CS320 Web and Internet Programming
MVC Architecture

Chengyu Sun
California State University, Los Angeles

Java Web Application
- Servlets
- Beans
- JSPs
  - Scripting elements, EL, JSTL
- Static resources
  - HTML, CSS, images, ...
- Metadata files
  - web.xml, ...

Model 1 Architecture
- JSP + Bean
  - JSP for presentation
  - Bean for business logic
- Example
  - GuestBook (Bean, EL, and JSTL)

Problems of Model 1 Architecture
- Using scripting elements mixes presentation and processing
- Hard to debug, maintain, or reuse code
- Not using scripting elements limits the interaction between presentation and processing to getters and setters
  - Tidious to program
  - Beans are no longer independent of the presentation layer, i.e. special getters/setters are needed

Improve Model 1 Architecture

Model 2 Architecture
- A.K.A. Model-View-Controller (MVC) Architecture
  - View
  - Controller
  - Model
  - JSP
  - Servlet
  - Bean

Data Models
- Independent of UI
- Bean (POJO)
- E.g. the GuestBookEntry class
MVC in a Web Application ...

1. Browser sends a request to controller
2. Controller processes the request, updates some data
3. Controller forwards the request and data to view
4. View generates the response that is sent back to the client

Guest Book Example Using MVC

- **Model**
  - GuestBookEntry.java
- **View**
  - GuestBook.jsp, AddComment.jsp, EditEntry.jsp
  - Redirect
- **Controller**
  - GuestBook.java, AddComment.java, EditEntry.java

Forward Request From Controller to View

```
controller

request

view

request.getRequestDispatcher("path_to_jsp")
.forward(request, response);
```

Forward vs. Redirect

```
request.getRequestDispatcher("path_to_jsp")
.forward(request, response);
```

Send Data From Controller to View

- Objects in *application* and *session scope* are shared by all servlets and JSPs of the application
- Additional data can be passed from servlet to JSP in *request scope*

```
request.setAttribute("objName", obj);
request.getRequestDispatcher("path_to_jsp")
.forward(request, response);
```
More About the MVC Example

- One operation, one controller
- Requests always go to controllers first
  - "Hide" JSPs under /WEB-INF/
- Controllers do not generate HTML
  - No `out.println()`
- JSPs are only used for display
  - No scripting elements in JSP