

Django

Framework web para perfeccionistas com prazos

Igor Sobreira
www.igorsobreira.com

o que é um framework?

“é uma estrutura de suporte definida em que um outro projeto de software pode ser organizado e desenvolvido”

“Frameworks são projetados com a intenção de facilitar o desenvolvimento de software, habilitando designers e programadores a gastarem mais tempo determinando as exigências do software do que com detalhes de baixo nível do sistema.”

o que é django?

- framework web escrito em python
- desenvolvido no Lawrence Journal-World
- publicado em julho de 2005
- open-source (licença BSD)

“Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design”



por que python?

- *altíssimo* nível
- multiparadigma (OO, funcional e procedural)
- tipagem dinâmica e forte
- coletor de lixo
- multiplataforma
- código limpo
- fácil de aprender

“Produtividade e legibilidade”

onde usar python?

- web
 - Django, TurboGears, Pylons. Zope, Plone, Grok...
- desktop
 - PyGTK, PyQt, WxPython, TkInter...
- mobile
 - PyS60 (Symbian), PyMaemo...

por que usar django?

- feito em Python :-)
- MTV == MVC
- ORM (Mapeamento Objeto Relacional)
- Interface de Administração automática
- URLs limpas
- Boa documentação (+inglês)

separando as tarefas

MVC (Model – View – Controller)

MTV (Model – Template – View)

organizando o código

um projeto possui uma ou mais aplicações

DRY – Don't Repeat Yourself

caminho de uma requisição

Request



Cliente

urls.py

```
(  
    (r'^contato/$',  
     'blog.contato'),  
    (r'^posts/$',  
     'blog.posts'),  
)
```



views.py

```
def posts(request):  
    p=Post.objects.all()  
    return render_to_response (  
        'posts.html',  
        {'posts':p}  
    )
```



models.py

```
class Post(Model):  
    title=CharField(max_length=50)  
    content=TextField()  
    pub_date=DateField()  
    cat=ForeignKey(Category)  
    autor=CharField(max_length=50)  
  
class Category(Model):  
    name=CharField()
```



posts.html

```
{% extends "base.html" %}  
  
{% block content %}  
  
    {% for p in posts %}  
        <h1> {{ p.title }} </h1>  
        <p> {{ p.content }} </p>  
    {% endfor %}  
  
{% endblock %}
```



Response

models.py

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

```
class Image(models.Model):  
    image = models.ImageField(upload_to='fotos')  
    description = models.CharField(max_length=50)  
    pub_date = models.DateTimeField(  
        default=datetime.now)
```

models.py (2)

```
BEGIN;
```

```
CREATE TABLE "teste_image" (
```

```
    "id" integer NOT NULL PRIMARY KEY,
```

```
    "image" varchar(100) NOT NULL,
```

```
    "description" varchar(50) NOT NULL,
```

```
    "pub_date" datetime NOT NULL
```

```
);
```

```
COMMIT;
```

models.py (3)

```
class Category(models.Model):  
    name = models.CharField(max_length=50)  
  
class Post(models.Model):  
    title = models.CharField(max_length=50)  
    content = models.TextField()  
    pub_date = models.DateTimeField(default=datetime.now)  
    author = models.CharField(max_length=50)  
    category = models.ForeignKey(Category)
```


models.py (4)

```
BEGIN;  
CREATE TABLE "teste_category" (  
    "id" integer NOT NULL PRIMARY KEY,  
    "name" varchar(50) NOT NULL  
);  
CREATE TABLE "teste_post" (  
    "id" integer NOT NULL PRIMARY KEY,  
    "title" varchar(50) NOT NULL,  
    "content" text NOT NULL,  
    "pub_date" datetime NOT NULL,  
    "author" varchar(50) NOT NULL,  
    "category_id" integer NOT NULL REFERENCES  
    "teste_category" ("id")  
);  
COMMIT;
```

persistindo os dados

```
cat = Category(name = 'Django')  
cat.save()
```

persistindo os dados (2)

```
post = Post( title = 'Primeira app em django',  
            content = 'vamos comecar...',  
            author = 'Igor',  
            category = cat )
```

```
post.save()
```

acessando os dados

```
p = Post.objects.all()
```

```
[<Post: Post object>, <Post: Post object>]
```

acessando os dados (2)

```
p = Post.objects.filter(author = 'Igor')
```

acessando os dados (3)

```
p = Post.objects.get(id=2)
```

urls.py

```
from django.conf.urls.defaults import *

urlpatterns = patterns('myproject.myapp.views',
    (r'^lastest/$', 'last_posts'),
    (r'^post/(\d+)/$', 'post_details'),
    (r'^contact/$', 'contact'),
)
```

views.py

```
from django.shortcuts import render_to_response
from myapp.models import Post
```

```
def last_posts(request):
    posts = Post.objects.order_by('-date')[:5]
    return render_to_response('latest.html',
                              {'posts': posts} )
```


templates

- hora dos web designers :-)
- simplesmente arquivos .html com “buracos” que serão preenchidos com o conteúdo passado pela view

```
<html>
<head>
  <title> Blog - {{ post.title }} </title>
</head>
<body>
  <h1> {{ post.title }} </h1>
  <p> {{ post.content }} </p>
  <small> Publicado por {{ post.author }} em
    {{ post.pub_date }} </small>
</body>
</html>
```

templates (2)

- variáveis
 - `{{ nome }}`, `{{ lista }}`
- filters
 - `{{ nome|safe }}`
 - `{{ lista|unordered_list }}`
- tags
 - `{% for nome in lista %} {{ nome }} {% endfor %}`
 - `{% if condicao %} OK {% endif %}`

templates (3)

```
<ul>
```

```
  {% for post in last_posts %}
```

```
    <li> {{ post|date:"d/m/Y h:i" }} -
```

```
      {{ post.title }} </li>
```

```
  {% endfor %}
```

```
</ul>
```

mais facilidades...

- formulários
- testes
- controle usuários (auth/auth)
- internacionalização (i18n)
- AJAX ?!?!
- feeds RSS / Atom
- cache

dando uma olhada no admin...



V2 Windsurf Center

Bem vindo, igor. Documentação / Alterar senha / Encerrar sessão

Administração do Site

Auth	
Grupos	+ Adicionar Modificar
Utilizadores	+ Adicionar Modificar
Sites	
Sites	+ Adicionar Modificar
Comments	
Comentários	+ Adicionar Modificar
Comentários livres	+ Adicionar Modificar
Flatpages	
Páginas planas	+ Adicionar Modificar
V2Photogallery	
Eventos	+ Adicionar Modificar
Imagens	+ Adicionar Modificar
V2News	
Imagem de Notícias	+ Adicionar Modificar
Notícias ou Previsões	+ Adicionar Modificar
V2Equipments	
Categorias dos Equipamentos	+ Adicionar Modificar
Equipamentos	+ Adicionar Modificar
Fabricantes	+ Adicionar Modificar

Ações Recentes

Minhas Ações

- [pedro](#)
Utilizador
- [pedro](#)
Utilizador
- [pedro](#)
Utilizador
- [+ Pedro Buscape \[pedro\]](#)
Perfil do Usuário
- [pedro](#)
Utilizador
- [+ Velejo irado](#)
Video
- [Velejo irado](#)
Video
- [+ Filme de teste](#)
Video
- [+ Igor Sobreira \[igor\]](#)
Perfil do Usuário
- [+ Velejo irado](#)
Video



Selecione Notícia ou Previsão para modificar

[Adicionar Notícia ou Previsão](#) +

2008				
Título	Categoria	Data	Autor	Quant. Imagens
Teste só	Notícia	Março 5, 2008, 1:38 a.m.	pedro	1
Mensagens para os usuários no Django	Notícia	Março 5, 2008, 1:11 a.m.	pedro	0
Vento forte	Previsão	Fev. 25, 2008, 5:29 p.m.	igor	0
Testando as notícias	Notícia	Fev. 23, 2008, 5:06 p.m.	igor	2

5 Notícias ou Previsões

Filtro

Por Data

- Qualquer data
- Hoje
- Últimos 7 dias
- Este mês
- Este ano

Por Categoria

- Todos
- Notícia
- Previsão

Por autor

- Todos
- igor
- pedro



Adicionar Notícia ou Previsão

Detalhes da Notícia

 Título:

 Categoria:

 Descrição Inicial:

Caso não seja preenchida será os primeiros caracteres da descrição. Máximo de 220 caracteres.

 Descrição:

 Data: Hoje

 Hora: Agora

Março 2008						
D	S	T	Q	Q	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Imagens

 Miniatura:

Miniatura para aparecer na página principal (diminuída automaticamente)



Adicionar Notícia ou Previsão

⊖ Por favor corrija os erros abaixo.

Detalhes da Notícia

⚠ Este campo é requerido.

Título:

⚠ Este campo é requerido.

Categoria:

Descrição Inicial:

Caso não seja preenchida será os primeiros caracteres da descrição. Máximo de 220 caracteres.

⚠ Este campo é requerido.

Descrição:

Data: Hoje

dúvidas

?

- <http://www.djangoproject.com/>
- <http://www.djangobrasil.org/>
- <http://groups.google.com/group/django-brasil/>
- <http://djangobook.com/>
- <http://www.python.org>
- <http://www.pythonbrasil.com.br/>
- <irc.freenode.net> (#django-br, #django)

Igor Sobreira

- www.igorsobreira.com
- <http://djangopeople.net/igorsobreira>
- G-Talk: v2.igor@gmail.com
- MSN: igor@v2windcenter.com
- IRC: igors

se der tempo...

formulários

```
from django import newforms as forms
```

```
class ContatoForm(forms.Form):  
    nome = forms.CharField(label=u'Nome')  
    email = forms.EmailField(label=u'Email',  
                             required=False)  
    msg = forms.CharField(label=u'Mensagem',  
                          widget=forms.Textarea)
```

Formulários (contato.html)

```
<form action="." method="post">
```

```
{{ form.as_p }}
```

```
<input type="submit" value="Enviar" />
```

```
</form>
```


Formulários (views.py)

```
def contato(request):  
    if request.method == "POST":  
        form = ContatoForm(request.POST)  
        if form.is_valid():  
            # faz algo aqui...  
            return HttpResponseRedirect("/sucesso/")  
    else:  
        form = ContatoForm()  
    return render_to_response("contato.html",  
                              {"form":form})
```