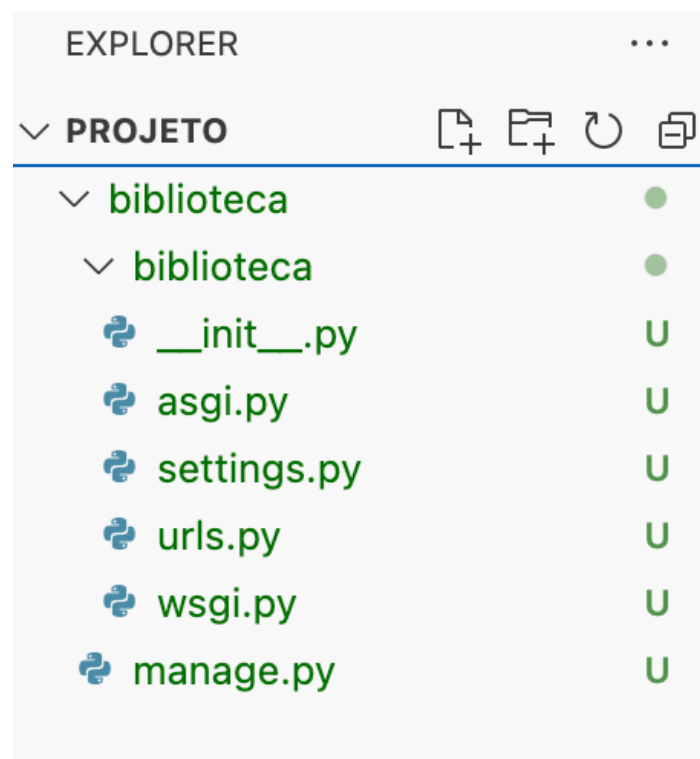
## Criação de API

- Digitar no terminal:  
`py -m django startproject biblioteca`





## Criação de API





## Criação de API

- Digitar no terminal:  
cd biblioteca



## Criação de API

- Digitar no terminal:  
`py -m django startapp livros`



## Criação de API

▼ biblioteca	●
▼ biblioteca	●
> __pycache__	●
+ __init__.py	U
+ asgi.py	U
+ settings.py	U
+ urls.py	U
+ wsgi.py	U
▼ livros	●
> __pycache__	●
> migrations	●
+ __init__.py	U
+ admin.py	U
+ apps.py	U
+ models.py	U
+ tests.py	U
+ views.py	U
≡ db.sqlite3	U
+ manage.py	U



## Criação de API

- Digitar no terminal:  
`py manage.py runserver`



## Criação de API

```
Watching for file changes with StatReloader
Performing system checks...
```

```
System check identified no issues (0 silenced).
```

```
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
```

```
Run 'python manage.py migrate' to apply them.
```

```
May 10, 2023 - 22:55:29
```

```
Django version 4.1.3, using settings 'biblioteca.settings'
```

```
Starting development server at http://127.0.0.1:8000/
```



## Criação de API

django

View [release notes](#) for Django 4.1



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.



## Criação de API

- Digitar no terminal:  
`py manage.py migrate`





## Criação de API

### Operations to perform:

Apply all migrations: admin, auth, contenttypes, sessions

### Running migrations:

```
Applying contenttypes.0001_initial... OK
Applying auth.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Applying admin.0003_logentry_add_action_flag_choices... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
Applying auth.0012_alter_user_first_name_max_length... OK
Applying sessions.0001_initial... OK
```



## Criação de API

- Na pasta biblioteca, abrir o arquivo “settings.py”;
- Buscar a linha 39 e deixar assim:



## Criação de API

```
33  ✓ INSTALLED_APPS = [  
34      'django.contrib.admin',  
35      'django.contrib.auth',  
36      'django.contrib.contenttypes',  
37      'django.contrib.sessions',  
38      'django.contrib.messages',  
39      'django.contrib.staticfiles',  
40      'livros'  
41  ]  
--
```



## Criação de API

- Na pasta livros, abrir o arquivo “views.py”;
- Deixar o arquivo assim:



## Criação de API

biblioteca > livros >  views.py > ...

```
1  from django.shortcuts import render
2
3  from django.http import HttpResponse
4  # Create your views here.
5  def index(request):
6      |   return HttpResponse("Olá, mundo!")
7
```



## Criação de API

- Na pasta livros, criar um arquivo chamado “urls.py”;
- Deixar o arquivo assim:



## Criação de API

biblioteca > livros >  urls.py > ...

```
1  ∨ from django.urls import path
2    from . import views
3
4  ∨ urlpatterns = [
5      |     path('', views.index, name='index'),
6      | ]
7
8  |
```



## Criação de API

- Na pasta biblioteca, alterar o arquivo “urls.py”;
- Deixar o arquivo assim:





## Criação de API

```
biblioteca > biblioteca >  urls.py > ...
```

```
17  from django.urls import path, include
18
19  urlpatterns = [
20      path('admin/', admin.site.urls),
21      path('', include('livros.urls'))
22  ]
```



## Criação de API

- Rodar o comando:  
`py manage.py runserver`



## Criação de API

● Resultado?

← → ↻ ⓘ 127.0.0.1:9000

Olá, mundo!



## Criação de API

- Rodar o comando:  
`py -m pip install requests`



## Criação de API

```
Collecting charset-normalizer<4,>=2
  Downloading charset_normalizer-3.1.0-cp310-cp310-macosx_10_9_x86_64.whl (124 kB)
|████████████████████████████████████████| 124 kB 1.5 MB/s
Collecting certifi>=2017.4.17
  Downloading certifi-2023.5.7-py3-none-any.whl (156 kB)
|████████████████████████████████████████| 156 kB 3.1 MB/s
Collecting urllib3<3,>=1.21.1
  Downloading urllib3-2.0.2-py3-none-any.whl (123 kB)
|████████████████████████████████████████| 123 kB 3.7 MB/s
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
Successfully installed certifi-2023.5.7 charset-normalizer-3.1.0 idna-3.4 requests-2.30.0 urllib3-2.0.2
WARNING: You are using pip version 21.2.4; however, version 23.1.2 is available.
You should consider upgrading via the '/Library/Frameworks/Python.framework/Versions/3.10/bin/python3 -m pip inst
all --upgrade pip' command.
```



## Criação de API

- Abrir o arquivo "views.py", dentro da pasta "livros";
- Deixar o arquivo assim:



## Criação de API

biblioteca > livros >  views.py > ...

```
1  ✓ from django.shortcuts import render
2
3  from django.http import HttpResponse
4  import requests
5  # Create your views here.
6  ✓ def index(request):
7      response = requests.get('https://araujofa.pythonanywhere.com/livros/')
8      livros = response.json()
9      return HttpResponse(livros)
```



## Criação de API

- Rodar o comando:  
`py manage.py runserver`





## Criação de API

### ● Resultado?

```
{'id_livro': 'efb4bbb7-f859-4ec1-8747-e9891aaa4698', 'titulo': 'Companhia Academia', 'autor': 'Autor do livro', 'ano_lancamento': 12323, 'estado': 'Novo', 'paginas': 123, 'editora': 'Editora do livro', 'dt_criacao': '2023-05-11T00:04:58.967803Z', 'imagem': 'https://araujofa.pythonanywhere.com/uploads/uploads/apple_capa-4-1060x596.jpg'}
```



## Criação de API

- Abrir o arquivo "views.py", dentro da pasta "livros";
- Deixar o arquivo assim:



## Criação de API

biblioteca > livros >  views.py > ...

```
1  from django.shortcuts import render
2
3  from django.http import HttpResponseRedirect
4  import requests
5  # Create your views here.
6  def index(request):
7      response = requests.get('https://araujofa.pythonanywhere.com/livros/')
8      livros = response.json()
9      context = {
10         'livros': livros
11     }
12     return render(request, 'index.html', context)
```



## Criação de **API**

- Dentro da pasta “livros”, criar uma pasta chamada “templates”;
- Dentro dessa pasta, criar um arquivo chamado “index.html”;
- Acessar:  
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- Copiar o código no item “2”;



## Criação de API

PROJETO

biblioteca

biblioteca

livros

\_\_pycache\_\_

migrations

templates

index.html

\_\_init\_\_.py

admin.py

apps.py

models.py

tests.py

urls.py

views.py

db.sqlite3

manage.py

biblioteca > livros > templates > index.html

1

2

3

4

5

6

7

8

9

10

11

12

13

```

<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/
bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe
+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJ0Z"
crossorigin="anonymous">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/
bootstrap.bundle.min.js"
integrity="sha384-ENjd04Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfA
dVQSZe" crossorigin="anonymous"></script>
  </body>
</html>

```



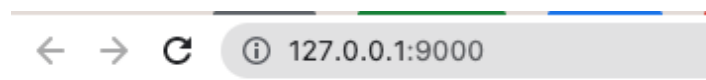
## Criação de API

- Rodar o comando:  
`py manage.py runserver`



## Criação de API

● Resultado?



# Hello, world!



## Criação de API

- Abrir o arquivo "index.html";
- Deixar o arquivo assim:





## Criação de API

```

9  <body>
10  <div class="container">
11    <div class="row">
12      <div class="col-12 py-3">
13        <a href="/adicionar" class="btn btn-primary">Adicionar</a>
14      </div>
15      {% for livro in livros %}
16      <div class="col-12 col-md-6 col-lg-4">
17        <div class="card" style="width: 18rem;">
18          
19          <div class="card-body">
20            <h5 class="card-title">{{ livro.titulo }}</h5>
21            <p class="card-text">
22              Autoria: {{ livro.autor }}
23            </p>
24            <a href="#" class="btn btn-primary">Editar</a>
25          </div>
26        </div>
27      </div>
28      {% endfor %}
29    </div>
30  </div>

```



## Criação de API

- Rodar o comando:  
`py manage.py runserver`



## Criação de API

● Resultado?





## Criação de API

- Abrir o arquivo "views.py" dentro da pasta "livros";
- Deixar o arquivo assim:



## Criação de API

```
14  ✓ def adicionar(request):  
15  ✓     context = {  
16  
17     }  
18     return render(request, 'adicionar.html', context)|
```



## Criação de API

- Criar um arquivo chamado “adicionar.html” na pasta “templates”;
- Copiar o conteúdo do arquivo “index.html” e colar no arquivo “adicionar.html”;
- O arquivo “adicionar.html” deve ficar assim:



## Criação de API

```
13  ✓ <form method="post" action="">
14    {% csrf_token %}
15  ✓ <div class="mb-3">
16    <label>Titulo</label>
17    <input type="text" class="form-control" name="titulo">
18  </div>
19  ✓ <div class="mb-3">
20    <label>Autor</label>
21    <input type="text" class="form-control" name="autor">
22  </div>
23  ✓ <div class="mb-3">
24    <label>Ano de lançamento</label>
25    <input type="text" class="form-control" name="ano_lancamento">
26  </div>
27  ✓ <div class="mb-3">
28    <label>Estado</label>
29    <input type="text" class="form-control" name="estado">
30  </div>
```



## Criação de API

```
31  ✓ <div class="mb-3">
32      <label>Número de páginas</label>
33      <input type="text" class="form-control" name="paginas">
34  </div>
35  ✓ <div class="mb-3">
36      <label>Editora</label>
37      <input type="text" class="form-control" name="editora">
38  </div>
39      <button type="submit" class="btn btn-primary">Enviar</button>
40  </form>
```





## Criação de API

- Abrir o arquivo "urls.py" dentro da pasta "livros";
- Deixar o arquivo assim:



## Criação de API

```
1  ✓ from django.urls import path
2    from . import views
3
4  ✓ urlpatterns = [
5      |     path('', views.index, name='index'),
6      |     path('adicionar/', views.adicionar, name='adicionar')
7      | ]
```



## Criação de API

- Rodar o comando:  
`py manage.py runserver`
- Acessar: `http://127.0.0.1:8000/adicionar/`



## Criação de API

### ○ Resultado?

Título

Autor

Ano de lançamento

Estado

Número de páginas

Editora

Enviar



## Criação de API

- Preencher o formulário e clicar e enviar;
- O que acontece?



## Criação de API

- Abrir o arquivo "views.py" dentro da pasta "livros";
- Deixar o arquivo assim:



## Criação de API

```
biblioteca > livros > views.py > ...
1  ∨ from django.shortcuts import render, redirect
2
3  from django.http import HttpResponseRedirect
4  import requests
5  # Create your views here.
6  > def index(request): ...
13
14 ∨ def adicionar(request):
15 ∨     if request.method == "POST":
16         dados = request.POST
17 ∨         novo_livro = {
18             "titulo": dados["titulo"],
19             "autor": dados["autor"],
20             "ano_lancamento": dados["ano_lancamento"],
21             "estado": dados["estado"],
22             "paginas": dados["paginas"],
23             "editora": dados["editora"]
24         }
25         post_response = requests.post('https://araujofa.pythonanywhere.com/
livros/', json=novo_livro)
26         return redirect("index")
27
28     context = {}
29     return render(request, 'adicionar.html', context)
```



## Criação de API

- Preencher o formulário e clicar e enviar;
- O que acontece?






## Criação de API

Adicionar




**Companhia Academia**  
Autoria: Autor do livro

Editar



**Companhia Academia**  
Autoria: Autor do livro

Editar



**Companhia Academia**  
Autoria: Autor do livro

Editar



# Trabalho avaliativo



## Trabalho **avaliativo**

- Fazer o que é descrito no arquivo trabalho2.pdf disponível no teams;



# Obrigado!

*Alguma pergunta?*

Contato:

☉ 040601692@prof.unama.br