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
  - Tedious to program
  - Beans are no longer independent of the presentation layer, i.e. special getters/setters are needed
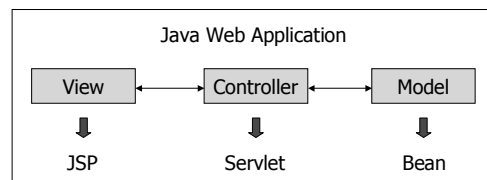

Improve Model 1 Architecture

Application

Presentation ↔ ?? ↔ Data Models
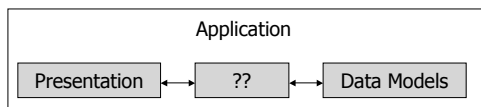
- ◆ Presentation
  - Create UI
  - Input and output
  - JSP, JFC/Swing ...
- ◆ Data Models
  - Independent of UI
  - Bean (POJO)
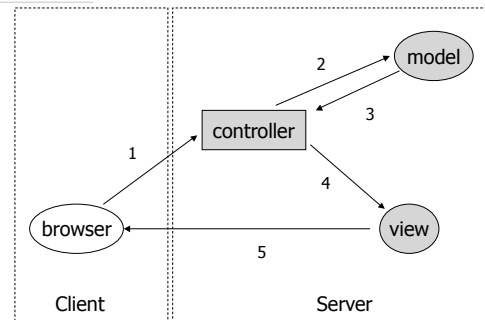 - E.g. the `GuestBookEntry` class

---

Model 2 Architecture

- ◆ A.K.A. Model-View-Controller (MVC) Architecture

Java Web Application

View ↔ Controller ↔ Model

JSP   Servlet   Bean

About MVC

◆ Originate from the work on *Smalltalk*
◆ Widely used in GUI applications

MVC in a Web Application ...

Diagram: browser (Client) → 1 → controller (Server); controller → 2 → model; model → 3 → controller; controller → 4 → view; view → 5 → browser.


... MVC in a Web Application

1. Process request
2. Create/update beans
3. Store beans in request, session, or application scope
4. Forward request to JSP page
5. Extract data from beans and display
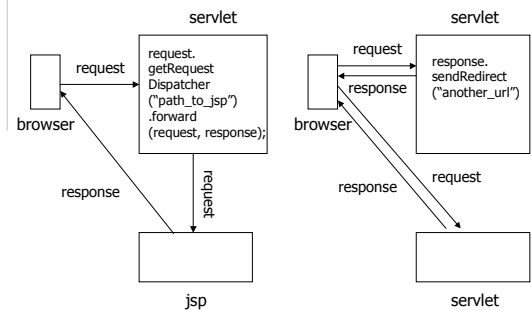

Guest Book Example Using MVC

◆ **M**odel
  - GuestBookEntry.java
◆ **V**iew
  - GuestBook.jsp, AddComment.jsp, EditEntry.jsp
  - *Redirect*
◆ **C**ontroller
  - GuestBook.java, AddComment.java, EditEntry.java

Forward Request From Controller to View

controller — data → view
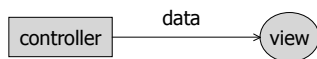
```
request.getRequestDispatcher( "path_to_jsp" )
        .forward( request, response );
```

Forward vs. Redirect

Left diagram: browser → request → servlet (request.getRequestDispatcher("path_to_jsp").forward(request, response);); servlet → request → jsp; response back to browser.

Right diagram: browser → request → servlet (response.sendRedirect("another_url")); response → browser; request → servlet.

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