

04 o que não te contaram sobre django



Settings

Existem algumas fraquezas em usar um conjunto de settings globais para todos os ambientes.

3 hacks para melhorar seu settings.py

PROJECT_PATH

settings_local.py

auto debug

PROJECT_PATH

A criação dinamica de um PROJECT_PATH facilita a execução em multiplos ambientes

```
import os  
import sys
```

```
PROJECT_PATH = os.path.abspath(os.path.split(__file__)[0])  
...  
MEDIA_ROOT = os.path.join(PROJECT_PATH, 'media')
```

settings_local.py

se existir settings_local.py executa ela

```
# ultima coisa no settings
try:
    execfile(PROJECT_PATH+'/settings_local.py')
    #print 'Usando configuracao LOCAL'
except IOError:
    #print 'Usando configuracao PADRAO'
    pass
```

auto debug

O menos aconselhavel das 3 modificações

```
import socket

# Set DEBUG = True if on the production server
if socket.gethostname() == 'your.domain.com':
    DEBUG = False
else:
    DEBUG = True

TEMPLATE_DEBUG = DEBUG
```

Bonus hack

no settings.py

```
TEMPLATE_TAGS = (  
    'projeto.foo.templatetags.foobar',  
    'projeto.bar.templatetags.foobar',  
)
```

autoload template_tags

no __init__.py

```
from django.conf import settings
from django.template import add_to_builtins
#import django.template.loader

try:
    for lib in settings.TEMPLATE_TAGS:
        add_to_builtins(lib)
except AttributeError:
    pass
```

Customizando o Admin

criar o diretorio de templates especial pro admin

```
mkdir -p templates/admin
```

```
cd templates/admin
```


base_site.html

```
{% extends "admin/base_index.html" %}

{% block title %}{{ title }} | Admin da pizzeria {% endblock %}

{% block branding %}
<h1 id="site-name" > Pizzeria</h1>
{% endblock %}
```

existe ainda um `{% block extrastyle %}` para colocar styles de css

OBS: incluir tambem o `<style>` e `</style>`

Aplicações para conhecer

`django-config` Gerencia varias configuracoes

`django-debug-toolbar` Fantastico!

`haystack` Facilita search

`sphinx` Grande ferramenta de documentação

`celery` Controle de tarefas assincronamente

`fabric` Deploy e gerenciamento remoto

`gunicorn` Servidor WSGI

`varnish` Solucao para cache