

# Deploying Django

Brian Rosner  
DjangoCon 2009

# About me

- core developer for Django and Pinax
- work at Eldarion
- host of Django Dose

# Overview

- best practices
- tools
- deployment options
- distributed deployment system

# Deployment should be...

- repeatable
- automated
- isolated

# Project best practices

django-admin.py startproject

# relative paths in settings.py

```
PROJECT_ROOT = os.path.abspath(  
    os.path.dirname(__file__)  
)
```



```
MEDIA_ROOT = os.path.join(  
    PROJECT_ROOT,  
    "site_media",  
    "media",  
)
```

```
TEMPLATE_DIRS = (  
    os.path.join(  
        PROJECT_ROOT,  
        "templates",  
    ),  
)
```

use PROJECT\_ROOT  
everywhere

```
from django.conf import settings  
settings.PROJECT_ROOT
```

you can even use it in  
manage.py

# manage.py

```
from django.conf import settings
from django.core.management import \
    setup_environ, execute_from_command_line

import settings as settings_mod

# setup the environment before we access settings
setup_environ(settings_mod)

settings.PROJECT_ROOT

if __name__ == "__main__":
    execute_from_command_line()
```



\* keep project directory  
clean of user files

separate file uploads  
root/URL from static  
media root/URL

MEDIA\_ROOT /  
MEDIA\_URL is used by  
FileField




create two new settings  
STATIC\_ROOT /  
STATIC\_URL

\* `manage.py` sets up the `sys.path` for you. puts project and parent dir on it.

\* project parent dir will always need to be on `sys.path`. you'll need it.

# pay attention to the `sys.path`



- \* using CVS or SVN i feel bad for you
- \* makes it easier for rollback even when deployment tools don't support it

# tag each deployment in VCS

\* when you are rolling  
back you will thank  
yourself for having  
written the down  
migration

use South or  
migrations that allow  
for up and down

# Tools

# virtualenv

# isolated Python environments

# dependancy / version control



# Structure

Global site-packages

virtualenv

virtualenv

`virtualenv --no-site-packages`

use virtualenvwrapper

workon  
mkvirtualenv  
rmvirtualenv

pip

`pip install -r`



PIP\_VIRTUALENV\_BASE

pip install -E env -r

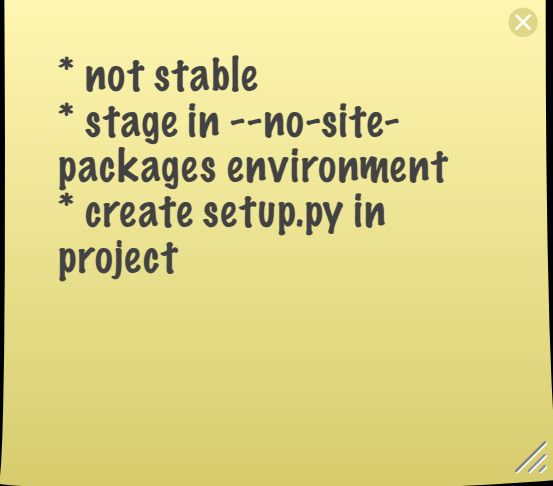
# pip requirement file

```
--requirement=anotherfile.txt  
--find-links=http://packages.a.com
```

```
Django>=1.1  
django-tagging==0.3
```



# pip freeze



- \* not stable
- \* stage in --no-site-packages environment
- \* create setup.py in project

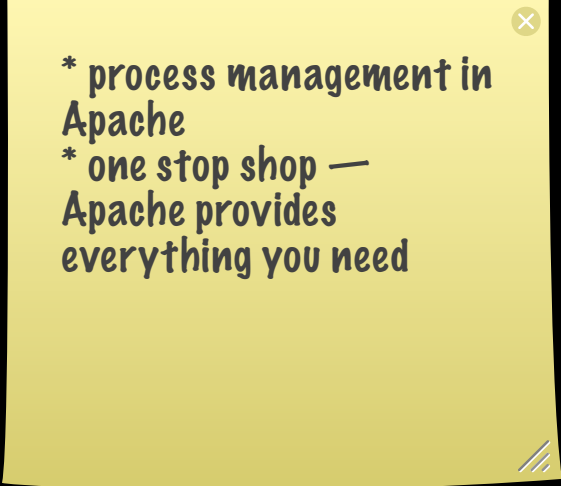
# pip bundle

# Fabric

# Deployment in three lines

```
run("cd project")  
run("git pull")  
run("touch project.wsgi")
```

# Deployment options



- \* process management in Apache
- \* one stop shop — Apache provides everything you need

~~mod\_python~~  
mod\_wsgi

use daemon mode

# sample WSGI script

```
import os
import sys
from os.path import abspath, dirname, join

from django.conf import settings
os.environ["DJANGO_SETTINGS_MODULE"] = "project.settings"

sys.path.insert(0, abspath(join(dirname(__file__), "../..")))
sys.path.insert(0, join(settings.PROJECT_ROOT, "apps"))

from django.core.handlers.wsgi import WSGIHandler
application = WSGIHandler()
```



# FastCGI

manage.py runfcgi

# sample FCGI script

```
import os
import sys
from os.path import abspath, dirname, join

from django.conf import settings
os.environ["DJANGO_SETTINGS_MODULE"] = "project.settings"

sys.path.insert(0, abspath(join(dirname(__file__), "../..")))
sys.path.insert(0, join(settings.PROJECT_ROOT, "apps"))

from django.core.servers.fastcgi import runfastcgi
runfastcgi(method="threaded", daemonize="false")
```

# Jython

# Distributed deployment system

# tolkien

- automated
- repeatable
- isolated
- zero external dependancies

# tolkien internals

- fabric / virtualenv / pip
- gondor-map
- server metadata
- application metadata

# Metadata file

```
from tolkien import packages
```

```
ADMINS = [  
    ("brosner", {"keys": ["<keydata>"]}),  
]
```

```
SSH_PORT = 22
```

```
PACKAGES = [  
    packages.nginx(),  
    packages.apache_proxied(),  
    packages.modwsgi(),  
    packages.git(),  
    packages.subversion(),  
    packages.setuptools(),  
    packages.psycpg2(),  
    packages.python_imaging(),  
    packages.pip(),  
    packages.virtualenv(),  
    packages.virtualenvwrapper(),  
]
```



Thank you!  
@brosner