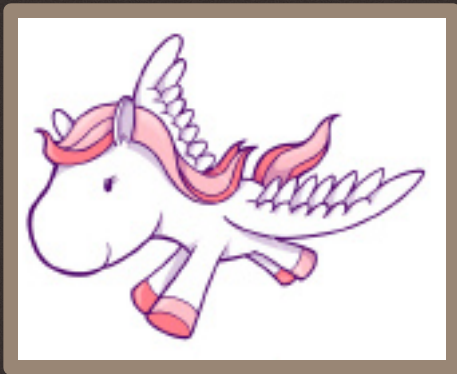


Tester son projet Django



Bruno Renié

@brutasse

<http://bruno.im>



Obey the goat!

Tester son projet Django

- Tests unitaires, principe et avantages
- Comment Django exécute les tests
- Écrire des tests
 - Client et réponses, cas particuliers et bonnes pratiques
- Conseils
 - Couverture, Django 1.2, utilitaires

Test

*Fonction vérifiant le comportement
d'une autre fonction*

```
def add(x, y):  
    return x + y
```

```
def test_add():  
    assert add(1, 2) == 3
```

Suite de test, automatisations

Avantages

- Qualité
- Modularité
- Flexibilité
- \w+ité

Avantages



Avantages



Conditions nécessaires

mais pas suffisantes

- Les tests sont faciles à écrire
- Les tests sont faciles à exécuter

ValueError at /

The view foo.views.index didn't return an HttpResponse object.

Request Method: GET

Request URL: http://127.0.0.1:8000/

Django Version: 1.2 beta 1 SVN-13009

Exception Type: ValueError

Exception Value: The view foo.views.index didn't return an HttpResponse object.

Exception Location: /home/brute/.virtualenvs/django-tests/lib/python2.6/site-packages/django/core/handlers/base.py in get_response, line 117

Python Executable: /home/brute/.virtualenvs/django-tests/bin/python

Python Version: 2.6.5

Python Path: ['/home/brute/blah', '/home/brute/.virtualenvs/django-tests/lib/python2.6/site-packages/setuptools-0.6c11-py2.6.egg', '/home/brute/.virtualenvs/django-tests/lib/python2.6/site-packages/pip-0.6.3-py2.6.egg', '/home/brute/blah', '/usr/lib/openoffice/basis-link/program', '/home/brute/.virtualenvs/django-tests/lib/python26.zip', '/home/brute/.virtualenvs/django-tests/lib/python2.6', '/home/brute/.virtualenvs/django-tests/lib/python2.6/plat-linux2', '/home/brute/.virtualenvs/django-tests/lib/python2.6/lib-tk', '/home/brute/.virtualenvs/django-tests/lib/python2.6/lib-old', '/home/brute/.virtualenvs/django-tests/lib/python2.6/lib-dynload', '/usr/lib/python2.6', '/usr/lib/python2.6/plat-linux2', '/usr/lib/python2.6/lib-tk', '/home/brute/.virtualenvs/django-tests/lib/python2.6/site-packages', '/usr/lib/python2.6/site-packages/Bitten-0.7dev_r776-py2.6.egg', '/usr/lib/python2.6/site-packages', '/usr/lib/python2.6/site-packages/PIL', '/usr/lib/python2.6/site-packages/gst-0.10', '/usr/lib/python2.6/site-packages/gtk-2.0', '/usr/lib/python2.6/site-packages/wx-2.8-gtk2-unicode']

Server time: Fri, 23 Apr 2010 04:36:03 -0500

Traceback [Switch to copy-and-paste view](#)

/home/brute/.virtualenvs/django-tests/lib/python2.6/site-packages/django/core/handlers/base.py in get_response

```
117.         raise ValueError("The view %s.%s didn't return an HttpResponse object." % (callback.__module__, view_name))
```

► Local vars

Request information

GET No GET data

POST No POST data

FILES No FILES data

Django

- Tester une application web
 - Serveur
 - Client
 - Requêtes
 - Réponses

⇒ Simulation

Exécuter les test

- Le projet par défaut en a déjà

```
python manage.py test app1 app2
Creating test database
.....
-----
Ran 154 tests in 2.679s

OK
Destroying test database
```

django 1.1 + python 2.6

MS-DOS Prompt

```
An exception 0E has occurred at 0028:C028B0B6 in UxD UCOND
00001546. This was called from 0028:C028ABE1 in UxD UCOND
00001071. It may be possible to continue normally.
```

- * Press any key to attempt to continue.
- * Press CTRL+ALT+DEL to restart your computer. You will lose any unsaved information in all applications.

Press any key to continue _

L'environnement de test

- Détection des tests à exécuter
- Création d'une base de données
- Installation des applications (syncdb)
- Exécutions des tests – affichage des résultats
- Destruction de la base

Écrire des tests



unittest

doctest

Unittest

- xUnit : jUnit / PHPUnit
- Classes python

```
from unittest import TestCase

class SomeTest(TestCase):
    # ...
```



```
from django.test import TestCase

class SomeTest(TestCase):
    # ...
```


Doctest

- Shell python dans les doctstrings

```
def add(x, y):  
    """ Retourne la somme de x et y  
  
    >>> add(1, 2)  
    3  
    >>> add('a', 'b')  
    'ab'  
    """"  
  
    return x + y
```

Mais...

- Maintenance
- Robustesse
- Souplesse

Utilisez unittest *

* à moins d'avoir
une bonne raison

Écrire des test



Où écrire ses test

- startapp crée un fichier tests.py

```
python manage.py startapp foo
```

```
foo/  
  __init__.py  
  models.py  
  tests.py  
  views.py
```

Quoi écrire

```
from django.test import TestCase

class FooTest(TestCase):
    def test_foo(self):
        # ...
```

Le client

- `django.test.Client`
 - Instanciable
 - Accessible via `self.client`
- Navigateur
 - Stateful
 - `client.cookies`
 - `client.session`

Login / logout

```
c = Client()

c.login(**credentials)

c.login(username='testuser',
        password='testpass')

c.logout()
```


GET

```
c = Client()
r = c.get('/foo/')
r = c.get('/foo/', data={'bar': 1})
r = c.get('/foo/?p=2')
```

La réponse

```
response.status_code
```

```
response.content
```

```
response.cookies
```

```
response['Content-Type']
```

La réponse (bonus)

```
response.template
```

```
response.context
```

```
response.context['user']
```

POST

```
c.post('/bar/')
```

```
c.post('/bar/?baz=12')
```

```
c.post('/bar/', data={'baz': 12})
```

```
c.post('/bar/?foo=4', data={'baz': 12})
```

HTTP, le reste

```
c.head('/foo/')
```

```
c.put('/foo/')
```

```
c.delete('/foo/')
```

```
c.options('/foo/')
```

Redirections

```
r = self.client.get('/foo')  
r = self.client.get('/foo', follow=True)  
self.assertEqual(r.status_code, 200)  
r.redirect_chain
```

django.test.TestCase

- Test complètement indépendants
 - Transaction
 - setUp()
 - tearDown()

```
class FooTest(TestCase):
    def setUp(self):
        self.user = User.objects.create_user('testuser',
            'test@email.fr', 'pass')

        self.client.login(username='testuser',
            password='pass')
```

Assertions

```
self.assertContains()
```

```
self.assertNotContains()
```

```
self.assertFormError()
```

```
self.assertTemplateUsed()
```

```
self.assertTemplateNotUsed()
```

```
self.assertRedirects()
```


En-têtes HTTP

- user-agent
- IP
- Ajax

```
self.client.get(**extra)

r = self.client.get('/updates/',
                    HTTP_X_REQUESTED_WITH='XMLHttpRequest')
```

- REMOTE_ADDR HTTP_USER_AGENT
 HTTP_REFERER HTTP_HOST ...

Alimenter la base de données

- setUp
- Fixtures (python manage.py dumpdata)

```
class FooTest(TestCase):  
    fixtures = ['test_fixtures.json',]
```

```
foo/  
  fixtures/  
    test_fixtures.json  
  __init__.py  
  models.py  
  tests.py  
  views.py
```

Envoi d'email

- `django.core.mail.outbox`

```
from django.core import mail
from django.test import TestCase

class FooTest(TestCase):
    def test_email(self):
        mail.send_mail('Sujet', 'Message',
                       'from@example.com',
                       ['to@example.com'])

        self.assertEqual(len(mail.outbox), 1)
        self.assertEqual(mail.outbox[0].subject, 'Sujet')
```

URLconf

- Application sans URLs

ou

- URLs redéfinies par l'utilisateur

```
class FooTest(TestCase):  
    urls = 'foo.test_urls'
```

- utiliser `reverse()`

```
client.get('/') ⇒ client.get(reverse('home'))
```

Conseils

- Eviter `response.content`
- Utiliser `reverse()`
- Vues testées \approx 80 % du code
- Tester aussi les mauvaises utilisations
- Style django
 - Commit = code + tests + docs

Exécution

- `manage.py test --settings=sqlite_settings`
`pg_settings`
- `TEST_DATABASE_NAME = ':memory:'`
- Django 1.2
 - `manage.py test --failfast`
 - Control-C
 - Le test runner est une classe

Utilitaires

- django-nose / py.test
- django-test-extensions
 - manage.py runtester
 - ...
- django-test-utils
 - Couverture du code
 - manage.py testmaker
 - ...



Merci

Eric Holsher

Russel Keith-Magee

<http://www.flickr.com/photos/mhines1/522575442/>

<http://www.voidspace.org.uk/python/articles/images/testing-goat.jpg>

<http://www.flickr.com/photos/gtrachier/4164815227/>

<http://www.flickr.com/photos/waldo4/2178788631/>

http://www.flickr.com/photos/restless_mind/2318914518/

http://www.flickr.com/photos/martini_dk/345915239/