CS320 Web and Internet Programming

Web Applications and MVC

Chengyu Sun
California State University, Los Angeles

Overview

- Web application
  - web.xml
- More ANT tricks
- MVC

Java Web Application

- Components
  - Servlets
  - JSPs
  - Classes
  - Static documents (HTML, images, sounds etc.)
  - Meta information
- *Everything in the same context is considered part of one application*

Directory Structure

- webapp root
  - WEB-INF/
  - WEB-INF/classes
  - WEB-INF/lib
  - WEB-INF/web.xml *(deployment descriptor)*

web.xml

```
<?xml version="1.0"?>

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
        version="2.4" >

  <description>
         A Java web application example for CS320.
  </description>
  <display-name>CS320 Web App Example</display-name>

</web-app>
```

Some <web-app> Elements

- <welcome-file-list>
- <servlet> and <servlet-mapping>
- <session-config>
- <context-param>

More About web.xml

- Java Servlet 2.4 Specification
  - SRV.13.4

Problems of Deployment

- Reloadable context
  - Degrade server performance
  - Somewhat inconsistent behavior
    - Classes and JSP are automatically recognized
    - Changes to .tld are automatically recognized
    - Changes to web.xml are not automatically recognized
    - Problems with *included* pages
- Non-reloadable context
  - Manual deployment using the manager interface is tedious

ANT to the Rescue

- Both developed originally by James Duncan Davidson
- A number of ANT tasks designed specifically for web development with Tomcat
- Usage
  - Include catalina-ant.jar in $CATALINE_HOME/server/lib
  - <taskdef> in build.xml

Create a WAR File

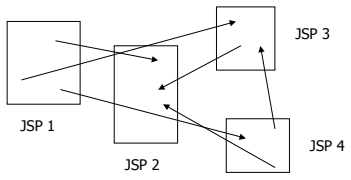
- WAR – JAR file for web applications
- <war>
  - destfile
  - webxml
  - <classes>
  - <lib>
  - <webinf>
  - <metainf>

Tomcat Manager Tasks

- <deploy>
- <undeploy>
- <start>
- <stop>
- <reload>
- and a few others
- *work both locally and remotely*

Model 1 Architecture

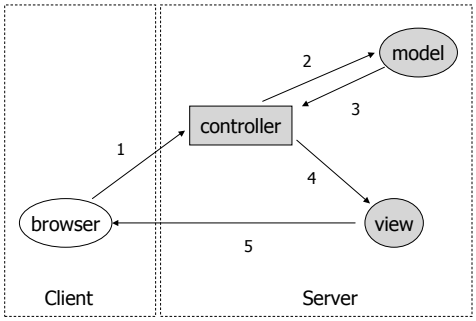
- JSPs + beans
  - JSPs for presentation
  - beans for business logic



Model 2 Architecture

- Also know as Model-View-Controler (MVC) architecture
  - JSPs + beans + servlet
  - Beans for business logic – Model
  - JSPs for presentations – View
  - servlet for web logic – Controller
    - HTTP related processing, e.g. request, response, sessions etc.
    - *Request dispatching*

MVC Control Flow ...



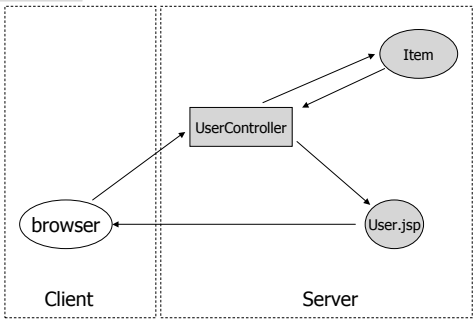
... MVC Control Flow

1. Process request
2. Populate beans
3. Store results in request, session, or servlet context
4. Forward request to JSP page
5. Extract bean data from beans and display

MVC Example

- **M**odel
  - User.java, Item.java
- **V**iew
  - Login.jsp, Admin.jsp, User.jsp
- **C**ontroller
  - LoginController.java, LogoutController.java
  - AdminController.java, UserController.java

MVC Example – User.html



MVC Frameworks

- Struts
  - http://struts.apache.org
  - Backed by SUN and Apache Foundation
  - Mature framework
  - Widely used and well supported
- Spring
  - http://www.springframework.org
  - New buzz in J2EE community
  - More flexible
  - Less dependency on the framework