Lecture 7 – Django Models

Web Application Development

February 3, 2015

Jeffrey L. Eppinger
Professor of the Practice
School of Computer Science

Lecture Schedule – 1st Half

(subject to change)

#1 Intro
#2 HTML & CSS
#3 JavaScript & DOM
#4 CSS Frameworks
#5 HTTP & Django
#6 Cookies & Sessions
#7 Django Models
#8 Transactions
#9 Django Templates
#10 Images
#11 AJAX
#12 jQuery
#13 Databases
#14 Cloud Deployment
#15 SSL
#16 Project Proposals

Agenda

→ Course Administration

Django Models

Django User Authentication

Administrative Issues

- HW#2 Grades Posted
  - It's in the grades branch in your student repos for this class
- HW#3 due last night
  - Last penalty-free late day is Wednesday
  - See me if you will turn it in after Thursday
- HW#4 to be posted
  - Tonight or tomorrow morning

Agenda

- ✓ Course Administration
- → Django Models
- Django User Authentication

models.py

- Declare Python classes to represent database tables
  - Each class subclasses django.db.models.Model
  - Each attribute of a class represents a field in the table
  - Many field types, including:
    - BooleanField
    - CharField
    - DateField, DateTimeField, TimeField
    - IntegerField
    - IPAddressField
    - Etc, etc
  - By default, every table gets a serial ID field
- For more info, see: https://docs.djangoproject.com/en/1.7/ref/models

Model Manager

- Django model superclass defines a model manager
  - It's called "objects"
- Has functions to access your DB objects, including:

  <model>.objects.all()

  <model>.objects.count()

  <model>.objects.get(…)

  <model>.objects.fetch(…)

  <model>.objects.exclude(…)
- Query constraints (the … params), include:

  <field>__contains='…'

  <field>__icontains='…'

  <field>__exact='…'

  <field>__startswith='…'

  <field>__lt='…'

Model Instance Functions

- Given an instance

  .save()

  .delete()

SQLite

- Default database implementation is SQLite
  - Public domain, serverless, no config databases
- Stores your data in db.sqlite3 file

- Add your app to the INSTALLED_APP in settings.py
- Set up the model:

  python manage.py migrate
- Then run server:

  python manage.py runserver

- Change database:

  remove db.sqlite3, then migrate again

Example

- A new version of "shared" to do list
  - http://real.wv.cs.cmu.edu:8000/shared

Things to Notice

- urls.py
  - Server with many applications includes other urls.py files
  - Capture values from URLs
    - For more info see: https://docs.djangoproject.com/en/1.7/topics/http/urls/
- models.py
  - __unicode__() function allows Django to print objects
- views.py (controller actions)
  - Use of the model
  - Pass model objects in the context
- Templated views
  - Use model objects to present the information
- Use of ID when deleting to uniquely identify row

Django shell

- You can interact with your model:

  python manage.py shell

  from shared.models import *

A More Complex Sample

```
class Student(models.Model):
      andrew_id = models.CharField(max_length=8, primary_key=True)
      fname = models.CharField(max_length=8)
      lname = models.CharField(max_length=8)

      def __unicode__(self):
           return self.fname+" "+self.lname+" ("+self.andrew_id+")"

class Course(models.Model):
      name = models.CharField(max_length=200)
      number = models.CharField(max_length=6)
      students = models.ManyToManyField(Student)

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

Things to Notice

- Many-to-Many Field Constructor
- Allows
  - Queries that span the relationship
    - E.g., Course.objects.filter(students__fname__contains='e')
  - Modifications to the other model objects
    - Student s: s.course_set.all()
- Default id is not available until you save() instance

Agenda

- ✓ Course Administration
- ✓ Django Models
- → Django User Authentication

Django User Authentication

- Build-in package
- Provides
  - User model
    - Does not user default model manager
  - Login “view” action
    - You can provide the template

Example

- A new version of “private” to do list
  - http://real.wv.cs.cmu.edu:8000/private

Agenda

- ✓ Course Administration
- ✓ Django Models
- ✓ Django User Authentication

Next Class

Transactions