

django

January 24th, 2007

Nicolas Seriot



Django

- Python Web Framework
- compares to Ruby On Rails
- since 2003, open-sourced in July, 2005
- origin: Kansas local newspaper
- today: Washington Post
- I personally use it since August, 2006

Python Language

- elegant, easy, interactive, interpreted, object-oriented, high level, open-source
- installed in every Mac OS X copy
- use Cocoa frameworks with PyObjC
- high level data structures

tuple	<code>t = (1, 2, 3)</code>	NSArray
list	<code>l = [1, 2, 3]</code>	NSMutableArray
dictionary	<code>d = {'ch': 'Swiss'}</code>	NSMutableDictionary

Django Install

- stable trunk

```
$ svn co http://code.djangoproject.com/svn/django/trunk/ django_src
```

```
$ ln -s django_src/django /Library/Python/2.5/site-packages/django
```

- db adaptor (PostgreSQL, MySQL, SQLite)
- apache mod_python for deployment

Start New Project

```
$ django-admin.py startproject myproject
$ cd myproject/
$ mate settings.py # edit db, create it if needed
$ python manage.py startapp myapp
$ mate myapp/models.py # define model
$ mate settings.py
    INSTALLED_APPS = (... ,
        'myproject.myapp',
        'django.contrib.admin'
    )
$ mate urls.py # uncomment admin
$ python manage.py syncdb
$ python manage.py runserver
Development server is running at http://127.0.0.1:8000/
```

Model

- `models.py`: object-relational mapper

- SQL code generation

```
$ python manage.py sql myapp
```

- admin site

```
http://127.0.0.1:8000/admin
```

- API

```
$ python manage.py shell
```

Model API

```
>>> from myproject.myapp.models import *
>>> Reporter.objects.all()
[<Reporter: Nicolas Seriot>]
>>> r = Reporter()
>>> r.full_name = "Rouletabille"
>>> r.save()
>>> r.id
2L
>>> x = Reporter.objects.filter(full_name__startswith="Rou")
>>> x
[<Reporter: Rouletabille>]
>>> x.delete()
>>> Reporter.objects.count()
1L
```

Views

- responsible to return a `HttpResponse` object with the requested page content
- typically fills a dictionary and associate it with an html file (demo ...)
- practically everything can be done with generic views
 - list / detail, create / update / delete
 - pagination

Templates

- template inheritance
- easy to change for graphic designers
- can use built-in or custom filters

```
<head>  
<title>{% block title %}hello{% endblock %}</title>  
</head>
```

```
{% extends "base.html" %}  
{% block title %}Archives {{ year }}{% endblock %}  
{{ article.article|textile }}
```

From request to html

<http://127.0.0.1:8000/myproject/articles/2007/>

- **urls.py**

```
(r'^articles/(\d{4})/$', 'myproject.myapp.views.year_archive')
```

- **views.py**

```
def year_archive(request, year):  
    articles = Article.objects.filter(pub_date__year=year)  
    d = {}  
    d['year'] = year  
    d['articles'] = articles  
    return render_to_response('year_archive.html', d)
```

- **year_archive.html**

```
{% for a in articles %}  
    <p>{{ a.headline }}</p>  
{% endfor %}
```

Problems Django solved for me

- forms validation (seninvoice)
- user authentication (seninvoice)
- email sending (seninvoice, smardtv)
- syndication framework (newsmemory)
- internationalization (smardtv)
- unit tests (seninvoice)

```
$ python manage test
```

Internationalization

- edit template files

```
{% trans "my_text" %}
```

- generate translation files

```
$ make-messages.py -a
```

- edit and compile locale strings

```
$ mate locale/fr/LC_MESSAGES/django.po
```

```
$ compile-messages.py
```

There is much more

- easy GET / POST / session access

```
def a_view(request):  
    # request.GET dict  
    # request.POST dict  
    # request.session dict
```

- powerful caching framework
 - chicagocrime.org survived slashdotting!
 - grono.net: 20'000 on 400'000 users, now one server instead of four for Java Tomcat

References

- Django Project
<http://www.djangoproject.com/>
- Django Book
<http://www.djangobook.com/>
- Django Users Mailing List
<http://groups.google.com/group/django-users>

Thank you