

Django App e URLs



Prof. Bruno Gomes



@profbrunogomes



Aula de Hoje

2

- » **Django:**
 - » **Aplicação**
 - » **URLs**
 - » **Views**



1.

Aplicação

Aplicação

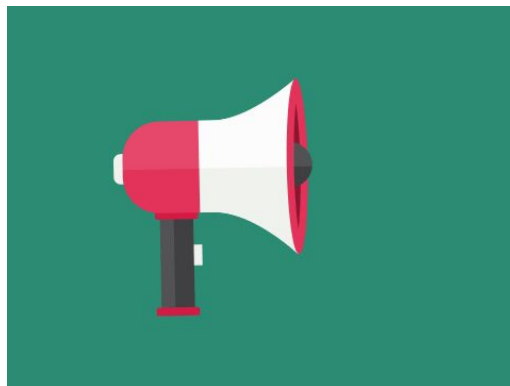
4

- » Todo Projeto Django é formado por 1 ou vários aplicativos.;
 - » Segundo a documentação: “Uma app é uma aplicação web que faz alguma coisa — i.e., um weblog ou uma simples enquete.”
 - » Um app deve ter uma responsabilidade única e bem definida.
 - » São os responsáveis por sua flexibilidade e alto reaproveitamento de componentes.
-
- » Documentação:
 - » <https://docs.djangoproject.com/en/4.1/intro/tutorial01/#creating-the-polls-app>

Observação

5

- » Por enquanto vamos trabalhar com apenas **1 aplicação**;
- » Posteriormente será ensinado como criar várias e integrá-las.



Criando a primeira aplicação

6

- » No diretório raiz do Projeto (o que contém **manage.py**), executar o comando (neste exemplo chamamos de core, mas poderia ser qualquer nome):

```
python manage.py startapp core
```

Terminal

Diretório do Projeto

7

» Resultado:

```
▼ PrimeiroSite
  ▼ core
    ► migrations
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    views.py
  ▼ PrimeiroSite
    ► __pycache__
    __init__.py
    settings.py
    urls.py
    wsgi.py
    manage.py
```


Informando ao Django que existe Core

8

- » Abrir o arquivo PrimeiroSite/settings.py, e acrescentar a app:

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

settings.py

Observação

9

```
▼ PrimeiroSite
  ▼ core
    > migrations
    • __init__.py
    • admin.py
    • apps.py
    • models.py
    • tests.py
    • views.py
  ▼ PrimeiroSite
    > __pycache__
    • __init__.py
    • asgi.py
    • settings.py
    • urls.py
    • wsgi.py
    ≡ db.sqlite3
    • manage.py
```

Diretório e arquivos da aplicação (core)

Diretório e arquivos do projeto (Primeiro Site)

2.

URLs

2.1

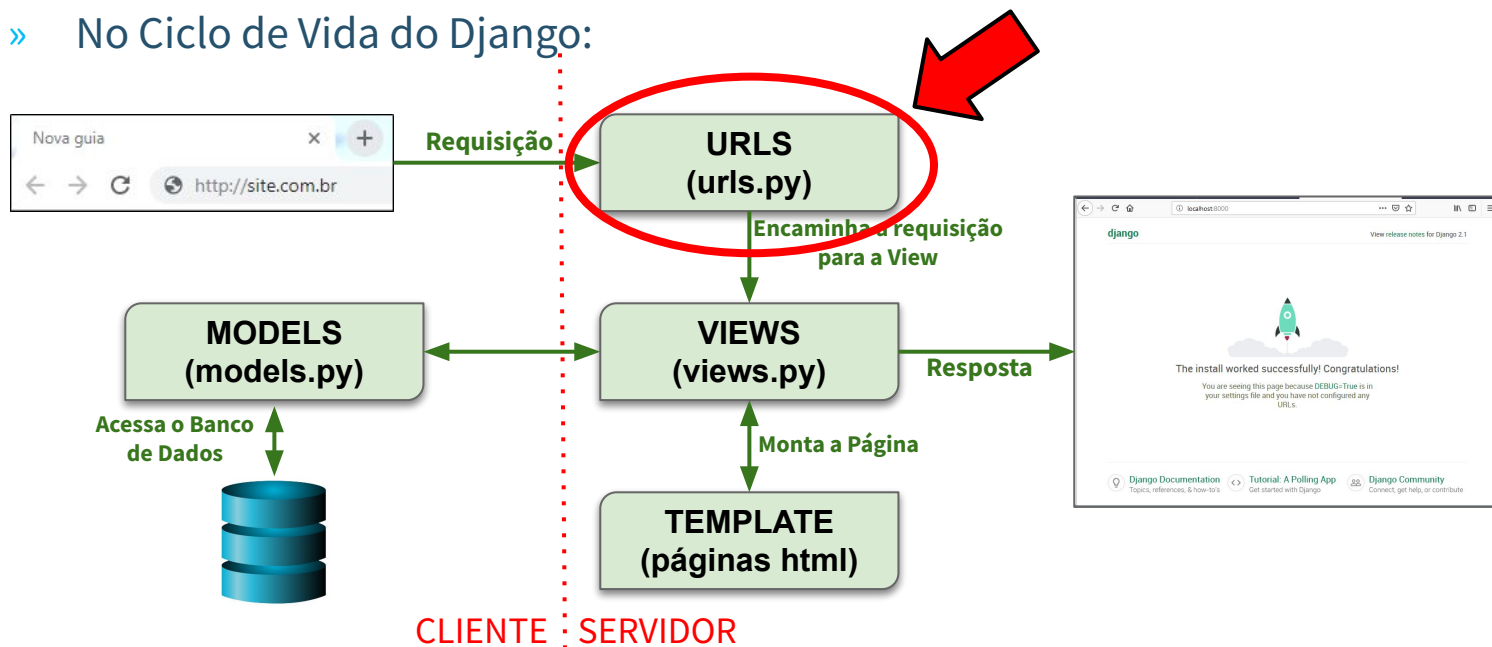
URLs

Conceitos Gerais

URLs

12

» No Ciclo de Vida do Django:



Documentação

13

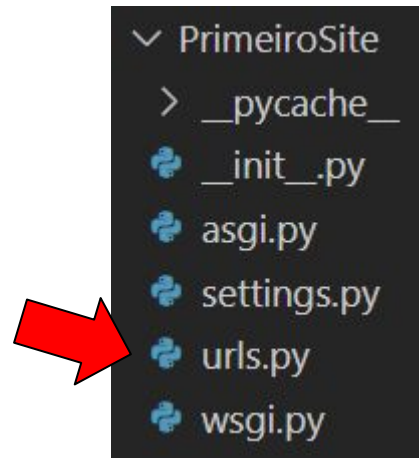
» <https://docs.djangoproject.com/en/4.1/topics/http/urls/>



URLs

14

- » As URLs recebem as requisições feitas pelos clientes, e direciona para uma determinada **função** localizada na **View**;
- » O arquivo **urls.py** serve como um mapeamento;



Arquivo urls.py

15

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

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

PrimeiroSite/urls.py

Testando a URL já criada

16

- » Iniciem o servidor, e acessem o endereço abaixo:
 - » Obs.: Necessita estar com o servidor ligado (**python manage.py runserver**)



```
http://localhost:8000/admin
```

Navegador

Vamos criar uma URL

17

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

urlpatterns = [
    path('ola/', olamundo),
    path('admin/', admin.site.urls),
]
```

PrimeiroSite/urls.py

```
path('ola/', olamundo),
```

- » **ola** é a URL que será acessada no navegador
 - » `http://localhost:8000/ola`
- » **olamundo** é o nome da função que deve ser criada na View (`views.py`)

Curiosidade

19

```
path('ola/', ola),
```

PrimeiroSite/urls.py

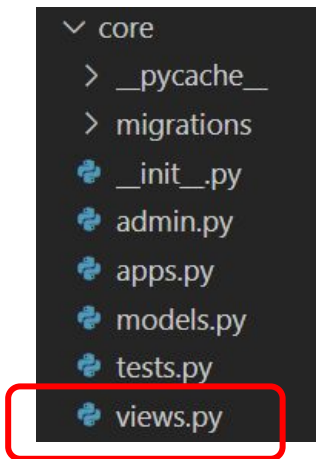
- » O nome da URL pode ser o mesmo nome da função da view;



Acessando a View de core

20

- » Por padrão, toda aplicação tem uma view;
- » Acessar o arquivo views.py da app core:



Criar a Função olamundo na View

21

```
from django.shortcuts import render
from django.http import HttpResponse

def olamundo(request):
    return HttpResponse('Olá Mundo')
```

core/views.py

Importar a função da View de core em url.py 22

```
from django.contrib import admin
from django.urls import path
from core.views import olamundo

urlpatterns = [
    path('ola/', olamundo),
    path('admin/', admin.site.urls),
]
```

PrimeiroSite/urls.py

Testando a URL já criada

23

- » Iniciem o servidor, e acessem o endereço abaixo:

`http://localhost:8000/ola`

Navegador



Olá Mundo

Explicando a View

24

```
def olamundo(request):  
    return HttpResponse('Olá Mundo')
```

core/views.py

- » Toda função acessada por uma URL deve receber um objeto **request** como parâmetro (pode ter qualquer outro nome);
- » A função deve retornar também um objeto **HttpResponse**;
 - » Neste exemplo, o objeto recebe no construtor um texto e retorna como página.

- » O retorno imprime diretamente no HTML, então é possível colocar tags:

```
from django.http import HttpResponse

def olamundo(request):
    return HttpResponse('<h1>Olá Mundo</h1>')
```

core/views.py

Página Inicial do Projeto

26

```
from django.contrib import admin
from django.urls import path
from core.views import olamundo
from core.views import home

urlpatterns = [
    path('', home),
    path('ola/', olamundo),
    path('admin/', admin.site.urls),
]
```

PrimeiroSite/urls.py

Função home em Views de core

27

» Adicionar a Função:

```
def home(request):  
    return HttpResponse('Página Inicial')
```

core/views.py

Testando a URL

28

- » Iniciem o servidor, e acessem o endereço abaixo:



`http://localhost:8000`

Navegador



Página Inicial

Dúvidas?



Vamos Praticar

30

- » Criar uma nova URL, que seja acessada através do endereço:
 - » <http://localhost:8000/pagina1>
 - » Nesta página deve inicialmente aparecer apenas o texto em parágrafo: “Página 1”;
- » Criar uma nova URL, que seja acessada através do endereço:
 - » <http://localhost:8000/pagina2>
 - » Nesta página deve inicialmente aparecer apenas o texto em parágrafo: “Página 2”;

Vamos Praticar

31

- » Alterando a atividade anterior, a página 1 deverá agora exibir apenas um link para a página 2;
- » Depois acrescentar um link na página 2 para a página 1 também.

2.2

URLs

Dicas

Outra forma de importar

33

```
from core import views

path('', views.home),
path('ola/', views.olamundo),
```

PrimeiroSite/urls.py

Simplificando a Importação

34

```
from core.views import olamundo  
from core.views import home
```

[PrimeiroSite/urls.py](#)



```
from core.views import olamundo, home
```

[PrimeiroSite/urls.py](#)

Simplificando a Importação

35

- » Se desejar importar todas as funções da view, basta acrescentar:

```
from core.views import *
```

[PrimeiroSite/urls.py](#)

- » Não é recomendado para views com muitas funções, o projeto poderá ficar lento.

2.3

URLs

Passagem de Parâmetros

Documentação

37

- » <https://docs.djangoproject.com/en/4.1/topics/http/urls/#path-converters>



Recebendo Parâmetros

38

- » Sintaxe:
 - » `/<tipo_dado:nome_var>`

- » Tipos de dados:
 - » **str** - String não vazia sem o caractere '/'
 - » **path** - String não vazia incluindo o caractere '/'. Exemplo: '/path/to/file'
 - » **int** - números inteiros
 - » **slug** - sequência de slug que consiste em letras ou números ASCII, além dos caracteres hífen e sublinhado (ex.: curso-django-iniciantes);
 - » **uuid** - Formato UUID que evita várias URLs mapeando a mesma página;
 - » Sobre UUID:
 - <https://medium.com/trainingcenter/o-que-%C3%A9-uuid-porque-us%C3%A1-lo-ad7a66644a2b>

Recebendo Parâmetros - URL

39

```
from core.views import olamundo, home, cadastro

urlpatterns = [

    path('dados/<str:nome>/', cadastro),

]
```

PrimeiroSite/urls.py

Recebendo Parâmetros - VIEW

40

```
def cadastro(request, nome):  
    return HttpResponse('Nome: '+nome)
```

[core/views.py](#)

- » Obs.: Sempre request deve ser o primeiro parâmetro;

Testando

41

- » Iniciem o servidor, e acessem o endereço abaixo:

```
http://localhost:8000/dados/Bruno/
```

Navegador



Nome: Bruno

Vários Parâmetros

42

```
from core.views import olamundo, home, cadastro

urlpatterns = [

    path('dados/<str:nome>/<int:idade>/', cadastro),

]
```

PrimeiroSite/urls.py

Vários Parâmetros

43

```
def cadastro(request, nome, idade):  
    return HttpResponse('Nome: '+nome+'; Idade: '+str(idade))
```

core/views.py

Testando

44

- » Iniciem o servidor, e acessem o endereço abaixo:

`http://localhost:8000/dados/Bruno/20`

Navegador



Nome: Bruno; Idade: 20

2.3

URLs

Nomeando URLs

Documentação

46

- » <https://docs.djangoproject.com/en/4.1/intro/tutorial01/#write-your-first-view>



Facilitando a utilização de URLs

47

- » É possível nomear as URLs, para que possam ser chamadas pelo nome e não pelo endereço da URL (ex: evita um link do tipo /nome/nome2/nome3).

```
path(' ', home, name="home"),  
path('ola/', olamundo, name="ola"),
```

- » Veremos na aula de Templates como utilizar o nome da URL.

Dúvidas?

