

Welcome to the Django

O curso onde *Prática* guia *Teoria*.

<http://welcometothedjango.com.br>

Introdução ao Python

“ “

Python fits your mind.

, , ,

Bruce Eckel

História



@gvanrossum

Características

- ➔ Alto nível
- ➔ Interpretada
- ➔ Dinâmica
- ➔ Dinamicamente Tipada
- ➔ Fortemente Tipada
- ➔ Multi-paradigma

“ “

Zen of Python

, , ,

Tim Peters

```
>>> import this
```

The Zen of Python, by Tim Peters

Bonito é melhor que feio.

Explícito é melhor que implícito.

Simples é melhor que complexo.

Complexo é melhor que complicado.

Plano é melhor que aninhado.

Esparsos é melhor que denso.

Legibilidade conta.

Casos especiais não são especiais o bastante para se quebrar as regras.

Embora a simplicidade supere o purismo.

Erros nunca deveriam passar silenciosamente.

A menos que explicitamente silenciados.

Ao encarar a ambiguidade, recuse a tentação de adivinhar.

Deveria haver uma - e preferencialmente apenas uma - maneira óbvia de se fazer isto.

Embora aquela maneira possa não ser óbvia à primeira vista se você não for holandês.

Agora é melhor que nunca.

Embora nunca, seja muitas vezes melhor que pra já.

Se a implementação é difícil de explicar, é uma má idéia.

Se a implementação é fácil de explicar, pode ser uma boa idéia.

Namespaces são uma grande ideia - vamos fazer mais deles!

Já vem com **baterias**

Python Standard Library

<http://docs.python.org/library/>

Python em uma página

http://coffeeghost.net/pybat/python_cheatsheet.png

```
import os

def main():
    print 'Hello world!'
    print "This is Alice's greeting."
    print 'This is Bob\'s greeting.'

    foo(5, 10)

    print '=' * 10
    text = 'Current working directory is '
    print text + os.getcwd()

    counter = 0
    counter += 1
    food = ['apples', 'oranges', 'cats']

    for i in food:
        print 'I like to eat ' + i

    print 'Count to ten:'
    for i in range(1, 11):
        print i
```

```
def foo(param1, secondParam):
    res = param1 + secondParam
    print '%s plus %s is equal to %s' % (
        param1, secondParam, res)

    if res < 50:
        print 'foo'
    elif (res >= 50) and ((param1 == 42) or
        (secondParam == 24)):
        print 'bar'
    else:
        print 'moo'

    return res # This is a one-line comment.
    '''A multi-
    line string, but can also be a
    multi-line comment.'''
if __name__ == '__main__':
    main()
```

Quem usa Python?



<https://us.pycon.org/2013/sponsors/>

Comunidade Python

- Python - <http://python.org>
- Listas - <http://www.python.org/community/lists/>

- PythonBrasil - <http://pythonbrasil.org.br>
- Lista - <https://groups.google.com/group/python-brasil>

- PythonRio - <http://pythonrio.org>
- Lista - <http://br.groups.yahoo.com/group/pythonrio/>

Introdução ao Django

“ “ Django é um framework web de alto nível escrito em Python que possibilita rápido desenvolvimento e design pragmático.

, ,

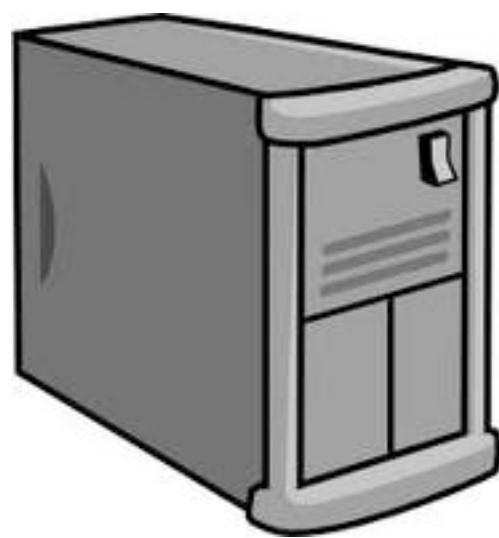
“Django é um framework
web de alto nível escrito
em Python...”

“...que possibilita rápido
desenvolvimento...”

“...e design pragmático.”

World Wide What?

Stateful vs Stateless



HTTP Request

```
$ curl -iv http://www.google.com/
* About to connect() to www.google.com port 80 (#0)
*   Trying 64.233.163.104... connected
* Connected to www.google.com (64.233.163.104) port 80 (#0)

GET / HTTP/1.1
User-Agent: curl/7.21.1 (i386-apple-darwin9.8.0)
Host: www.google.com
Accept: */*
```

HTTP Response

```
$ curl -i http://www.google.com/
```

HTTP/1.1 302 Found

Location: http://www.google.com.br/

Cache-Control: private

Content-Type: text/html; charset=UTF-8

Set-Cookie: PREF=ID=61dd6; expires=-1; path=/; domain=.google.com

Date: Thu, 14 Oct 2010 03:44:29 GMT

Server: gws

Content-Length: 222

<HTML><HEAD>

...

</BODY></HTML>

História



@adrianholovaty

@simonw

@jacobian



Arquitetura

Model

Template

View

Filosofias

- ➔ Desenvolvimento rápido
- ➔ Don't repeat yourself
- ➔ Menos código
- ➔ Acoplamento fraco
- ➔ Consistência

http://www.djangoproject.com/documentation/design_phилosophies/

Timeline

2003 - Início

2005 ~ 2007 - Pré 1.0

2008 - Release 1.0

2009 - Release 1.1

2010 - Release 1.2

2011 - Release 1.3

2012 - Release 1.4

2013 - Release 1.5

Quem usa Django?

Pinterest

globo.com



Instagram

canonical



DISQUS

<http://www.djangosites.org>

Comunidade Django

- ➔ Django Project - <http://djangoproject.com>
- ➔ Developers List - <http://groups.google.com.br/group/django-developers>
- ➔ Users List - <http://groups.google.com.br/group/django-users>

- ➔ Django Brasil - <http://djangobrasil.org>
- ➔ Lista de Usuários - <http://groups.google.com/group/django-brasil/>

Instalação

Ferramentas de Trabalho

- ➔ Python 2.7.3
 - ➔ Django 1.5.1
 - ➔ Git
 - ➔ Pip & Virtualenv
-
- ➔ GitHub.com
 - ➔ Heroku.com

Windows

Python 2.7.3

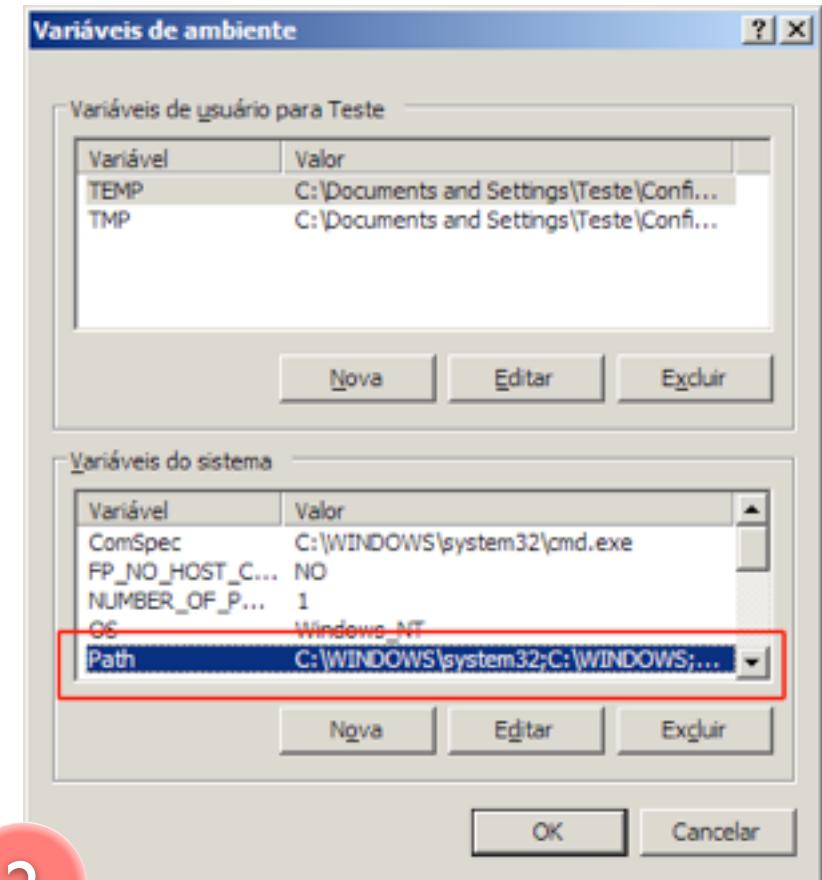
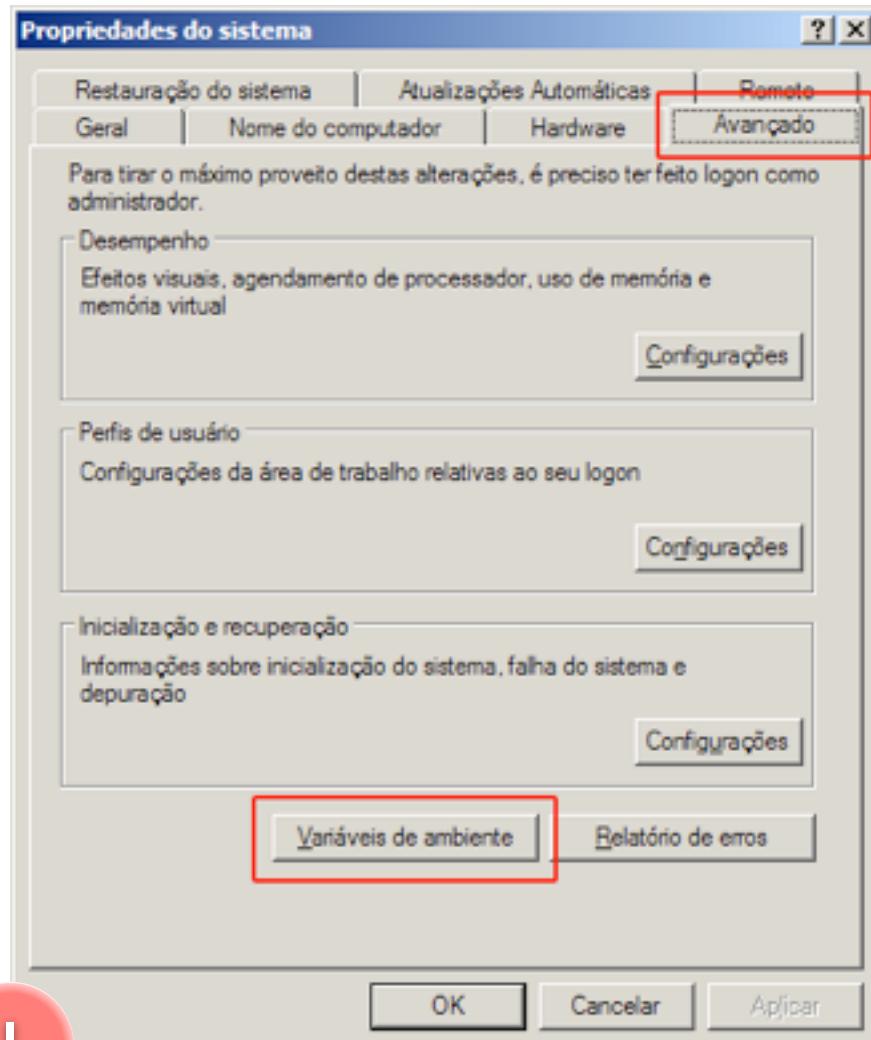
<http://python.org/ftp/python/2.7.3/python-2.7.3.msi>

Distribute

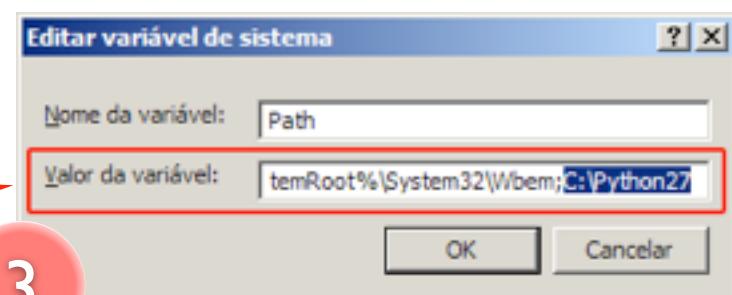
http://python-distribute.org/distribute_setup.py

Heroku Toolbelt

<https://toolbelt.heroku.com/download/windows>



2



3

C:\Python27;C:\Python27\Scripts

Pip & Virtualenv no Windows

No Prompt de Comando:

```
$ easy_install pip  
$ pip install virtualenv
```

Checkup no Windows

No Prompt de Comando:

```
$ python -V
```

Python 2.7.3

Linux (Debian/Ubuntu)

Já vem com Python instalado! \m/

Pip & Virtualenv no Linux

No Terminal:

```
$ curl -O http://python-distribute.org/distribute_setup.py  
$ sudo python distribute_setup.py  
  
$ sudo easy_install pip  
$ sudo pip install virtualenv
```

Heroku no Linux

Debian/Ubuntu:

```
$ wget -qO- https://toolbelt.heroku.com/install.sh | sh
```

Outras Distros:

<https://toolbelt.heroku.com/linux>

Mac

Lion

Já vem com Python 2.7 instalado! \m/

Versões Anteriores

Python 2.7

<http://www.python.org/ftp/python/2.7.3/python-2.7.3-macosx10.6.dmg>

Pip & Virtualenv no Mac

No Terminal:

```
$ curl -O http://python-distribute.org/distribute_setup.py  
$ sudo python distribute_setup.py
```

```
$ sudo easy_install pip  
$ sudo pip install virtualenv
```

Heroku no Mac

Heroku Toolbelt

<https://toolbelt.heroku.com/download/osx>

Conheçam nossa Cliente

- ➔ Sistema de Evento
- ➔ Divulgação
- ➔ Inscrição
- ➔ Instruções via email
- ➔ Gestão dos inscritos
- ➔ ...



Landingpage?



EventeX Google

EventeX

Bem-vindo ao EventeX!

O evento mais sensacional do ano depois do Dev in Rio!

Você não pode perder esta oportunidade!

Abertura das inscrições em 21/04/2013.

Para dúvidas e mais informações entre em contato com eventex@eventex.com.br

© 2013. EventeX, some rights reserved.

django SITE

```
# Cria um virtualenv  
$ virtualenv --distribute --unzip-setuptools wttd  
  
$ cd wttd  
  
# Ativa o virtualenv no Windows  
$ Scripts\activate  
  
# Ativa o virtualenv no Linux ou Mac  
$ source bin/activate
```

```
# Instala o Django no Virtualenv  
$ pip install django==1.5.1
```

```
# Instala o Django no Virtualenv  
$ pip install django==1.5.1
```

Dica

```
# Windows Hack para facilitar o uso do django-admin  
# Crie o django-admin.bat para chamar o django-admin.py  
$ echo @python "%VIRTUAL_ENV%\Scripts\django-admin.py" %* >  
"%VIRTUAL_ENV%\Scripts\django-admin.bat"
```

```
# Instala o Django no Virtualenv  
$ pip install django==1.5.1
```

```
# Windows Hack para facilitar o uso do django-admin  
# Crie o django-admin.bat para chamar o django-admin.py  
$ echo @python "%VIRTUAL_ENV%\Scripts\django-admin.py" %* >  
"%VIRTUAL_ENV%\Scripts\django-admin.bat"
```

```
# Cria o projeto Django no diretório atual (wttd).  
# Windows:  
$ django-admin startproject eventex .
```

```
# Linux ou Mac:  
$ django-admin.py startproject eventex .
```

Estrutura de Arquivos

wttd/

 manage.py

 eventex/

 __init__.py

 settings.py

 urls.py

 wsgi.py

```
$ python manage.py
```

Usage: manage.py subcommand [options] [args]

Options:

-v VERBOSITY, --verbosity=VERBOSITY

Verbosity level; 0=minimal output, 1=normal output,
2=verbose output, 3=very verbose output

--settings=SETTINGS The Python path to a settings module, e.g.

"myproject.settings.main". If this isn't provided, the
DJANGO_SETTINGS_MODULE environment variable will be
used.

--pythonpath=PYTHONPATH

A directory to add to the Python path, e.g.
"/home/djangoprojects/myproject".

--traceback

Print traceback on exception

--version

Show program's version number and exit

-h, --help

Show this help message and exit

Type 'manage.py help <subcommand>' for help on a specific subcommand.

Available subcommands:

...



Dica

```
# Se você estiver em wttd/  
python manage.py help
```

```
# Se você estiver em wttd/eventex  
python ../manage.py help
```

```
# Se você estiver em wttd/eventex/core/templates  
python ../../..../manage.py help
```

```
# Para generalizar, só com caminho absoluto  
/wttd/bin/python /wttd/manage.py help
```



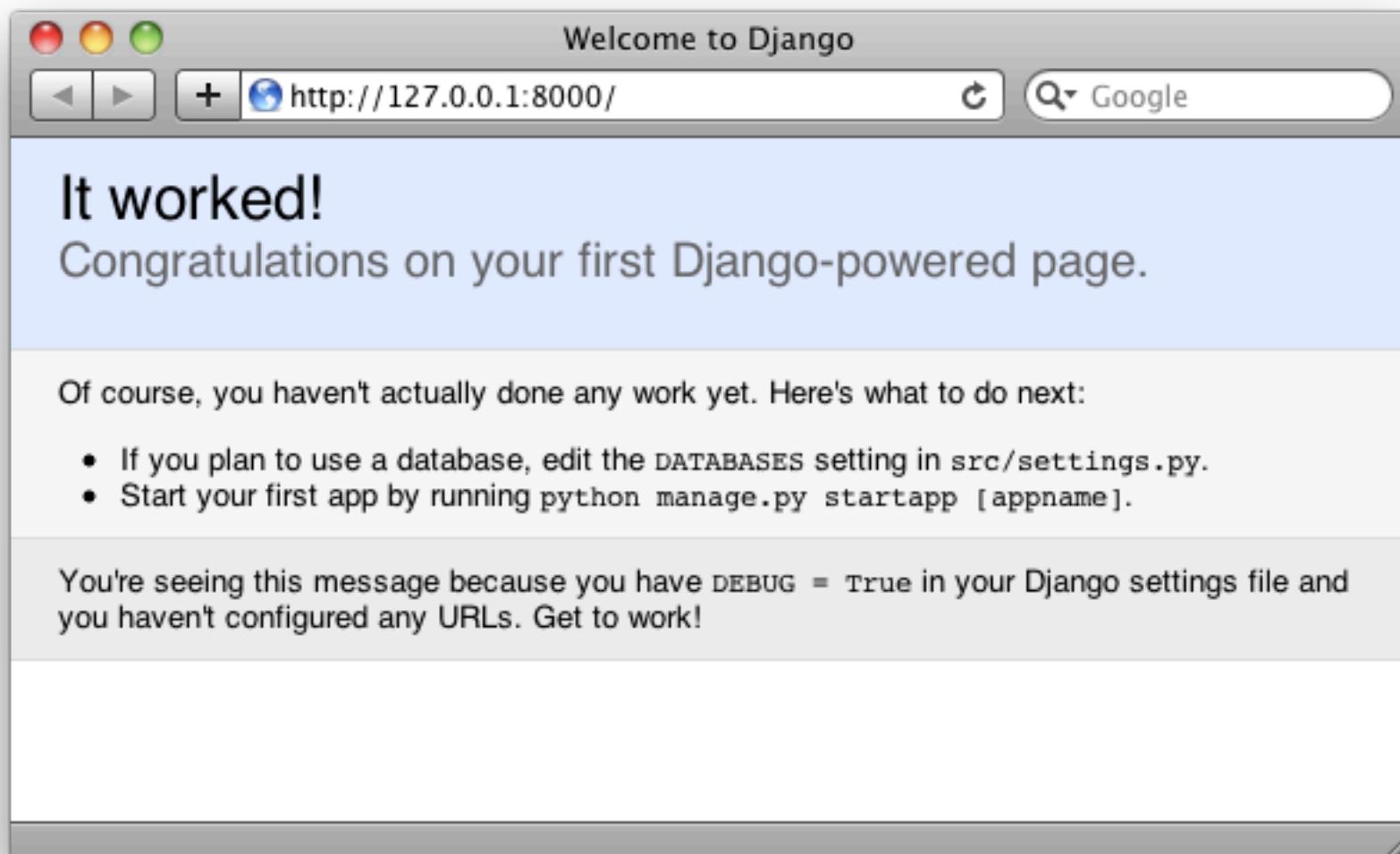
Dica

```
# De *qualquer lugar*, quero apenas:  
manage help
```

```
# no Windows  
# Crie o arquivo wttd\Scripts\manage.bat:  
@python "%VIRTUAL_ENV%\manage.py" %*  
  
# no Mac ou Linux  
# Crie um alias no seu ~/.bashrc ou ~/.profile  
alias manage='python $VIRTUAL_ENV/manage.py'
```

```
$ manage runserver  
Validating models...  
0 errors found
```

```
Django version 1.5.1, using settings 'eventex.settings'  
Development server is running at http://127.0.0.1:8000/  
Quit the server with CONTROL-C.
```



Unipath

Dica

```
# Instala o unipath no virtualenv do projeto  
$ pip install unipath
```

<http://sluggo.scraping.cc/python/unipath/Unipath-current/README.html>

eventex/settings.py

```
from unipath import Path
PROJECT_DIR = Path(__file__).parent

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': PROJECT_DIR.child('database.db'),
        'USER': '',
        'PASSWORD': '',
        'HOST': '',
        'PORT': ''
    }
}
```

```
$ manage syncdb
```

```
Creating tables ...
```

```
Creating table auth_permission
```

```
Creating table auth_group_permissions
```

```
Creating table auth_group
```

```
...
```

```
You just installed Django's auth system,  
which means you don't have any superusers defined.
```

```
Would you like to create one now? (yes/no): yes
```

```
Username (Leave blank to use 'henrique'): admin
```

```
E-mail address: admin@email.com
```

```
Password: admin
```

```
Password (again): admin
```

```
Superuser created successfully.
```

```
...
```

```
Installing custom SQL ...
```

```
Installing indexes ...
```

```
...
```

Mais...

<https://docs.djangoproject.com/en/1.5/topics/settings/>

“App Core”



```
# Criando uma app core em wttd/eventex  
$ cd eventex  
$ manage startapp core
```

wtdt/

manage.py

eventex/

__init__.py

settings.py

urls.py

wsgi.py

core/

__init__.py

models.py

tests.py

views.py

eventex/settings.py

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    # Uncomment the next line to enable the admin:
    # 'django.contrib.admin',
    # Uncomment the next line to enable admin documentation:
    # 'django.contrib.admindocs',
    'eventex.core',
)
```

eventex/urls.py

```
from django.conf.urls import patterns, include, url

# Uncomment the next two lines to enable the admin:
# from django.contrib import admin
# admin.autodiscover()

urlpatterns = patterns('eventex.core.views',
    url(r'^$', 'homepage', name='homepage'),

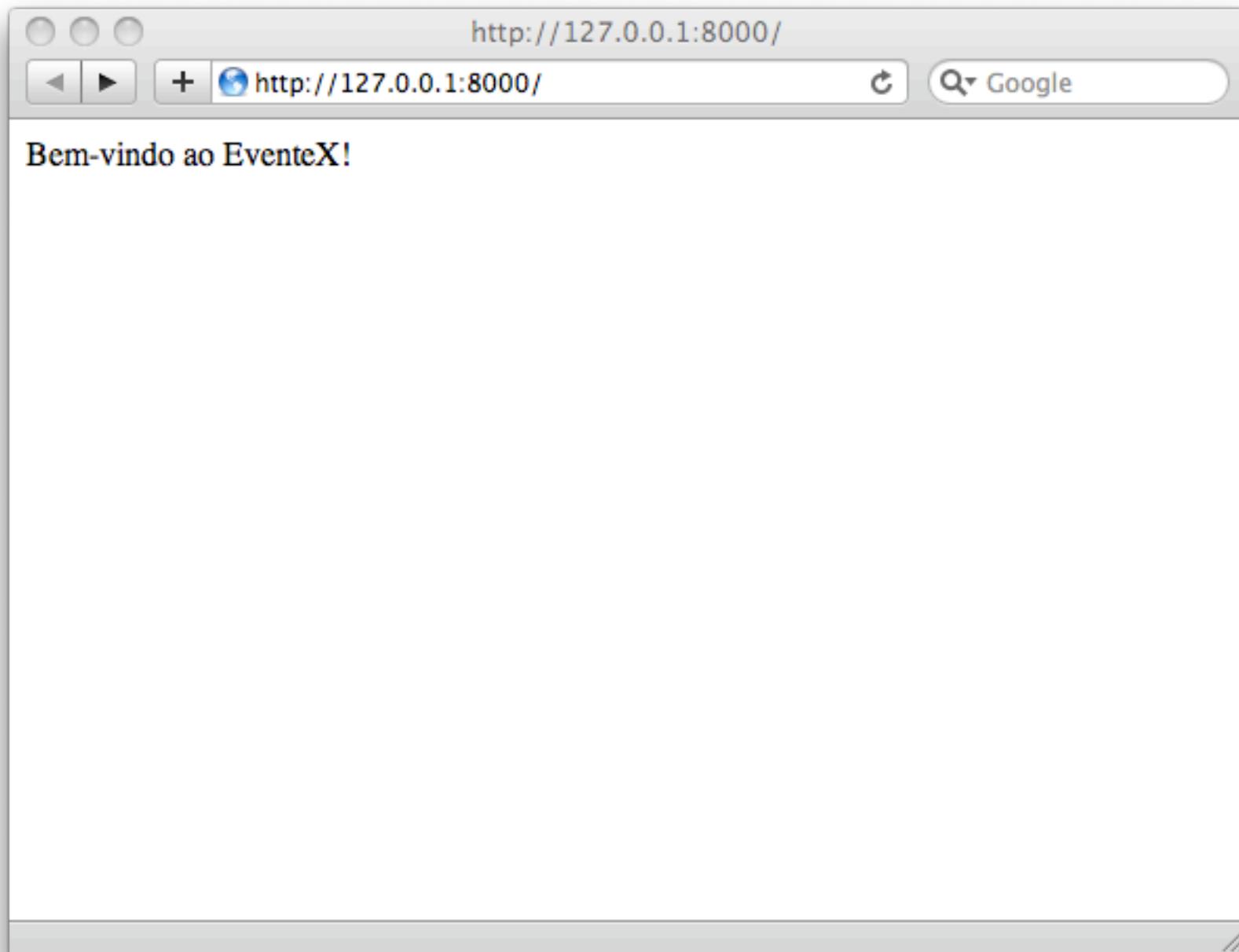
    # Uncomment the next line to enable the admin:
    # url(r'^admin/', include(admin.site.urls)),
)


```

eventex/core/views.py

```
# coding: utf-8
from django.http import HttpResponse

def homepage(request):
    return HttpResponse('Bem-vindo ao EventeX!')
```



HTML?



Onde colocar os
templates?

Onde colocar os
templates?

Dentro da App Core

Crie o diretório
wttd/eventex/core/templates

eventex/core/templates/index.html

`<h1>Bem-vindo ao EventeX!</h1>`

O que temos?

eventex/core/views.py

```
# coding: utf-8
from django.http import HttpResponse

def homepage(request):
    return HttpResponse('Bem-vindo ao EventeX!')
```

eventex/core/views.py

```
# coding: utf-8
from django.http import HttpResponse
from django.template import loader, Context

def homepage(request):
    t = loader.get_template('index.html')
    c = Context()

    content = t.render(c)

    return HttpResponse(content)
```

Renderizando templates



Refatore

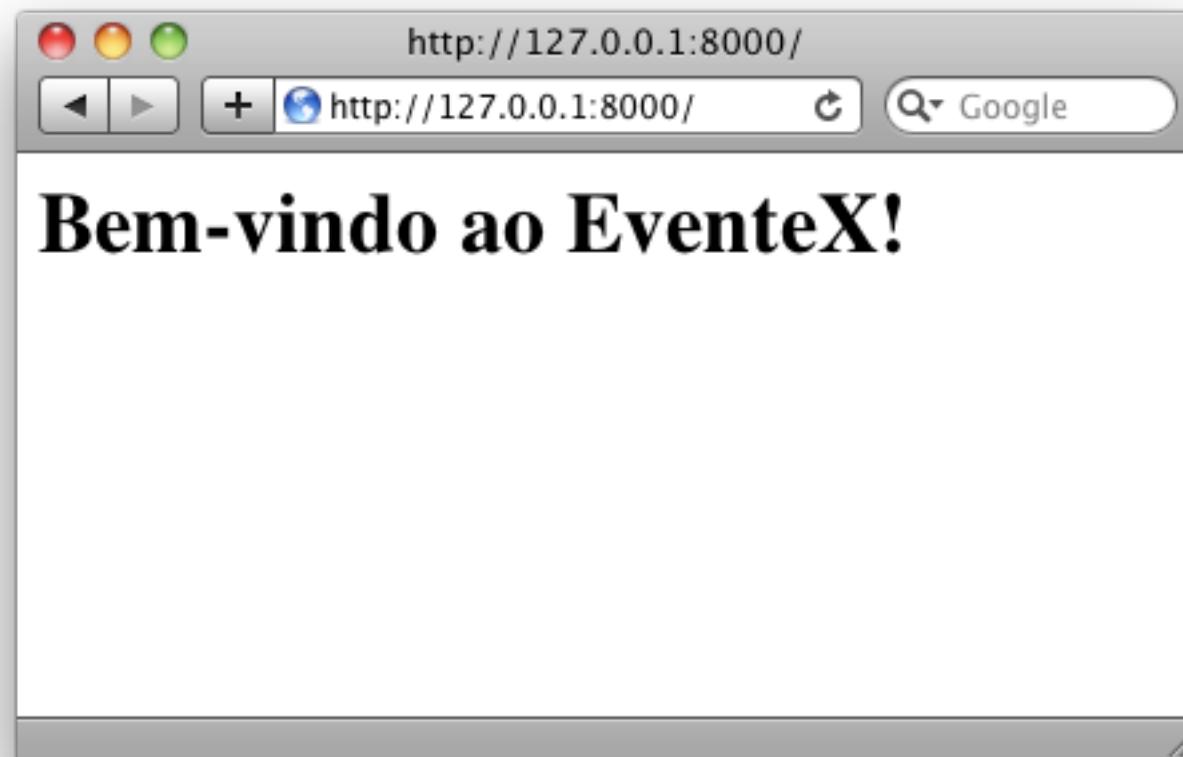
a view homepage

eventex/core/views.py

```
# coding: utf-8
from django.shortcuts import render_to_response

def homepage(request):
    return render_to_response('index.html')
```

Renderizando templates



Como o Django sabe
onde estão os templates?

filesystem Loader

e/ou

appdirectories Loader

Exemplo de settings.py

```
from unipath import Path
PROJECT_DIR = Path(__file__).parent

# ...

TEMPLATE_LOADERS = (
    'django.template.loaders.filesystem.Loader',
    'django.template.loaders.app_directories.Loader',
)

TEMPLATE_DIRS = (
    PROJECT_DIR.child('dir-templates-1'),
    PROJECT_DIR.child('dir-templates-2'),
    # ...
)
```

Por exemplo, suponha...

wtdt/

eventex/

...

templates/

index.html

core/

...

templates/

about.html

core/

contato.html

subscriptions/

...

templates/

subscriptions/

subscription_form.html

```
wttd/
eventex/
...
templates/          # filesystem.Loader
    index.html
core/
...
templates/          # app_directories.Loader
    about.html
core/
    contato.html
subscriptions/
...
templates/          # app_directories.Loader
    subscriptions/
        subscription_form.html
```

Lista de Templates como o Django detecta

index.html

about.html

core/contato.html

subscriptions/subscription_form.html

Anatomia do Template

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head><title>EventEx</title></head>
<body>
    
    <h1>Parabéns {{ subscription.name }}!</h1>
    <p>
        Sua inscrição foi realizada em
        {{ subscription.created_at|date:"d/m/Y" }}.
    </p>
    <p>
        Em breve, alguém entrará em contato pelo
        {% if subscription.phone %}
            telefone {{ subscription.phone }}.
        {% else %}
            e-mail {{ subscription.email }}.
        {% endif %}
    </p>
</body>
</html>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head><title>Eventex</title></head>
<body>
    
    <h1>Parabéns {{ subscription.name }}!</h1>
    <p>
        Sua inscrição foi realizada em
        {{ subscription.created_at|date:"d/m/Y" }}.
    </p>
    <p>
        Em breve, alguém entrará em contato pelo
        {% if subscription.phone %}
            telefone {{ subscription.phone }}.
        {% else %}
            e-mail {{ subscription.email }}.
        {% endif %}
    </p>
</body>
</html>
```

Template **Variables**

O ponto-chave

`{{ subscription.name }}`

1. `subscription["name"]`
2. `subscription.name`
3. `subscription.name()`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head><title>Eventex</title></head>
<body>
    
    <h1>Parabéns {{ subscription.name }}!</h1>
    <p>
        Sua inscrição foi realizada em
        {{ subscription.created_at|date:"d/m/Y" }}.
    </p>
    <p>
        Em breve, alguém entrará em contato pelo
        {% if subscription.phone %}
            telefone {{ subscription.phone }}.
        {% else %}
            e-mail {{ subscription.email }}.
        {% endif %}
    </p>
</body>
</html>
```

Template **Filters**

```
{ { subscription.created_at|date:"d/m/Y" } }
```

```
{ { text|escape|linebreaks } }
```

```
{ { text|truncatewords:"30" } }
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head><title>Eventex</title></head>
<body>
    
    <h1>Parabéns {{ subscription.name }}!</h1>
    <p>
        Sua inscrição foi realizada em
        {{ subscription.created_at|date:"d/m/Y" }}.
    </p>
    <p>
        Em breve, alguém entrará em contato pelo
        {% if subscription.phone %}
            telefone {{ subscription.phone }}.
        {% else %}
            e-mail {{ subscription.email }}.
        {% endif %}
    </p>
</body>
</html>
```

Template **Tags**

<https://docs.djangoproject.com/en/1.4/ref/templates/builtins/>

Dá para botar
uma logo e
deixar bonitinho?



```
# Estando no diretório wttd/eventex  
  
# Crie o diretório wttd/eventex/core/static  
$ mkdir core/static  
  
# Crie os subdiretórios para os assets  
$ mkdir core/static/{img,css,js}
```

```
# Crie uma logo e um css
```

```
# Adicione uma logo ao projeto em
```

```
$ cp umalogo.png core/static/img/logo.png
```

```
# Crie o stylesheet do site
```

```
$ cp somestyle.css core/static/css/style.css
```

```
# Ou copie... -.-
```

CSS: <http://bit.ly/css-eventex>

Logo: <http://bit.ly/logo-do-eventex>

Html: <http://bit.ly/html-landingpage>

Implemente o HTML
da landingpage

eventex/core/templates/index.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
<head>
  <title>EventeX</title>
  <link type="text/css" href="{{ STATIC_URL }}css/style.css"
        rel="stylesheet" media="screen" />
</head>
<body>
  
  <h1>Bem-vindo ao EventeX!</h1>
  <p>O evento mais sensacional do ano depois do Dev in Rio!</p>
</body>
</html>
```

The screenshot shows a web browser window with the following details:

- Title Bar:** The title is "EventeX".
- Address Bar:** The URL is "http://127.0.0.1:8000/".
- Toolbar:** Standard browser controls for back, forward, and search.
- Content Area:**
 - A large heading **Bem-vindo ao EventeX!**
 - Text: "O evento mais sensacional do ano depois do Dev in Rio!"
 - Text: "Você não pode perder esta oportunidade!"
 - Text: "Abertura das inscrições em **21/04/2013**.
 - Text: "Para dúvidas e mais informações entre em contato com eventex@eventex.com.br"
- Footer:** Copyright notice: "© 2013. [EventeX](#), some rights reserved." followed by a Creative Commons license logo: CC BY-NC-SA.

O que deu errado?

```
$ manage runserver  
Validating models...
```

```
0 errors found  
Django version 1.5.1, using settings 'eventex.settings'  
Development server is running at http://127.0.0.1:8000/  
Quit the server with CONTROL-C.
```

```
[01/Jan/0001 00:00:07] "GET / HTTP/1.1" 200 336  
[01/Jan/0001 00:00:08] "GET /css/style.css HTTP/1.1" 404 1949  
[01/Jan/0001 00:00:08] "GET /img/logo.png HTTP/1.1" 404 1943
```

O que temos?

eventex/core/views.py

```
# coding: utf-8
from django.shortcuts import render_to_response

def homepage(request):
    return render_to_response('index.html')
```

Refatore

Indique o contexto ao render do template.

eventex/core/views.py

```
# coding: utf-8
from django.shortcuts import render_to_response
from django.conf import settings

def homepage(request):
    context = {'STATIC_URL': settings.STATIC_URL}
    return render_to_response('index.html', context)
```

A screenshot of a web browser window titled "EventeX". The address bar shows the URL "http://welcometothedjango.herokuapp.com/". The page content features a large, stylized logo "EventeX" in blue and white. Below the logo, the text "Bem-vindo ao EventeX!" is displayed in a large, bold, dark blue font. A subtitle follows: "O evento mais sensacional do ano depois do Dev in Rio!". Another line of text reads: "Você não pode perder esta oportunidade!". A bold statement "Abertura das inscrições em 21/04/2013." is present. At the bottom, there is contact information: "Para dúvidas e mais informações entre em contato com eventex@eventex.com.br". The footer contains the copyright notice "© 2013. EventeX, some rights reserved." followed by a Creative Commons BY-NC-SA license logo. On the right side of the footer, there is a "django SITE" badge.

EventeX

http://welcometothedjango.herokuapp.com/ Google

EventeX

Bem-vindo ao EventeX!

O evento mais sensacional do ano depois do Dev in Rio!

Você não pode perder esta oportunidade!

Abertura das inscrições em 21/04/2013.

Para dúvidas e mais informações entre em contato com eventex@eventex.com.br

© 2013. EventeX, some rights reserved.

DRY

Use o RequestContext!

eventex/core/views.py

```
# coding: utf-8
from django.shortcuts import render_to_response
from django.template import RequestContext

def homepage(request):
    context = RequestContext(request)
    return render_to_response('index.html', context)
```

EventeX

<http://welcometothedjango.herokuapp.com/> Google

EventeX

Bem-vindo ao EventeX!

O evento mais sensacional do ano depois do Dev in Rio!

Você não pode perder esta oportunidade!

Abertura das inscrições em 21/04/2013.

Para dúvidas e mais informações entre em contato com eventex@eventex.com.br

© 2013. EventeX, some rights reserved.

SITE

DRY++

GenericViews

eventex/core/views.py

```
# coding: utf-8
from django.shortcuts import render

def homepage(request):
    return render(request, 'index.html')
```

EventeX

+ <http://welcometothedjango.herokuapp.com/> ⌂ Google

EventeX

Bem-vindo ao EventeX!

O evento mais sensacional do ano depois do Dev in Rio!

Você não pode perder esta oportunidade!

Abertura das inscrições em 21/04/2013.

Para dúvidas e mais informações entre em contato com eventex@eventex.com.br

© 2013. EventeX, some rights reserved.

django SITE

Mídias estáticas

css js png jpg ico etc

Media **vs** Static

`{{ MEDIA_URL }}` **vs** `{{ STATIC_URL }}`

Como o Django sabe
onde estão os **assets**?

EventeX

http://127.0.0.1:8000/ Google

EventeX

Elements Resources Network Scripts Timeline Profiles Search Elements

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <div id="page">
      <div id="header">
        <center>
          
        </center>
      </div>
      <hr>
      <div id="content">...</div>
      <hr>
      <div id="footer">...</div>
    </body>
</html>
```

Computed Style

Styles

```
element.style {
```

}

Matched CSS Rules

```
img { style.css:34
  border: 0;
}
```

Inherited from center

```
center { user agent stylesheet
  text-align: -webkit-center;
}
```

Inherited from div#page

```
div#page { style.css:17
  text-align: center;
}
```

http:// 127.0.0.1:8000 /static/img/logo.png

html body div#page div#header center img

Acesso à um asset

- 1) GET /static/img/logo.png
- 2) url(r'^static/(?P<path>.*\$',
 'django.contrib.staticfiles.views.serve')
- 3) serve(request, path='img/logo.png')
- 4) HttpResponse

Acesso à um asset

- 1) GET /static/img/logo.png 
- 2) url(r'^static/(?P<path>.*\$',
 'django.contrib.staticfiles.views.serve')
- 3) serve(request, path='img/logo.png')
- 4) HttpResponse  STATIC_ROOT

http://bit.ly/staticfiles

```
#####
# STATICFILES #
#####

# A list of locations of additional static files
STATICFILES_DIRS = ()

# The default file storage backend used during the build process
STATICFILES_STORAGE = 'django.contrib.staticfiles.storage.StaticFilesStorage'

# List of finder classes that know how to find static files in
# various locations.
STATICFILES_FINDERS = (
    'django.contrib.staticfiles.finders.FileSystemFinder',
    'django.contrib.staticfiles.finders.AppDirectoriesFinder',
#     'django.contrib.staticfiles.finders.DefaultStorageFinder',
)
```

Onde posso ver
o que já foi feito?



“ “

É a entrega que faz o
sucesso do orador.

, ,

Johann Wolfgang von Goethe

Banco de Dados e 12 Factors

```
# Instale a lib dj-database-url  
$ pip install dj-database-url==0.2.1
```

https://devcenter.heroku.com/articles/django-injection#recommended_settings

eventex/settings.py

...

```
import dj_database_url
```

...

```
DATABASES = {
    'default': dj_database_url.config(
        default='sqlite:///{} + PROJECT_DIR.child('database.db'))}
```

Deploy de mídias estáticas

#comofaz?

```
# Estando no diretório wttd/eventex
```

```
# Crie o diretório static  
$ mkdir static
```

```
# Crie o diretório media  
$ mkdir media
```

eventex/settings.py

```
from unipath import Path
PROJECT_DIR = Path(__file__).parent

# ...

MEDIA_ROOT = PROJECT_DIR.child('media')
MEDIA_URL = '/media/'

STATIC_ROOT = PROJECT_DIR.child('static')
STATIC_URL = '/static/'

# ...
```

Wsgi de Produção

```
# Instale o gunicorn  
$ pip install gunicorn==0.17.2
```

<http://gunicorn.org/>

Procfile

```
# Crie o arquivo texto wttd/Procfile
```

```
web: gunicorn eventex.wsgi
```

Informe as
dependências
do Projeto

wttd/requirements.txt

django==1.5.1

dj-database-url==0.2.1

Unipath==1.0

gunicorn==0.17.2

psycopg2

Repositório de Código

#comofaz?

Tutorial de Git

<http://blip.tv/akitaonrails/screencast-come-ando-com-git-6074964>

Preparando Repositório

```
# Configurando seu Git
```

```
$ git config --global user.name "Seu Nome"  
$ git config --global user.email "seu@email.com"
```

```
# Iniciando o repositório
```

```
$ cd wttd  
$ git init
```

Ignorando Arquivos

wttd/.gitignore

```
bin  
build  
include  
lib  
local  
man  
share  
Scripts  
.Python  
*.pyc  
*~  
*.db  
.DS_Store
```

Commit Inicial

```
$ git add .gitignore
```

```
$ git commit -m "Ignora arqs que nao sao do projeto"
```

```
$ git status
```

```
# On branch master
```

```
# Untracked files:
```

```
#   (use "git add <file>..." to include in what will be committed)
```

```
#
```

```
# eventex/
```

```
# manage.py
```

```
# Procfile
```

```
# requirements.txt
```

Grave o projeto
no repositório

Projeto no Git

```
# No diretório wttd  
  
$ git add .  
$ git commit -m "All your base are belong to us"  
  
$ git status  
# On branch master  
nothing to commit (working directory clean)
```

[How it Works](#) | [Pricing](#) | [Add-ons](#) | [Dev Center](#) | [Support](#) | [Contact](#)[Login](#)

cloud application platform

deploy and scale powerful apps

[Forget Servers](#)[Run Anything NEW](#)[See Everything NEW](#)[Trust & Manage](#)

Agile deployment for Ruby, Node.js, Clojure, Java, Python, and Scala.

Get up and running in minutes, and deploy instantly with git. Focus 100% on your code, and never think about servers, instances, or VMs again.

[Agile Deployment on Heroku](#)

```
$ heroku create  
Created sushi.herokuapp.com | git@heroku.com:sushi.git
```

```
$ git push heroku master  
-----> Heroku receiving push  
-----> Rails app detected  
-----> Compiled slug size is 8.0MB  
-----> Launching... done, v1  
http://sushi.herokuapp.com deployed to Heroku
```

[How it Works](#)

It's free to get started and sign up is instant.

[Sign Up](#)

Heroku Auth

```
# Inicie uma sessão com o Heroku Toolbelt
```

```
$ heroku login
```

```
Enter your Heroku credentials.
```

```
Email: henrique@bastos.net
```

```
Password:
```

```
Could not find an existing public key.
```

```
Would you like to generate one? [Yn]
```

```
Generating new SSH public key.
```

```
Uploading ssh public key /Users/henrique/.ssh/id_rsa.pub
```

Heroku Stack

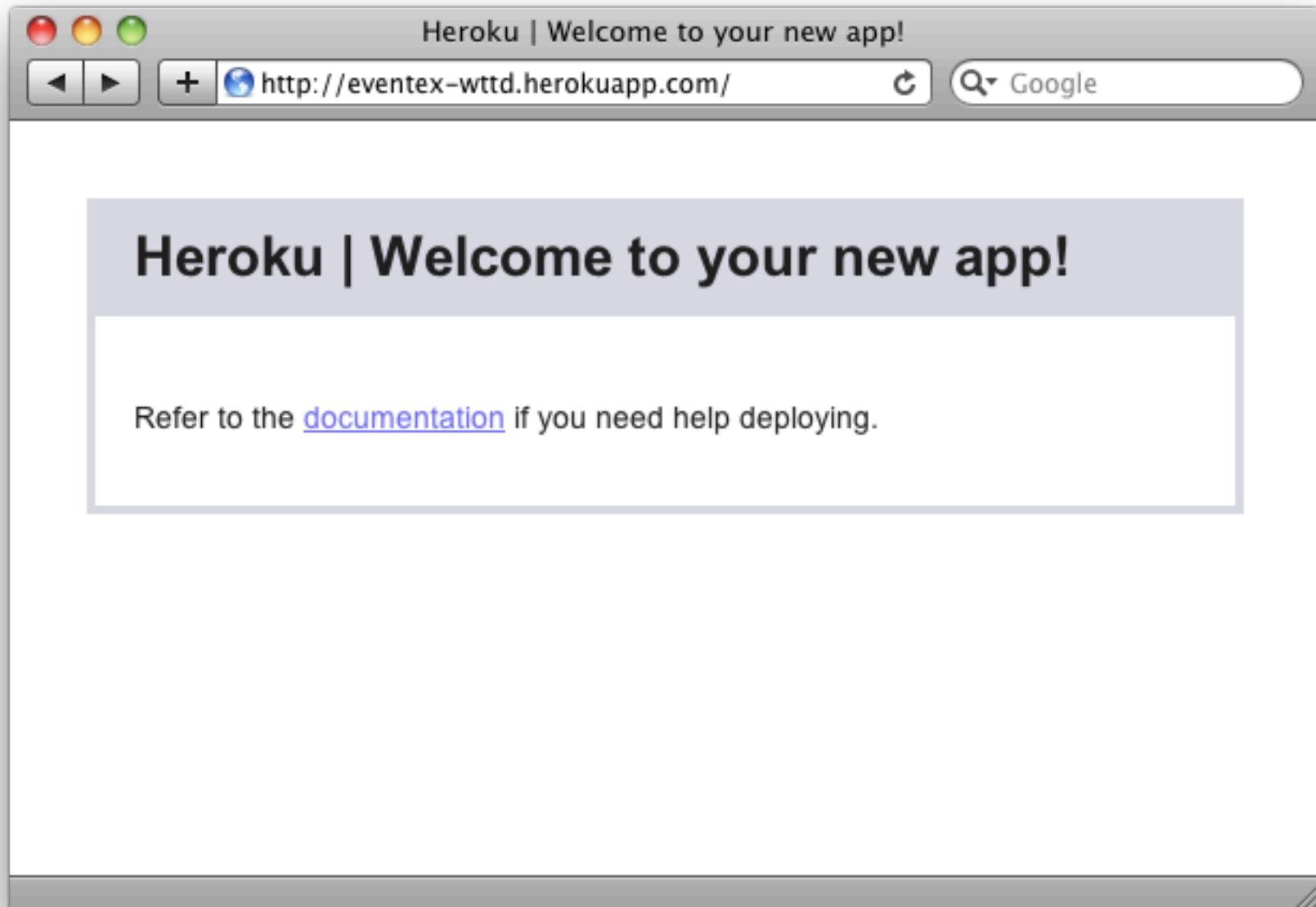
```
# Crie uma stack no Heroku
```

```
$ heroku create NOME_DA_APP --stack cedar
Creating NOME_DA_APP... done, stack is cedar
Git remote heroku added
```

Heroku

Veja sua stack no browser

\$ heroku open

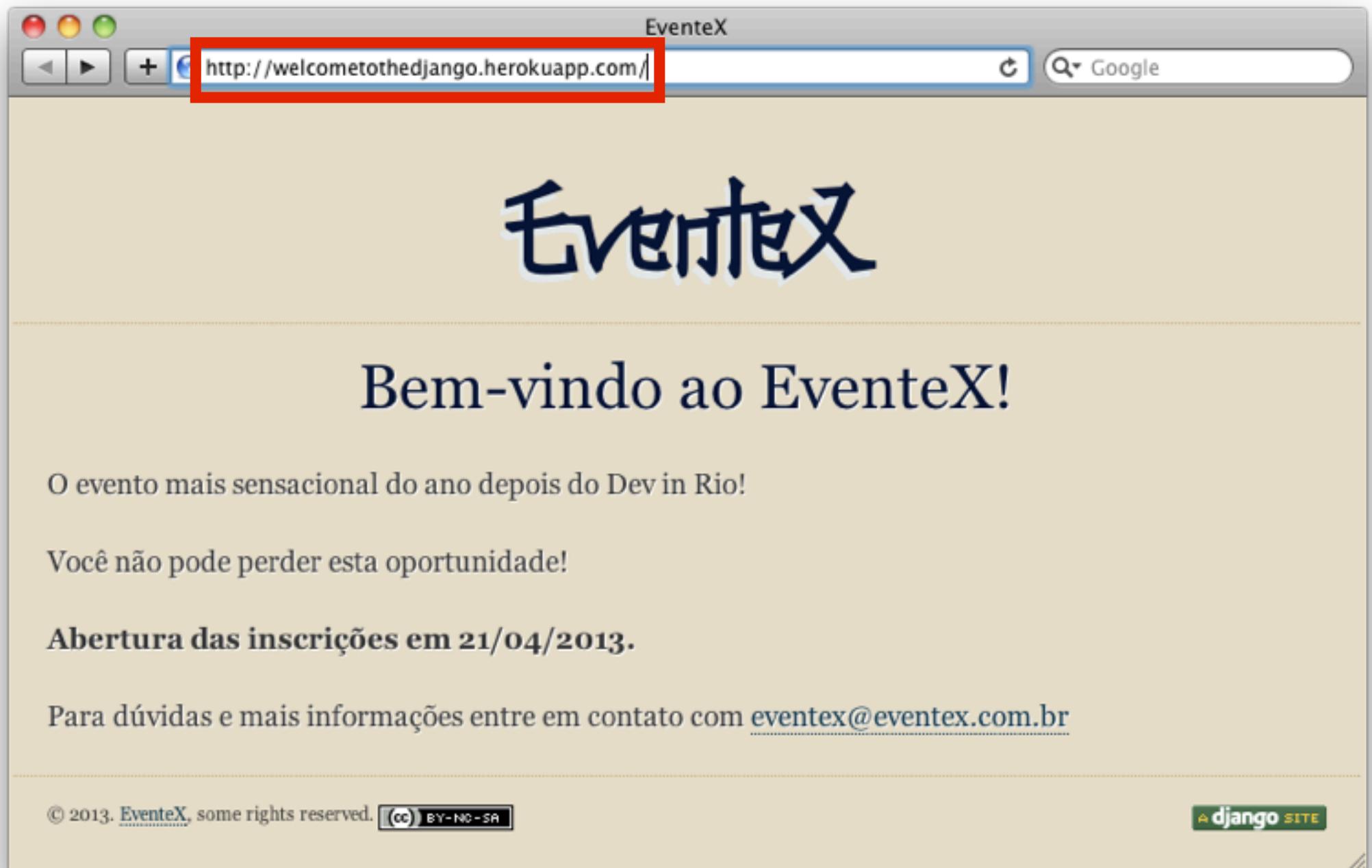


No ar!

```
$ git push heroku master --force
```

...

```
-----> Heroku receiving push
-----> Python/Django app detected
-----> Preparing virtualenv version 1.6.4
          New python executable in ./bin/python
          Installing setuptools.....done.
          Installing pip.....done.
-----> Django settings injection
          Injecting code into eventex/settings.py to read from
DATABASE_URL
-----> Installing dependencies using pip version 1.0.2
```





Dica

Como o comando
heroku sabe das coisas?

Não é feitiçaria!

Dica

```
$ git remote -v
```

```
heroku git@heroku.com:welcome-to-the-django.git (fetch)  
heroku git@heroku.com:welcome-to-the-django.git (push)
```

O que é essa
página amarela?





Page not found (404)

Request Method: GET

Request URL: http://welcometothedjango.herokuapp.com/p%C3%A1gina-que-n%C3%A3o-existe

Using the URLconf defined in eventex.urls, Django tried these URL patterns, in this order:

1. ^\$ [name='homepage']

The current URL, página-que-não-existe, didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.

eventex/settings.py

```
# Django settings for eventex project.
import os
import dj_database_url
from unipath import Path
PROJECT_DIR = Path(__file__).parent

DEBUG = os.environ.get('DEBUG') == 'True'
TEMPLATE_DEBUG = DEBUG
```

```
$ DEBUG=True manage runserver
```

Validating models...

0 errors found

Django version 1.5.1, using settings 'eventex.settings'

Development server is running at <http://127.0.0.1:8000/>

Quit the server with CONTROL-C.

```
[01/Jan/2013 00:00:00] "GET / HTTP/1.1" 200 1644
```

```
[01/Jan/2013 00:00:00] "GET /static/css/style.css HTTP/1.1" 200 3859
```

```
[01/Jan/2013 00:00:00] "GET /static/img/logo.png HTTP/1.1" 200 12189
```

The screenshot shows a web browser window with the title bar "EventeX". The address bar contains the URL "http://127.0.0.1:8000/". To the right of the address bar are standard browser controls: back, forward, and search, along with a "Google" button. The main content area of the browser displays the "EventeX" logo in a large, stylized, blue font. Below the logo, the text "Bem-vindo ao EventeX!" is displayed in a large, bold, dark blue font. Underneath this, there are two lines of text: "O evento mais sensacional do ano depois do Dev in Rio!" and "Você não pode perder esta oportunidade!". A bold, dark blue line of text follows: "Abertura das inscrições em 21/04/2013.". At the bottom of the page, there is a footer with the text "© 2013. EventeX, some rights reserved." and a small "CC BY-NC-SA" license logo.

EventeX

http://127.0.0.1:8000/ Google

EventeX

Bem-vindo ao EventeX!

O evento mais sensacional do ano depois do Dev in Rio!

Você não pode perder esta oportunidade!

Abertura das inscrições em 21/04/2013.

Para dúvidas e mais informações entre em contato com eventex@eventex.com.br

© 2013. EventeX, some rights reserved.

```
$ manage runserver
```

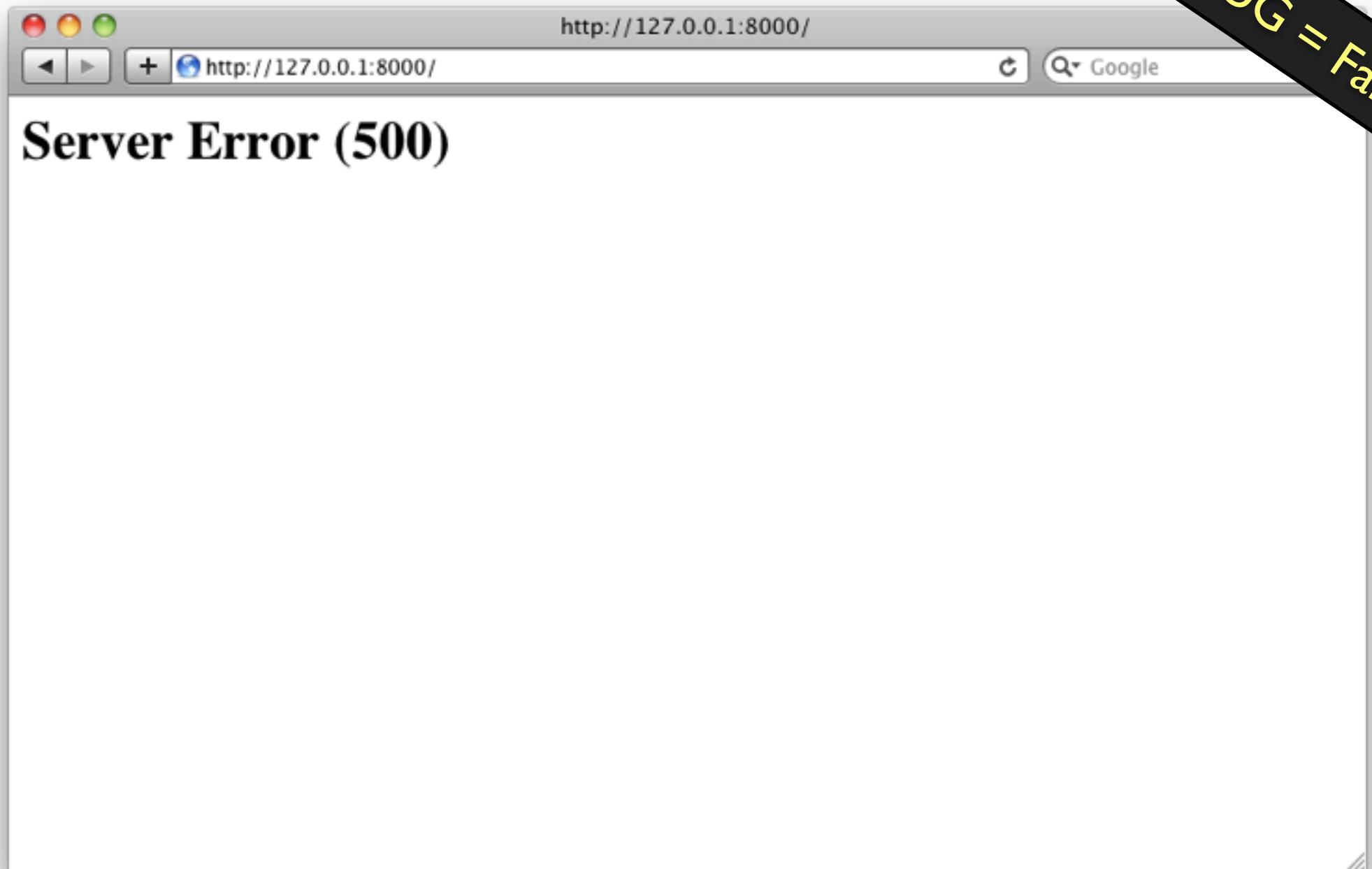
Validating models...

0 errors found

Django version 1.5.1, using settings 'eventex.settings'
Development server is running at <http://127.0.0.1:8000/>
Quit the server with CONTROL-C.

[01/Jan/2013 00:00:00] "GET / HTTP/1.1" 500 27

DEBUG = False



eventex/settings.py

```
LOGGING = {
    'version': 1,
    'disable_existing_loggers': False,
    'filters': {
        ...
    },
    'handlers': {
        'mail_admins': {
            ...
        },
        'console':{
            'level': 'DEBUG',
            'class': 'logging.StreamHandler',
        },
    },
    'loggers': {
        'django.request': {
            'handlers': ['mail_admins', 'console'],
            ...
        },
    }
}
```

DEBUG = False

```
$ manage runserver
```

Validating models...

0 errors found

Django version 1.5.1, using settings 'eventex.settings'
Development server is running at <http://127.0.0.1:8000/>
Quit the server with CONTROL-C.

Internal Server Error: /

Traceback (most recent call last):

...

SuspiciousOperation: Invalid HTTP_HOST header (you may need to set
ALLOWED_HOSTS): 127.0.0.1:8000

[01/Jan/2013 00:00:00] "GET / HTTP/1.1" 500 27



ALLOWED_HOSTS

Default: [] (Empty list)

A list of strings representing the host/domain names that this Django site can serve. This is a security measure to prevent an attacker from poisoning caches and password reset emails with links to malicious hosts by submitting requests with a fake HTTP `Host` header, which is possible even under many seemingly-safe webserver configurations.

Values in this list can be fully qualified names (e.g. '`www.example.com`'), in which case they will be matched against the request's `Host` header exactly (case-insensitive, not including port). A value beginning with a period can be used as a subdomain wildcard: '`.example.com`' will match `example.com`, `www.example.com`, and any other subdomain of `example.com`. A value of '*' will match anything; in this case you are responsible to provide your own validation of the `Host` header (perhaps in a middleware; if so this middleware must be listed first in `MIDDLEWARE_CLASSES`).

If the `Host` header (or `X-Forwarded-Host` if `USE_X_FORWARDED_HOST` is enabled) does not match any value in this list, the `django.http.HttpRequest.get_host()` method will raise `SuspiciousOperation`.

When `DEBUG` is `True` or when running tests, host validation is disabled; any host will be accepted. Thus it's usually only necessary to set it in production.

This validation only applies via `get_host()`; if your code accesses the `Host` header directly from `request.META` you are bypassing this security protection.

- `TEST_MIRROR`
- `TEST_NAME`
- `TEST_CREATE`
- `TEST_USER`
- `TEST_USER_CREATE`
- `TEST_PASSWD`
- `TEST_TBLSPACE`
- `TEST_TBLSPACE_TMP`
- `DATABASE_ROUTERS`
- `DATE_FORMAT`
- `DATE_INPUT_FORMATS`
- `DATETIME_FORMAT`
- `DATETIME_INPUT_FORMATS`
- `DEBUG`
- `DEBUG_PROPAGATE_EXCEPTIONS`
- `DECIMAL_SEPARATOR`
- `DEFAULT_CHARSET`
- `DEFAULT_CONTENT_TYPE`
- `DEFAULT_EXCEPTION_REPORTER`

◀ Documentation version: 1.5 ▶

eventex/settings.py

```
ALLOWED_HOSTS = ['.localhost', '127.0.0.1', '.herokuapp.com']
```

DEBUG = False

The screenshot shows a web browser window with the following details:

- Title Bar:** EventeX
- Address Bar:** http://127.0.0.1:8000/
- Toolbar:** Back, Forward, Stop, Refresh, Home, and a search bar with Google.
- Content Area:**
 - # Bem-vindo ao EventeX!
 - O evento mais sensacional do ano depois do Dev in Rio!
 - Você não pode perder esta oportunidade!
 - Abertura das inscrições em 21/04/2013.**
 - Para dúvidas e mais informações entre em contato com eventex@eventex.com.br
- Footer:** © 2013. [EventeX](#), some rights reserved.

eventex/urls.py

```
from django.conf.urls import patterns, include, url

urlpatterns = patterns('eventex.core.views',
    url(r'^$', 'homepage', name='homepage'),
)

from django.conf import settings
urlpatterns += patterns('django.views.static',
    url(r'^static/(?P<path>.*)$', 'serve',
        {'document_root': settings.STATIC_ROOT}),
)
```

```
$ manage collectstatic --noinput
Copying '/wttd/eventex/core/static/css/style.css'
Copying '/wttd/eventex/core/static/img/logo.png'
```

2 static files copied.

```
$ manage runserver
Validating models...
```

0 errors found
Django version 1.5.1, using settings 'eventex.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

```
[01/Jan/2013 00:00:00] "GET / HTTP/1.1" 200 1644
[01/Jan/2013 00:00:00] "GET /static/css/style.css HTTP/1.1" 200 3859
[01/Jan/2013 00:00:00] "GET /static/img/logo.png HTTP/1.1" 200 12189
```

DEBUG = False

A screenshot of a web browser window titled "EventeX". The address bar shows the URL "http://127.0.0.1:8000/". The page content features a large, stylized logo "EventeX" in blue and white. Below the logo, the text "Bem-vindo ao EventeX!" is displayed in a large, bold, dark blue font. A message follows: "O evento mais sensacional do ano depois do Dev in Rio! Você não pode perder esta oportunidade!" In bold text, it says "Abertura das inscrições em 21/04/2013.". At the bottom, there is contact information: "Para dúvidas e mais informações entre em contato com eventex@eventex.com.br". The footer contains the copyright notice "© 2013. EventeX, some rights reserved." and a Creative Commons license logo "(cc) BY-NC-SA".

EventeX

http://127.0.0.1:8000/ Google

EventeX

Bem-vindo ao EventeX!

O evento mais sensacional do ano depois do Dev in Rio!
Você não pode perder esta oportunidade!

Abertura das inscrições em 21/04/2013.

Para dúvidas e mais informações entre em contato com eventex@eventex.com.br

© 2013. EventeX, some rights reserved. [\(cc\) BY-NC-SA](#)

Onde vejo a
correção?



```
$ echo "eventex/static" >> .gitignore

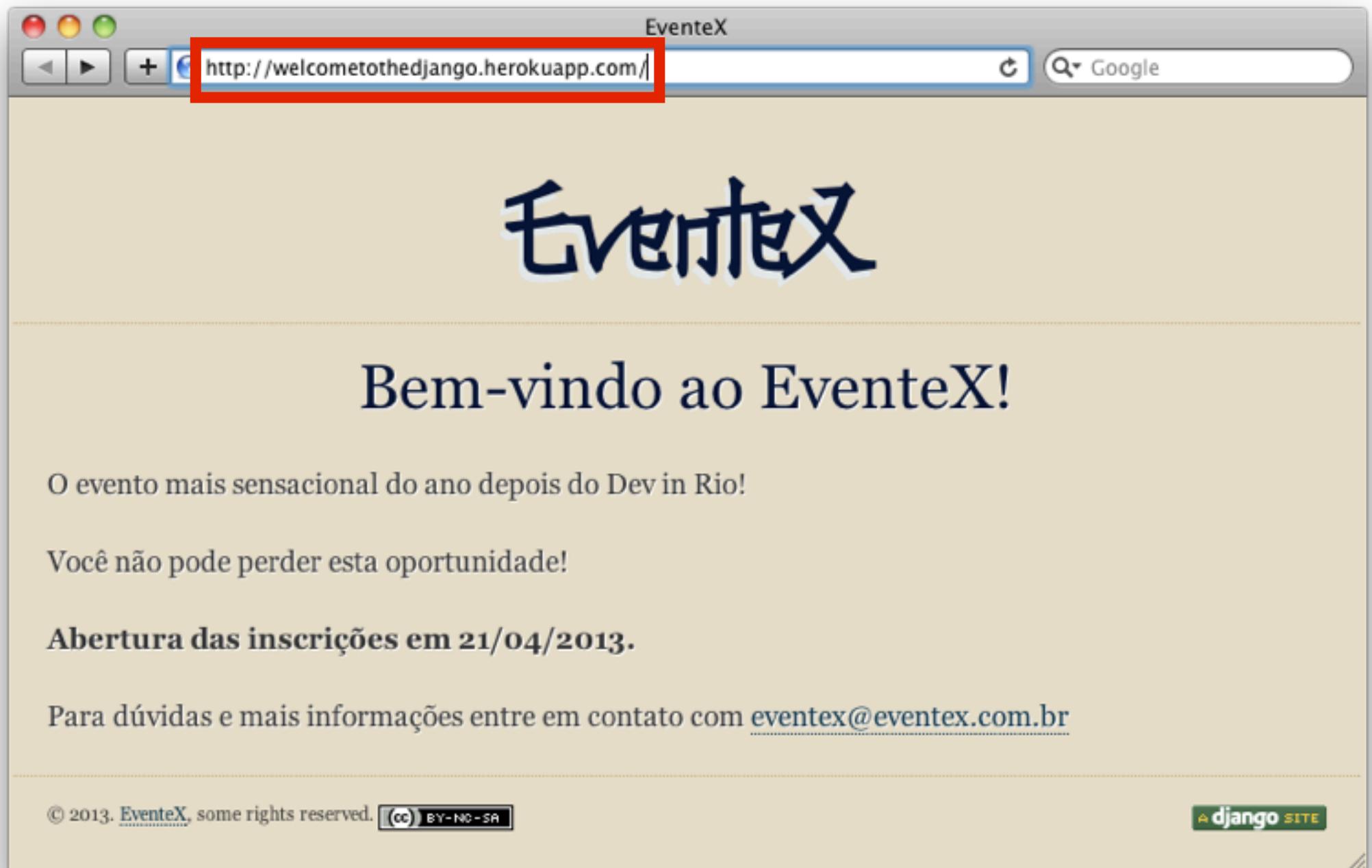
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   .gitignore
#       modified:   eventex/settings.py
#       modified:   eventex/urls.py
#
no changes added to commit (use "git add" and/or "git commit -a")

$ git commit -am 'Debug false'
[master 225b2d7] Debug false
 3 files changed, 15 insertions(+), 4 deletions(-)

$ git status
# On branch master
nothing to commit (working directory clean)

$ git push heroku master --force
```





Como liga o DEBUG remoto?

```
# Para ligar...
```

```
$ heroku config:add DEBUG=True --app welcometothedjango
Setting config vars and restarting welcometothedjango... done, v12
DEBUG: True
```

```
# Para desligar...
```

```
$ heroku config:add DEBUG=False --app welcometothedjango
Setting config vars and restarting welcometothedjango... done, v13
DEBUG: False
```

Compartilhe!

- ➔ Crie uma conta no Github.com
- ➔ Adicione sua chave ssh à sua conta
- ➔ Crie um repositório remoto
- ➔ Publique o seu código do Eventex no Github.
- ➔ Envie p/ a lista o link do seu *repo* no Github!!!
- ➔ Envie p/ a lista o link da sua app no Heroku!!!