Web Application Frameworks

Robert M. Dondero, Ph.D.
Princeton University
Objectives

- You will learn about:
  - The Django web app framework
  - Other MVC web app frameworks (briefly)
  - Other web app frameworks (briefly)
  - Web app framework assessment
Motivation

- Conventional approach to building a website:
  - Write ad hoc client-side code
    - In HTML, CSS, JavaScript, etc.
  - Write ad hoc server-side code
    - In Python, Java, PHP, etc.
    - Write ad hoc code to access DBMS
- Well understood
- Web app frameworks mechanize (parts of) the process
Part 1:
The Django Web App Framework
Django

Jean "Django" Reinhardt
1910-1953

Jazz guitarist and composer
The Django Web App Framework

- **Who**: Adrian Holovaty & Jacob Kaplan-Moss (Benevolent Dictators for Life)
- **When**: 1995
- **Why**: Help web developers do more, use good style, and avoid security vulnerabilities
Django Characteristics

- A Python package (collection of modules)

"Django lets you build deep, dynamic, interesting sites in an extremely **short time**. Django is designed to let you focus on the **fun**, interesting parts of your job while easing the pain of the repetitive bits. In doing so, it provides high-level **abstractions** of common Web-development patterns, **shortcuts** for frequent programming tasks, and clear **conventions** on how to solve problems. At the same time, Django tries to stay out of your way, letting you work outside the scope of the framework as needed."

– Holovaty and Kaplan-Moss
Django Test Server

- Django test web server
  - Good for development
  - Not good for deployed version
    - Handles only a single request at a time reliably
    - No security audit
- To use:
  - Issue command: cd *projectDirectory*
  - Issue command: python manage.py runserver
  - In browser: http://localhost:8000/*somePage*
PennypackDjango1 App

- See PennypackDjango1 App
  - book.py, database.py, common.py
  - manage.py
  - settings.py
  - urls.py
  - views.py
- Generalizing...
PennypackDjango1 App

- Django separates URL from file name
  - Can use "pretty" URLs
  - Can map multiple URLs to same function
  - Can change function name without changing URL
  - Can change URL without changing function name
PennypackDjango2 App

- See PennypackDjango2 App
  - book.py, database.py
  - manage.py
  - settings.py
  - urls.py
  - header.html, footer.html, searchform.html, searchresults.html
  - views.py
- Generalizing...
PennypackDjango2 App

- Templates
  - Template tags, variables
  - Can be used independent of Django

- Django separates:
  - Presentation (.html templates)
  - Business logic (views.py)

- Easy to compose HTML
  - No need for redirection when author name is missing/empty!
See PennypackDjango3 App

- book.py, database.py, common.py
- settings.py
- urls.py
- header.html, footer.html, searchform.html, results.html
- views.py
- models.py

Generalizing...
Django separates:

- Presentation (.html templates)
- Business logic (views.py)
- Model (the database)

Programmer need not know SQL
PennypackDjango4 App

- PennypackDjango4 App
  - book.py, database.py, common.py
  - settings.py
  - urls.py
  - header.html, footer.html, searchform.html, results.html
  - views.py
  - models.py
  - admin.py

- Generalizing...
PennypackDjango4 App

- Try in browser:
  - http://localhost:8000/admin/

- Django automatically provides an "administrator" interface
  - Allows site administrator to manage content of application's database tables
  - "A less powerful but more convenient version of your database's command-line utility"
  - **Not** a public interface to application's data
And More!

- There is much more to Django
- See the Django book:
- Also online
Part 2: MVC Web App Frameworks
MVC

- Overall architecture...
- The **Model-View-Controller (MVC)** design pattern
  - **Model**: The system's **data access** logic
  - **View**: The system's **data presentation** logic
  - **Controller**: (vague) The system's **business** logic; the logic that connect the data with the presentation
MVC in Django

- In Django
  - **Model**: models.py
  - **View**: HTML templates
  - **Controller**: view.py (confusingly)
Other MVC Frameworks

- See

- For Python
  - CherryPy, **Django**, Grok, Pyjamas, Pylins, Pyramid, TurboGears, web2py, Webward, Zope 3, Zope2

- For Java
  - Apache Click, Sling, **Apache Struts**, Apache Wicket, ItsNat, **JavaServer Faces**, JBoss Seam, Jspx-bay, Jvx WebUI, OpenXava, Play, RIFE, **Spring**, Stripes, Tapestry, WaveMaker, ztemplates, ZK
Other MVC Frameworks

- For PHP

- For Ruby
  - Camping, Nitro, *Ruby on Rails*, Sinatra
Part 3: Web App Frameworks
Google Web Toolkit

**Input:**
- Java code, similar to Swing
- Strong type checking (unlike JavaScript)
- Specifies browser GUI

**Output:**
- Equivalent JavaScript/HTML/CSS code
- One script for each popular browser!!!
  - Hides browser incompatibilities
Google App Engine (GAE)

- Web app framework (MVC)
  - Java or Python
  - Python framework uses Django template engine
- Hosting service
  - Web server, file space, persistent storage
  - Must use Google persistent data store (and not the DBMS of your choice)
Amazon Elastic Compute Cloud (EC2)

- Hosting service
  - Users rent predefined "virtual machines"
    - E.g. Ubuntu Linux, Apache Web server, MySQL, Python, and Django, running on the East coast of the U.S.
  - Users can create virtual machines
Part 4: Web App Framework Assessment
Framework Assessment

- Positives
  - Yields reliable code
    - Framework code has been thoroughly tested
  - Yields consistent code
    - Code has uniform structure across apps
  - Makes efficient use of programmer time
    - Framework handles repetitive parts
    - "DRY" (don't repeat yourself) is encouraged
Framework Assessment

- Positives
  - Allows separation of concerns:
    - Model: DB administrator (SQL, normalization, ...)
    - View: Web designer (HTML, CSS, ...)
    - Controller: Programmer (Python, Java, ...)
  - Facilitates parallel development
    - Can develop model, view, controller concurrently
  - Facilitates maintenance
    - Framework provides loose coupling
    - Can change model, view, controller independently
Framework Assessment

- **Negatives**
  - Can yield code that is difficult to understand and change
  - Can yield systems that are large
  - Can yield systems that are slow

- See Joel Spolsky's "Why I Hate Frameworks"
Summary

- We have covered:
  - The Django web app framework
  - Other MVC web app frameworks (briefly)
  - Other web app frameworks (briefly)
  - Web app framework assessment