Building Websites With Django

Malcolm Tredinnick

linux.conf.au
17 Jan 2007

(Slides will be available from
http://www.pointy-stick.com/lca2007/)

Photo Credits

- JD Lasica (jdlasica on Flickr) for photo of Adrian Holovaty (used under Creative Commons by-sa-nc license)
- Jonas Luster (jluster on Flickr) for photo of Jacob Kaplan-Moss (used under Creative Commons by-sa-nc license)
- Natalie Downe (NatBat) for photo of Simon Willison (used under Creative Commons by-sa-nc license)
- Jacob Kaplan-Moss for image of Django sites (used with permission).

What?

Who? When?

I WISH I WERE
POWERED BY
django

Where?

- Originally written for http://ljworld.com/ and related sites.
- Now, everywhere...
  - Blogs
  - Media sites
    - http://projects.washingtonpost.com/congress/
  - Game sites
  - Crime sites (!)
    - http://www.chicagocrime.org/

For A Good Time, Call....

http://code.djangoproject.com/wiki/DjangoPoweredSites

How Cool Is This?

“...In fact Django reminds me a bit of the character in Airplane who always answers the 'what do you make of that?' question literally... Why, I can make a hat or a brooch or a pterodactyl...”

Scott Gilbertson, wired.com

The Gory Details...

http://www.djangoproject.com/

http://www.djangobook.com/

Videos

- Search for “Django” on http://video.google.com/
  - One talk given at Google by Jacob Kaplan-Moss.
  - “Snakes and Rubies” talk: the Snakes side was given by Adrian Holovaty (in Chicago).

How Does It Work?

Life Of A Web Page Request

- HTTP request sent to server.
- Who can process a request to this URL?
- Pass control to processing function (view)
- Collect data for result
  - Some “mostly static” data (the template)
  - Some dynamic bits
- Put template and dynamic bits together
- Send HTTP response back to user

Bonus Features

- Retrieving answer from cache
- Translating result
- Gzipping returned data
- Session management
- Authentication management
- Admin interface

Interactive Demo

Starting a new project...

Models: Players

```
from django.db import models

class Player(models.Model):
    name = models.CharField(maxlength=100)
    city = models.CharField(maxlength=100)

    class Admin:
        list_display = ('name', 'city')

    def __str__(self):
        return '%s' % self.name
```

Models: Matches

```
class Match(models.Model):
    home_team = models.ForeignKey(Player, related_name='home_matches')
    away_team = models.ForeignKey(Player, related_name='away_matches')
    home_score = models.IntegerField()
    away_score = models.IntegerField()

    class Meta:
        verbose_name_plural = 'Matches'

    class Admin:
        list_display = ('match_name', 'score')

    def __str__(self):
        return '%s vs. %s' % (self.home_team, self.away_team)

    def match_name(self):
        return str(self)

    def score(self):
        return '%s - %s' % (self.home_score, self.away_score)
```

Interactive Demo

We can access the data from the Python prompt...

Other Query Uses

```
class Player(models.Model):
    ....

    def for_against(self):
        for_total = 0
        against_total = 0
        for score in self.home_matches.all():
            for_total += score.home_score
            against_total += score.away_score
        for score in self.away_matches.all():
            for_total += score.away_score
            against_total += score.home_score
        return '%s : %s' % (for_total, against_total)
```

Templates

```
<html>
   <head>
      <title>Match Results</title>
   </head>

   <body>

      <h1>Results of matches played to date</h1>
      <ul>
      {% for match in object_list %}
         <li>{{ match.home_team.name }} ({{ match.home_team.city }}) -
             {{ match.away_team.name }} ({{ match.away_team.city }}):
             {{ match.home_score }} : {{ match.away_score }}</li>
      {% endfor %}
      </ul>

   </body>
</html>
```

Populating The Template

We are going to cheat and use “generic views”...

URL Configuration

```
from django.conf.urls.defaults import *
from django.views.generic.list_detail import object_list
from lca2007.sports.models import Match

info_dict = {
    'queryset': Match.objects.all(),
    'template_name': 'matches.html',
}

urlpatterns = patterns('',
    (r'^admin/', include('django.contrib.admin.urls')),

    (r'^sports/', object_list, info_dict),
)
```

A Real-Life View

```
def topic(request, suffix = None, page = 0):
    """
    Display paginated entries for a particular tag and all the tags under that
    tag. If no suffix is supplied, show the page that allows browsing of all
    tags.
    """
    if suffix is None:
        # Present the topic summary page.
        return browse_topics(request)

    try:
        tag_list = Tag.objects.get_subtree(suffix)
    except Tag.DoesNotExist:
        return bad_or_missing(request, 'The topic you have requested ("%s") '
                'does not exist.' % suffix)

    tag_idents = [t.id for t in tag_list]
    entries = Entry.published.filter(tags__id__in = tag_idents).distinct()
    tag = tag_list[0]

    return page_display(request, entries, 'weblog/entry_list_topic.html',
            page)
```

Interactive Demo

Custom template tags (and their uses)...

Example Code

- My blog software:
  http://www.pointy-stick.com/software/website-latest.tar.gz
- The Django website itself:
  http://code.djangoproject.com/browser/djangoproject.com
- Other projects on the Django projects page mentioned earlier
- Google code repository
- Mailing list archives

Sites To Read

- The Lawrence mob are great!
  - www.b-list.org/ (James Bennett)
  - www2.jeffcroft.com/ (Jeff Croft)
  - www.jacobian.org/ (*The* Jacob Kaplan-Moss)
- Django community aggregator
  - www.djangoproject.com/community/