



# Boîte à outils Django

du développement au déploiement

gestion d'environnements différents

applications Django utiles

environnements Python virtuels

l'installation de bibliothèques

le déploiement facilité

gestion d'environnements différents

applications Django utiles

environnements Python virtuels

l'installation de bibliothèques

le déploiement facilité

# local\_settings

```
## fichier settings.py
try:
    from local_settings import *
except ImportError:
    pass
```

```
## fichier local_settings.py
DEBUG = True
TEMPLATE_DEBUG = DEBUG
DATABASE_ENGINE = 'sqlite3'
DATABASE_NAME = 'db.sqlite'
...
```

gestion d'environnements différents

applications Django utiles

environnements Python virtuels

l'installation de bibliothèques

le déploiement facilité

# django-debug-toolbar

Un ensemble configurable de panneaux affichant différentes informations sur la requête/réponse courante.

# django-debug-toolbar

```
# easy_install django-debug-toolbar
```

```
## fichier local_settings.py
```

```
from settings import *
```

```
MIDDLEWARE_CLASSES += (  
    'debug_toolbar.middleware.DebugToolbarMiddleware',)  
INTERNAL_IPS = ('127.0.0.1',)  
INSTALLED_APPS += ('debug_toolbar',)
```

# django-command-extensions

Une collection de commandes de gestion  
(*management commands*) pour Django



# django-command-extensions

```
$ python manage.py [COMMAND]
```

```
create_superuser
```

```
passwd
```

```
generate_secret_key
```

```
dumpscript
```

```
show_urls
```

```
runserver_plus
```

```
shell_plus
```

# django-command-extensions

```
# easy_install django-extensions
```

```
## pour runserver_plus  
# easy_install werkzeug
```

```
## fichier local_settings.py
```

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

```
INSTALLED_APPS += ('django_extensions',)
```

gestion d'environnements différents

applications Django utiles

environnements Python virtuels

l'installation de bibliothèques

le déploiement facilité

# virtualenv

```
# aptitude install python-django
```

ou

```
# easy_install django
```

ou

```
# easy_install django==1.0.4
```

# virtualenv

```
# easy_install virtualenv
```

```
$ virtualenv [OPTIONS] venv_monprojet
```

```
    --python=PYTHON_EXE
```

```
    --distribute
```

```
    --no-site-packages
```

# virtualenv

```
$ tree -L 2 venv_monprojet
venv_monprojet/
├── bin
│   ├── activate
│   ├── activate_this.py
│   ├── easy_install
│   ├── easy_install-2.5
│   ├── pip
│   └── python
├── include
│   └── python2.5 -> /usr/include/python2.5
└── lib
    └── python2.5
```

# virtualenv

```
$ cd venv_monprojet  
$ source bin/activate
```

```
(venv_monprojet)$ easy_install django
```

ou

```
(venv_monprojet)$ easy_install django==1.0.4
```

```
(venv_monprojet)$ deactivate
```

gestion d'environnements différents

applications Django utiles

environnements Python virtuels

l'installation de bibliothèques

le déploiement facilité



# pip

pip installs packages.  
Python packages.  
An easy\_install replacement.

# pip

```
$ easy_install pip
```

```
$ pip install django
```

```
$ pip install django==1.0.4
```

ou

```
$ pip install -e hg+http://bitbucket.org/mirror/django#egg=django
```

```
$ pip install -e hg+http://bitbucket.org/mirror/django@rev#egg=django
```

# pip

```
$ pip install --upgrade django
```

```
$ pip uninstall django
```

```
$ pip freeze
```

```
$ pip freeze > requirements.txt
```

```
$ pip install -r requirements.txt
```

# virtualenv et pip

```
$ cd venv_monprojet
```

```
$ source bin/activate
```

```
(venv_monprojet)$ pip install django
```

ou

```
$ pip -E venv_monprojet install django
```

gestion d'environnements différents

applications Django utiles

environnements Python virtuels

l'installation de bibliothèques

le déploiement facilité

# fabric

```
## fabfile.py

from __future__ import with_statement
from fabric.api import *
from fabric.operations import sudo
from fabric.context_managers import cd

run('ls /var/www')

local('ls bin/')

put('bin/project.zip', '/tmp/project.zip')

get('/tmp/project.zip', 'bin/project.zip')

sudo('/etc/init.d/apache2 restart')

with cd('/var/www'):
    run('ls')
```

# fabric

```
## fabfile.py
```

```
from __future__ import with_statement
from fabric.api import run, env
from fabric.operations import put, run
from fabric.context_managers import cd
```

```
def production():
    env.hosts = ['hote.domaine.tld',]
    env.user = "www-data"
    env.base_dir = '/home/eric/www'
    env.projet = '%s/projet.tgz' % env.base_dir
    env.remote_dir = '/var/www'
```

```
def deploy():
    put(env.project, env.remote_dir)
    with cd(env.remote_dir):
        run('tar xvzf projet.tgz')
```

# fabric

```
$ fab --list
```

Available commands:

```
    deploy
    upload_project
```

```
$ fab production deploy
```

```
$ fab production upload_project
```

```
$ fab -H localhost,autre.domaine.tld deploy
```



# Liens

<http://github.com/robhudson/django-debug-toolbar>

<http://code.google.com/p/django-command-extensions/>

<http://pypi.python.org/pypi/virtualenv>

<http://pip.openplans.org/>

<http://fabfile.org/>

# Aller plus loin

virtualenv :

- virtualenwrapper
- fichiers de bootstrap

buildout

distribute : <http://guide.python-distribute.org/>

# Me trouver

Eric Veiras Galisson

<http://blog.sietch-tabr.com>

<http://bitbucket.org/daks>

daks sur identi.ca

daks\_ sur twitter

daks sur IRC #django-fr



**Le contenu de cette présentation est sous licence  
Creative Commons Paternité  
(<http://creativecommons.org/licenses/by/2.0/fr/>),  
paternité Eric Veiras Galisson**