Django Quickstart

kite systems

Works for

That’s me

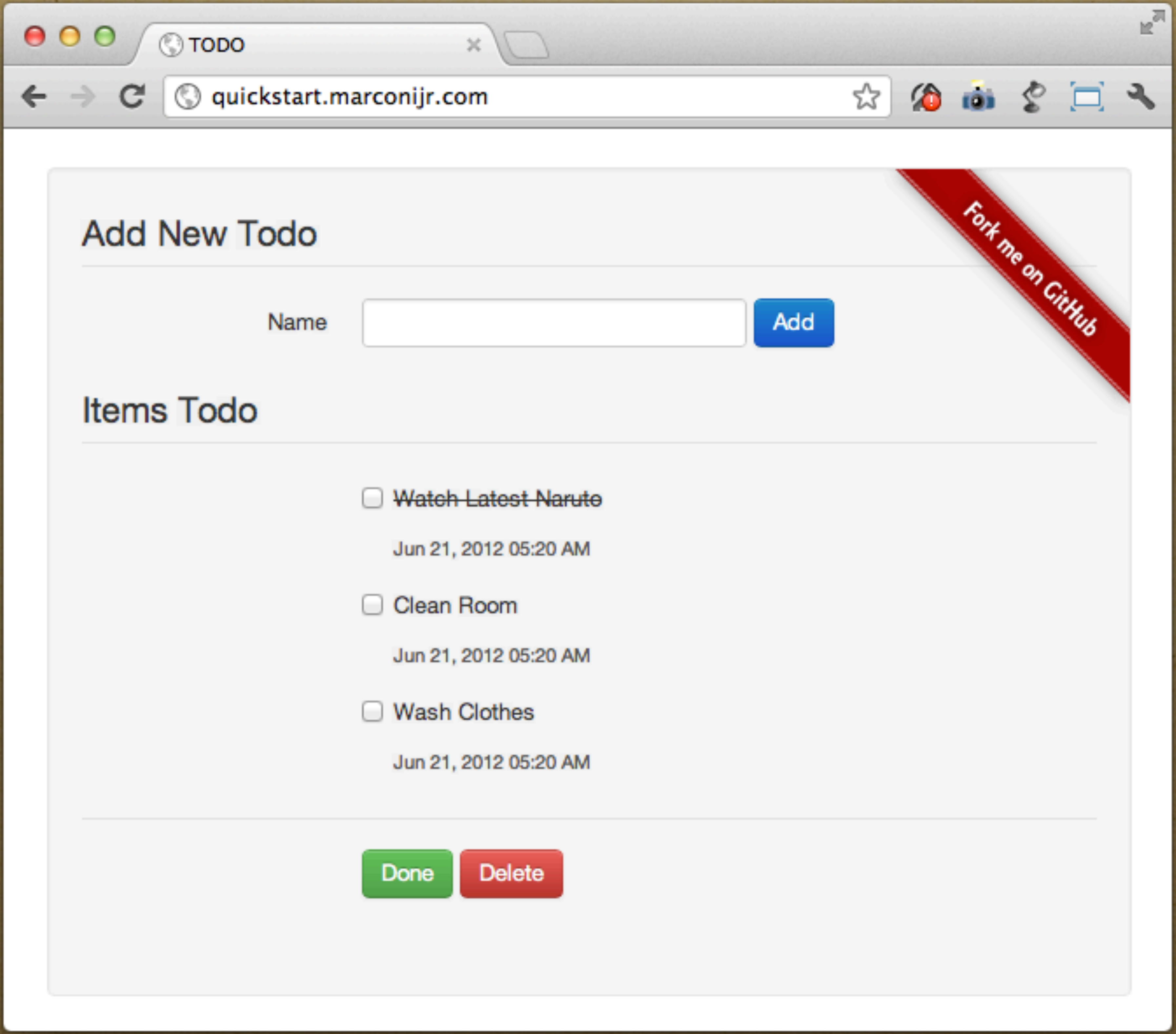
Alexander Pierce

https://www.facebook.com/groups/pinoypug

What is Django?

django
The web framework for
ponies with magical powers.

Demo: http://quickstart.marconijr.com/



Add New Todo

Name

Items Todo

□ ~~Watch Latest Naruto~~

Jun 21, 2012 05:20 AM

□ Clean Room

Jun 21, 2012 05:20 AM

□ Wash Clothes

Jun 21, 2012 05:20 AM

Setting-up development environment

- virtualenv + virtualenvwrapper
- Django 1.4

Installing virtualenv and virtualenvwrapper

```
$ pip install virtualenv virtualenvwrapper
```

then edit your .profile on osx or .bashrc
on linux and add the following lines:

```
export WORKON_HOME=~/Envs
source /usr/local/bin/virtualenvwrapper.sh
```

reload your .profile or .bashrc files:

```
$ . ~/.profile
```

Creating your virtual environment

```
$ mkvirtualenv pyconph
```

activating a virtual environment:

```
$ workon pyconph
```

for windows users:

- https://github.com/davidmarble/virtualenvwrapper-win
- https://bitbucket.org/guillermooo/virtualenvwrapper-powershell

Installing Django

```
$ pip install Django
```

testing your Django installation:

```
$ python
>>> import django
>>> django.VERSION
(1, 4, 0, 'final', 0)
```

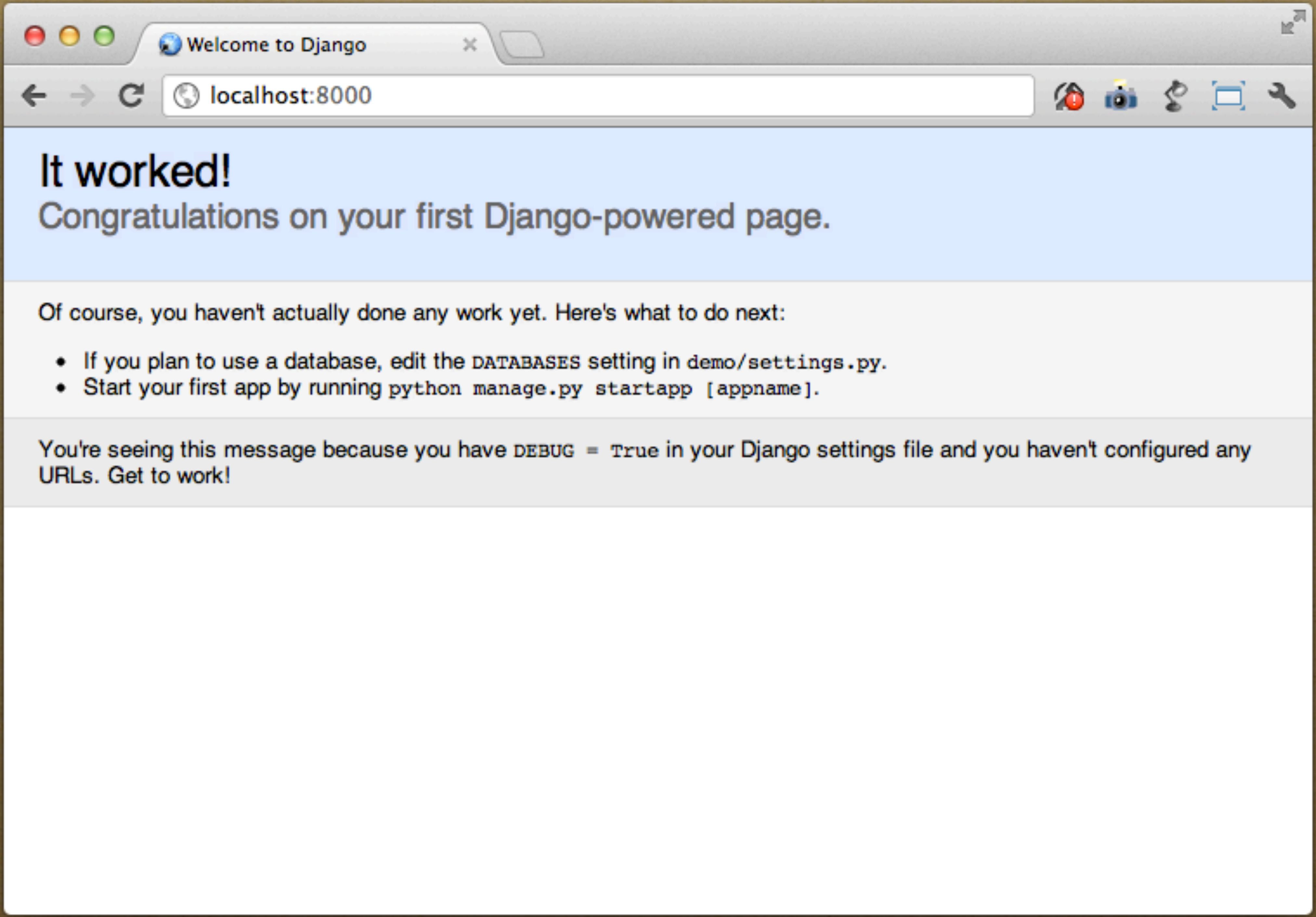
Creating a Django project

```
$ django-admin.py startproject quickstart
```

Starting Django's development server:

```
$ cd quickstart
$ python manage.py runserver
...
Development server is running at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

http://localhost:8000



It worked!

Congratulations on your first Django-powered page.

Of course, you haven't actually done any work yet. Here's what to do next:

- If you plan to use a database, edit the `DATABASES` setting in `demo/settings.py`.
- Start your first app by running `python manage.py startapp [appname]`.

You're seeing this message because you have `DEBUG = True` in your Django settings file and you haven't configured any URLs. Get to work!

Configuring your Django project

Django project directory structure:

```
quickstart
├── manage.py
└── quickstart
    ├── __init__.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py
```

Setting-up database

quickstart/settings.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': 'quickstart.sqlite',
        'USER': '',
        'PASSWORD': '',
        'HOST': '',
        'PORT': '',
    }
}
```

Setting-up directories

quickstart/settings.py

```python
import os

PROJECT_ROOT = os.path.split(os.path.abspath(
    os.path.dirname(__file__)))[0]

STATIC_ROOT = os.path.join(PROJECT_ROOT, 'static')
STATIC_URL = '/static/'

STATICFILES_DIRS = (
    os.path.join(PROJECT_ROOT, 'assets'),
)
...
TEMPLATE_DIRS = (
    os.path.join(PROJECT_ROOT, 'templates'),
)
```

Creating directories

```
$ mkdir assets static templates
```

```
quickstart
├── assets
├── manage.py
├── quickstart
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   ├── wsgi.py
├── static
└── templates
```

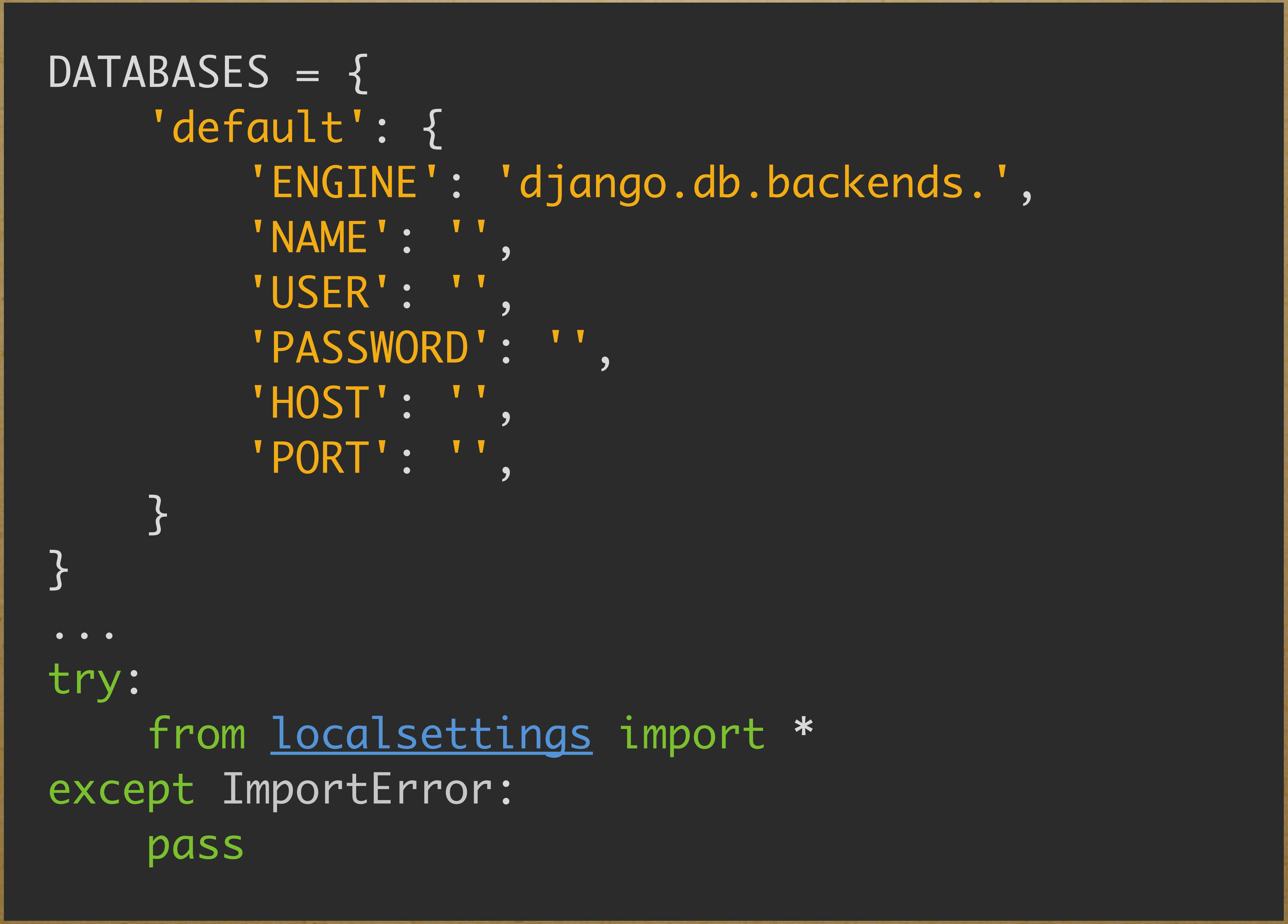
Creating local settings

```
$ vim quickstart/localsettings.py
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': 'quickstart.sqlite',
        'USER': '',
        'PASSWORD': '',
        'HOST': '',
        'PORT': '',
    }
}
```

Creating local settings

quickstart/settings.py



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.',
        'NAME': '',
        'USER': '',
        'PASSWORD': '',
        'HOST': '',
        'PORT': '',
    }
}
...
try:
    from localsettings import *
except ImportError:
    pass
```

Syncing the database

```
$ python manage.py syncdb
Creating tables ...
...
Would you like to create one now? (yes/no): yes
Username (leave blank to use 'marconi'):
E-mail address: wombat@gmail.com
Password:
Password (again):
Superuser created successfully.
Installing custom SQL ...
Installing indexes ...
Installed 0 object(s) from 0 fixture(s)
```

Enabling the Django Admin

quickstart/settings.py

```
INSTALLED_APPS = (
    ...
    'django.contrib.admin',
)
```

Enabling the Django Admin

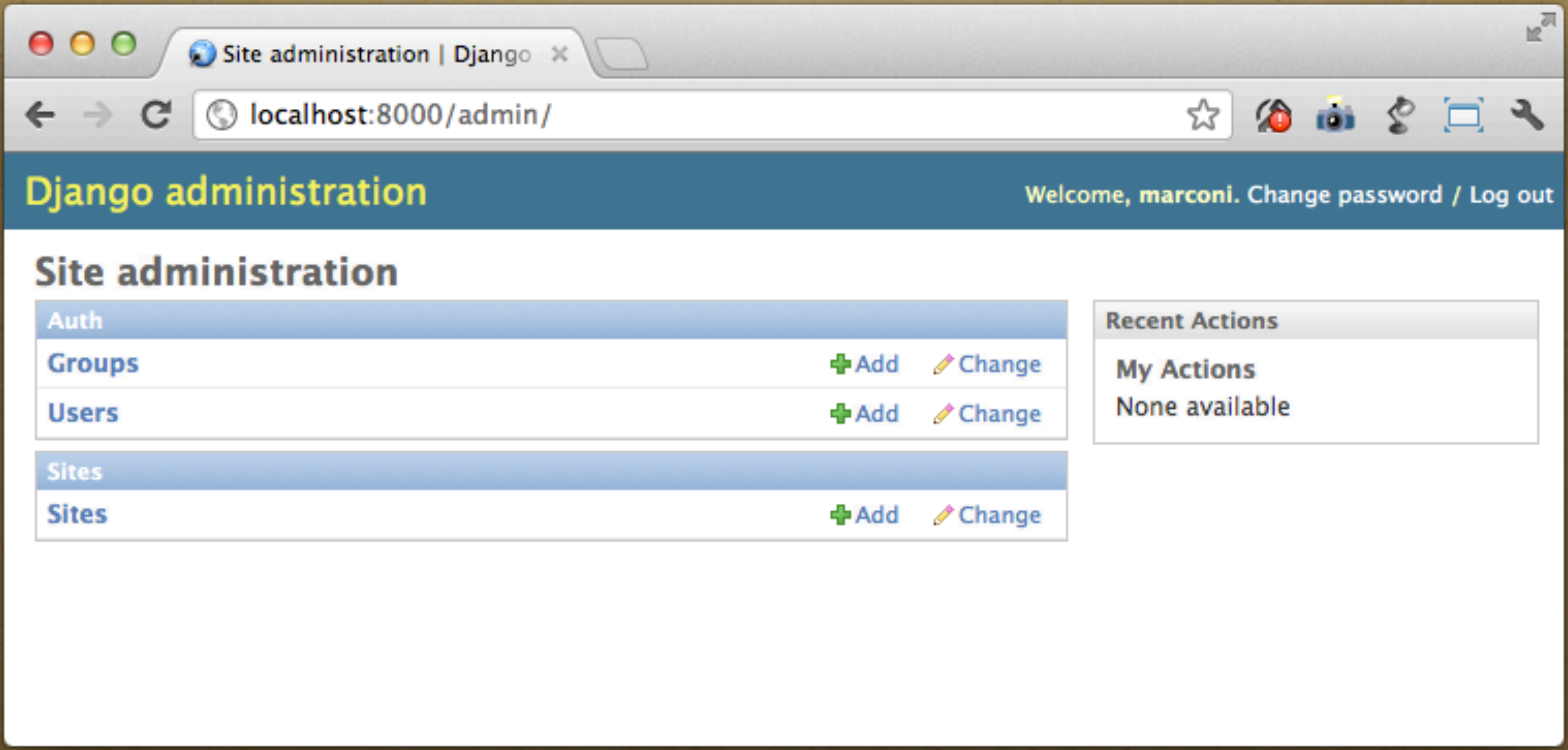
quickstart/urls.py

```
from django.conf.urls import patterns, include, url

from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    url(r'^admin/', include(admin.site.urls)),
)
```

Enabling the Django Admin

```
$ python manage.py syncdb
```

visit http://127.0.0.1/admin/



Site administration | Django
localhost:8000/admin/
Django administration
Welcome, marconi. Change password / Log out

Site administration

Auth		
Groups	Add	Change
Users	Add	Change

Sites		
Sites	Add	Change

Recent Actions

My Actions

None available

Creating a Django App

```
$ django-admin.py startapp todo
```

default app structure:

```
todo
├── __init__.py
├── models.py
├── tests.py
└── views.py
```

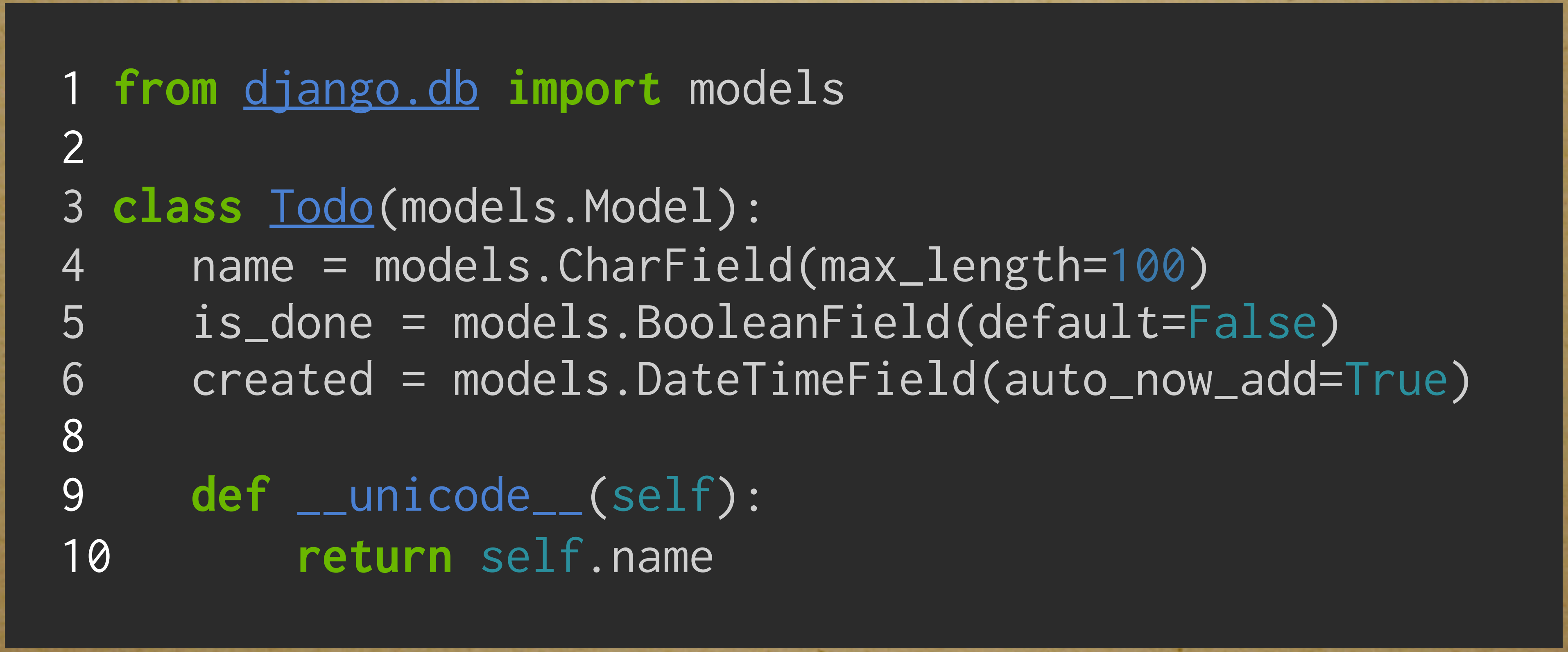
Installing your app

quickstart/settings.py

```
INSTALLED_APPS = (
    ...
    'todo',
)
```

Creating models

todo/models.py



```python
from django.db import models

class Todo(models.Model):
    name = models.CharField(max_length=100)
    is_done = models.BooleanField(default=False)
    created = models.DateTimeField(auto_now_add=True)

    def __unicode__(self):
        return self.name
```

Syncing

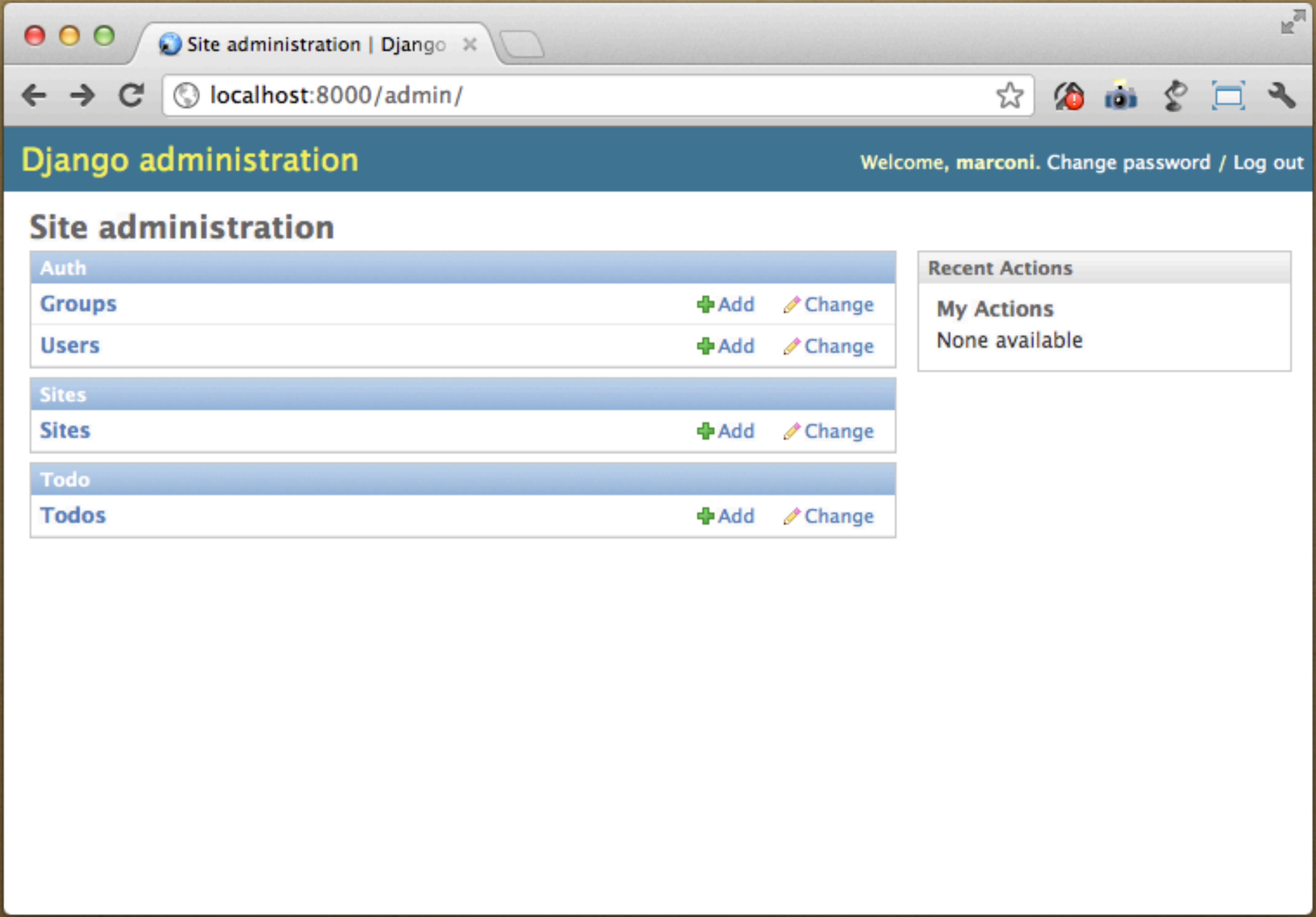
```
$ python manage.py syncdb
```

Exposing models to Django Admin

```
$ vim todo/admin.py
```

```
from django.contrib import admin
from todo.models import Todo

class TodoAdmin(admin.ModelAdmin):
    pass

admin.site.register(Todo, TodoAdmin)
```

Exposing models to Django Admin



Site administration | Django

localhost:8000/admin/

Django administration

Welcome, **marconi**. Change password / Log out

Site administration

Auth		
Groups	Add	Change
Users	Add	Change

Sites		
Sites	Add	Change

Todo		
Todos	Add	Change

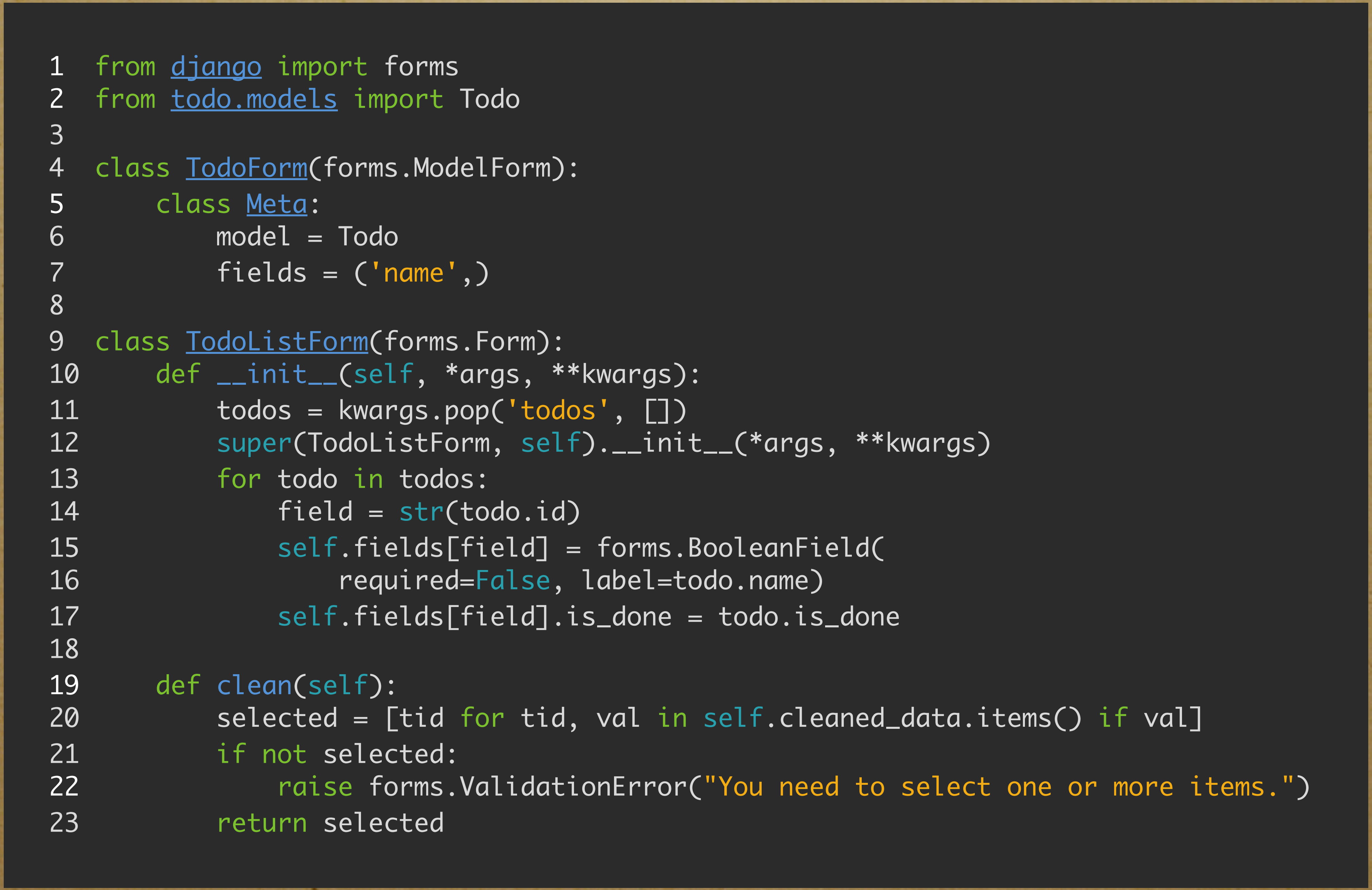
Recent Actions

My Actions

None available

Creating forms

```
$ vim todo/forms.py
```



```python
from django import forms
from todo.models import Todo

class TodoForm(forms.ModelForm):
    class Meta:
        model = Todo
        fields = ('name',)

class TodoListForm(forms.Form):
    def __init__(self, *args, **kwargs):
        todos = kwargs.pop('todos', [])
        super(TodoListForm, self).__init__(*args, **kwargs)
        for todo in todos:
            field = str(todo.id)
            self.fields[field] = forms.BooleanField(
                required=False, label=todo.name)
            self.fields[field].is_done = todo.is_done

    def clean(self):
        selected = [tid for tid, val in self.cleaned_data.items() if val]
        if not selected:
            raise forms.ValidationError("You need to select one or more items.")
        return selected
```

Creating views

todo/views.py

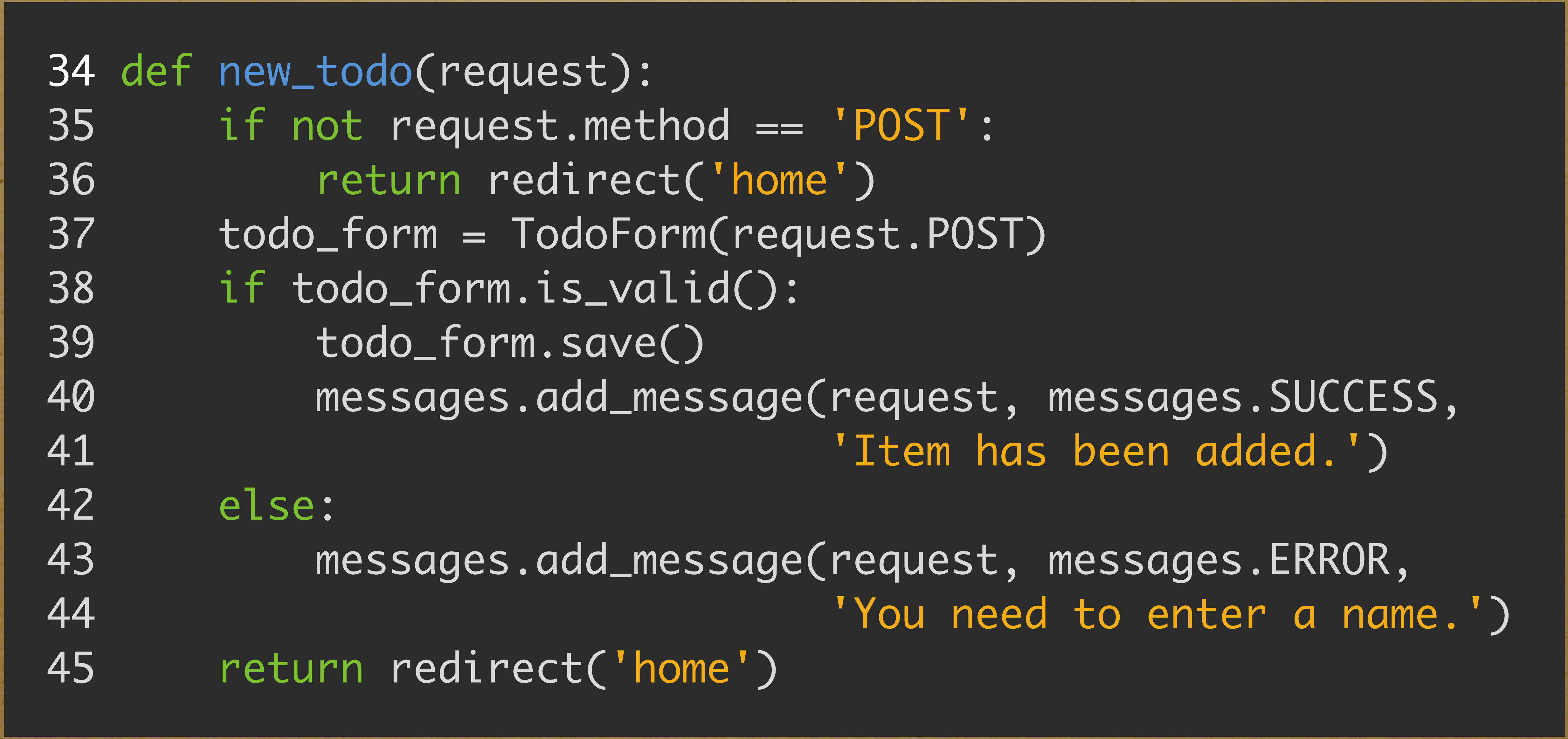
```
from django.shortcuts import render_to_response, redirect
from django.template import RequestContext
from django.contrib import messages

from todo.forms import TodoListForm, TodoForm
from todo.models import Todo

...
```

todo/views.py

```python
def home(request):
    todos = Todo.objects.order_by('-created')
    if request.method == 'POST':
        action = request.POST['action'].lower()
        todo_list_form = TodoListForm(data=request.POST, todos=todos)
        if todo_list_form.is_valid():
            selected = Todo.objects.filter(
                id__in=todo_list_form.cleaned_data)
            actions = {'done': lambda items: items.update(is_done=True),
                       'delete': lambda items: items.delete()}
            actions.get(action)(selected)
            messages.add_message(request, messages.SUCCESS,
                                 'Items has been updated.')
        else:
            messages.add_message(request, messages.ERROR,
                                 ''.join(todo_list_form.non_field_errors()))
        return redirect('home')
    else:
        todo_form = TodoForm()
        todo_list_form = TodoListForm(todos=todos)
    context = {'todo_list_form': todo_list_form, 'todo_form': todo_form}
    return render_to_response('todo/home.html', context,
                              RequestContext(request))
```

Creating views

todo/views.py



```python
def new_todo(request):
    if not request.method == 'POST':
        return redirect('home')
    todo_form = TodoForm(request.POST)
    if todo_form.is_valid():
        todo_form.save()
        messages.add_message(request, messages.SUCCESS,
                             'Item has been added.')
    else:
        messages.add_message(request, messages.ERROR,
                             'You need to enter a name.')
    return redirect('home')
```

Creating templates

```
$ mkdir templates/todo
$ touch templates/base.html
$ touch templates/todo/home.html
```

templates/base.html

```
<html>
<head>
   <title>TODO</title>
   <style type="text/css">
       fieldset {
            width: 300px;
       }
   </style>
</head>
<body>
{% if messages %}
<section id="messages">
    <ul>
        {% for message in messages %}
            <li>
                {{ message }}
            </li>
        {% endfor %}
    </ul>
</section>
{% endif %}
{% block content %}
{% endblock content %}
</body>
</html>
```

templates/todo/home.html

```
{% extends "base.html" %}

{% block content %}
<form method="POST" action="{% url new_todo %}">
   {% csrf_token %}
   <fieldset>
       <legend>Add new todo</legend>
       <div class="field">
           {{ todo_form.name.label_tag }}
           {{ todo_form.name }}
           {% if todo_form.name.errors %}
               {{ todo_form.name.errors }}
           {% endif %}
       </div>
       <input type="submit" name="action" value="Add">
   </fieldset>
</form>

...
```

templates/todo/home.html

```
...

<form method="POST">
    {% csrf_token %}
    <fieldset>
        <legend>Items todo</legend>
        {% for field in todo_list_form %}
            <div class="field">
                {{ field }}
                <label for="{{ field.auto_id }}"
                    {% if field.field.is_done %}
                        style="text-decoration:line-through;"
                    {% endif %}>
                    {{ field.label }}
                </label>
            </div>
        {% empty %}
            <p>You don't have anymore items todo.</p>
        {% endfor %}
    </fieldset>
    <br />
    <input type="submit" name="action" value="Done">
    <input type="submit" name="action" value="Delete">
</form>
{% endblock content %}
```

Adding urls:

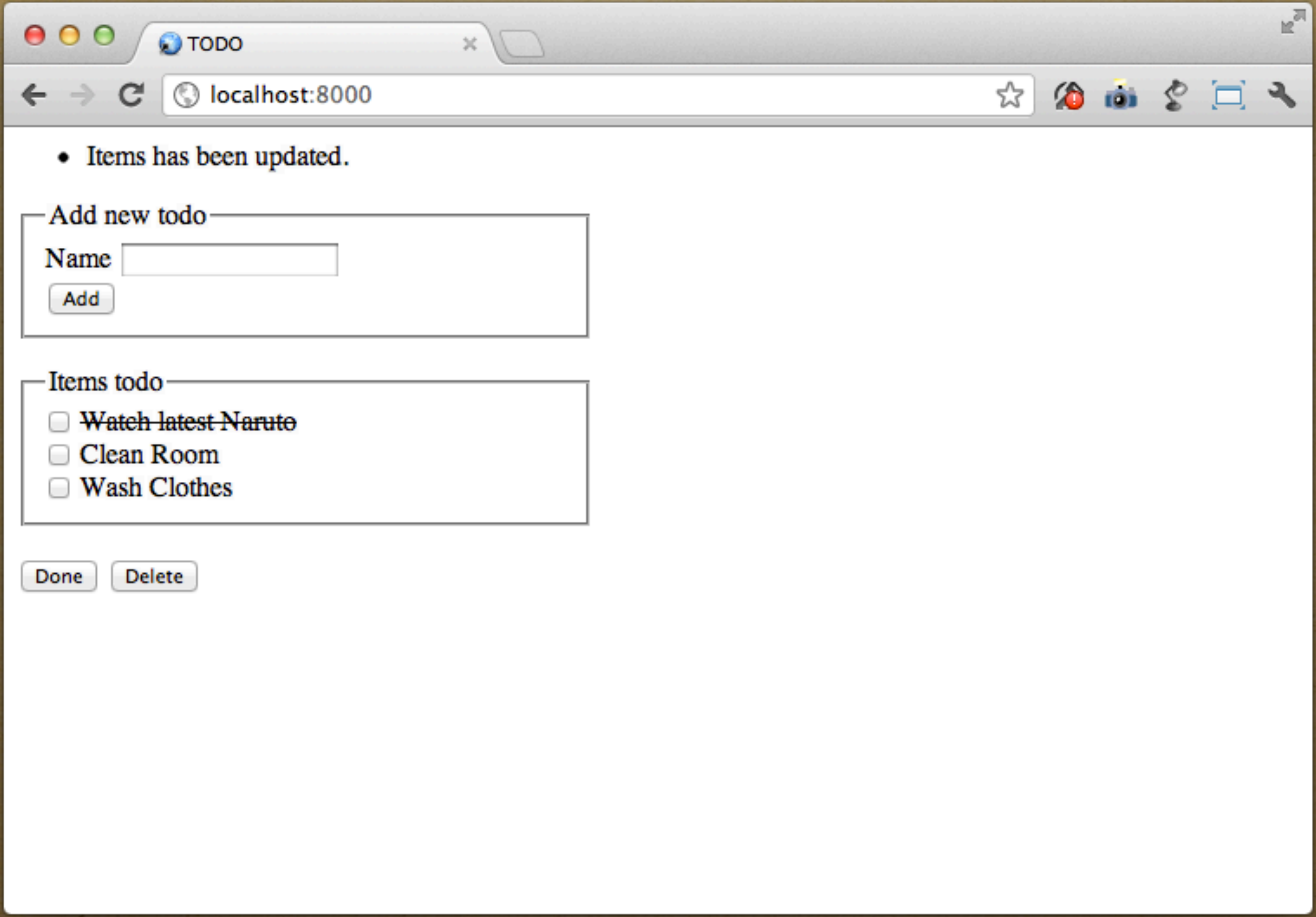
quickstart/urls.py

```
urlpatterns = patterns('',
    ...
    url(r'^$', 'todo.views.home', name='home'),
    url(r'^todo/', include('todo.urls')),
)
```

```
$ vim todo/urls.py
```

```
from django.conf.urls import patterns, url

urlpatterns = patterns('todo.views',
    url(r'^new/$', 'new_todo', name='new_todo'),
)
```

Our TODO site:



- Items has been updated.

Add new todo

Name

Items todo

□ ~~Watch latest Naruto~~
□ Clean Room
□ Wash Clothes

Thank you :)