**from** **django.test** **import** TestCase

**from** **myapp.models** **import** Animal

**class** **AnimalTestCase**(TestCase):

**def** setUp(self):

Animal.objects.create(name="lion", sound="roar")

Animal.objects.create(name="cat", sound="meow")

**def** test\_animals\_can\_speak(self):

*"""Animals that can speak are correctly identified"""*

lion = Animal.objects.get(name="lion")

cat = Animal.objects.get(name="cat")

self.assertEqual(lion.speak(), 'The lion says "roar"')

self.assertEqual(cat.speak(), 'The cat says "meow"')

# Run all the tests in the animals.tests module

$ ./manage.py test animals.tests

# Run all the tests found within the 'animals' package

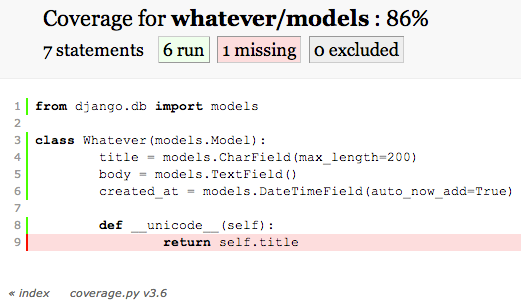
$ ./manage.py test animals

# Run just one test case

$ ./manage.py test animals.tests.AnimalTestCase

# Run just one test method

$ ./manage.py test animals.tests.AnimalTestCase.test\_animals\_can\_speak



Tests.py

from django.test import TestCase

from whatever.models import Whatever

from django.utils import timezone

from django.core.urlresolvers import reverse

from whatever.forms import WhateverForm

# models test

class WhateverTest(TestCase):

def create\_whatever(self, title="only a test", body="yes, this is only a test"):

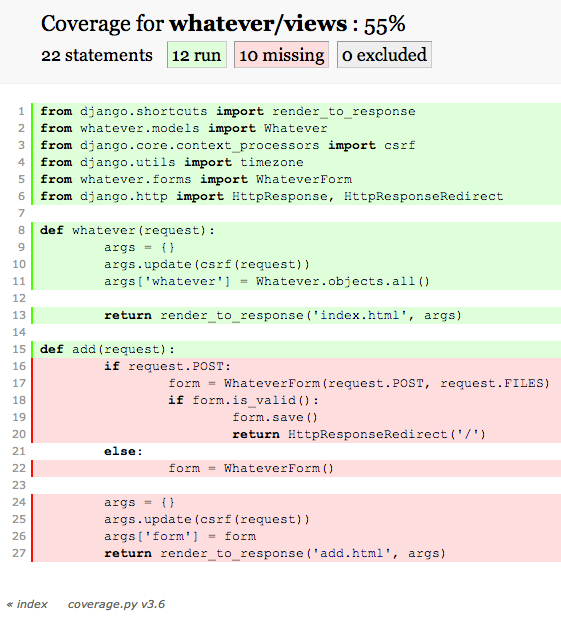
return Whatever.objects.create(title=title, body=body, created\_at=timezone.now())

def test\_whatever\_creation(self):

w = self.create\_whatever()

self.assertTrue(isinstance(w, Whatever))

self.assertEqual(w.\_\_unicode\_\_(), w.title)



def test\_whatever\_list\_view(self):

w = self.create\_whatever()

url = reverse("whatever.views.whatever")

resp = self.client.get(url)

self.assertEqual(resp.status\_code, 200)

self.assertIn(w.title, resp.content)

import unittest

from selenium import webdriver

class TestSignup(unittest.TestCase):

def setUp(self):

self.driver = webdriver.Firefox()

def test\_signup\_fire(self):

self.driver.get("http://localhost:8000/add/")

self.driver.find\_element\_by\_id('id\_title').send\_keys("test title")

self.driver.find\_element\_by\_id('id\_body').send\_keys("test body")

self.driver.find\_element\_by\_id('submit').click()

self.assertIn("http://localhost:8000/", self.driver.current\_url)

def tearDown(self):

self.driver.quit

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

def test\_valid\_form(self):

w = Whatever.objects.create(title='Foo', body='Bar')

data = {'title': w.title, 'body': w.body,}

form = WhateverForm(data=data)

self.assertTrue(form.is\_valid())

def test\_invalid\_form(self):

w = Whatever.objects.create(title='Foo', body='')

data = {'title': w.title, 'body': w.body,}

form = WhateverForm(data=data)

self.assertFalse(form.is\_valid())