CS3220 Web and Internet Programming

Introduction to JSP and MVC Architecture

Chengyu Sun

California State University, Los Angeles

Java Server Page (JSP)

- Why?
  - It's tedious to generate HTML using println()
  - Separate presentation from processing
- How?
  - *Code elements* embedded in *HTML documents*

HelloJSP.jsp

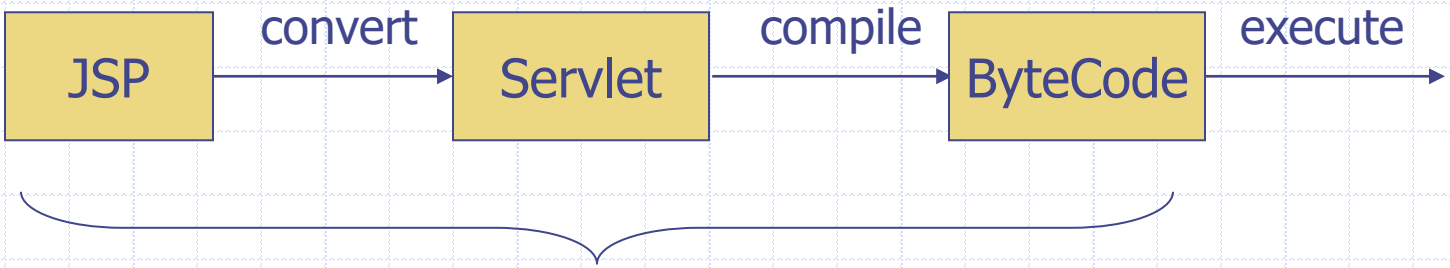
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8"><title>HelloJSP</title>
</head>
<body>
    <p>A JSP without J or S.</p>
</body>
</html>
```

Change default JSP encoding to UTF-8 in Eclipse Preferences

Access JSP

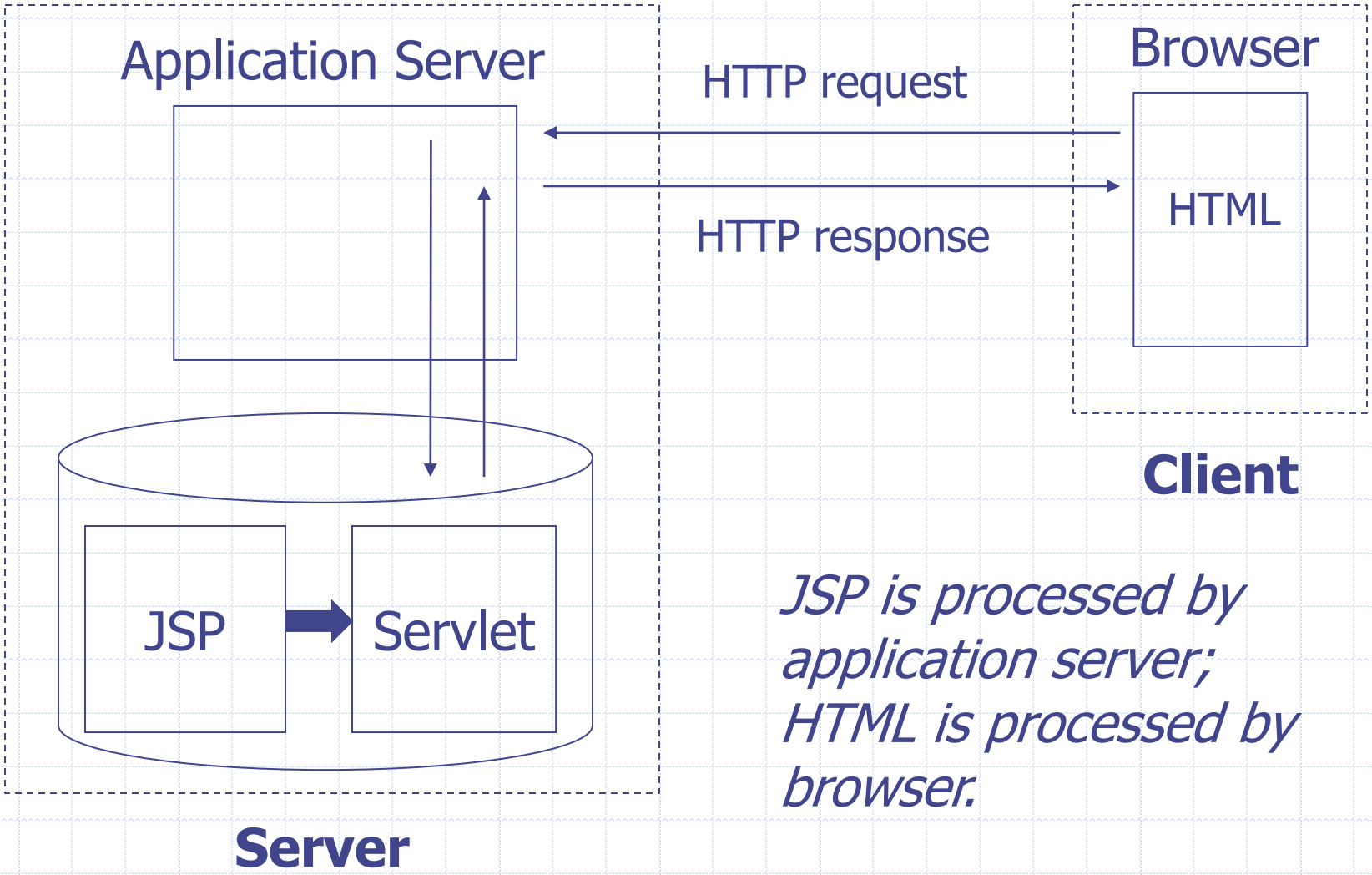
- Accessing JSP is similar to accessing static resources like HTML pages and images – it's based on *path* rather than URL mapping like for servlets

How JSP Works ...



Automatically done by server

… How JSP Works



JSP is processed by application server; HTML is processed by browser.

What A JSP-Servlet Looks Like

- Eclipse's Tomcat folder is at $WORKSPACE\.metadata\.plugins\org.eclipse.wst.server.core
- Look under $TOMCAT_HOME/work/Catalina/localhost/context_name

Two Ways to Use JSP

- JSP the old way (i.e. *wrong way*)
  - HTML + Java code
  - E.g. `Add1.jsp`
- JSP as a *template engine*
  - HTML + EL/JSTL
  - E.g. `Add2.jsp`

Add1.jsp

```
<!DOCTYPE html>
<html>
<head><meta charset="UTF-8"><title>Add1</title></head>
<body>
<p>The sum of <%= request.getParameter("a") %>
and <%= request.getParameter("b") %> is
<%= Integer.parseInt(request.getParameter("a"))
+ Integer.parseInt(request.getParameter("b")) %>
</p>
</body>
</html>
```

Add2.jsp

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8"><title>Add2</title>
</head>
<body>
  <p>The sum of ${param.a} and ${param.b} is
  ${param.a + param.b}.</p>
</body>
</html>
```

JSP The Old Way – RequestCounter.jsp

```
<%! int counter = 1; %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Request Counter</title>
</head>
<body>You are visitor #
<%= counter %>.
<% ++counter; %>
</body>
</html>
```

JSP Declaration

JSP Expression

JSP Scriptlet

JSP Scripting Elements

Understand JSP Scripting Elements

```
int counter = 1;

_jspService()
{
    out.println("<!DOCTYPE html>");
    out.println("<head>");
    ...
    out.println(counter);
    ++counter;
    out.println("</body>");
    out.println("</html>");
}
```

Why The Old Way Is The Wrong Way

- Mixing presentation and processing
  - Hard to read, debug, or maintain
- No clean and easy way to reuse code

```
<% if( Math.random() < 0.5) { %>
<H1>Have a <I>nice</I> day!</H1>
<% } else { %>
<H1>Have a <I>lousy</I> day!</H1>
<% } %>
```

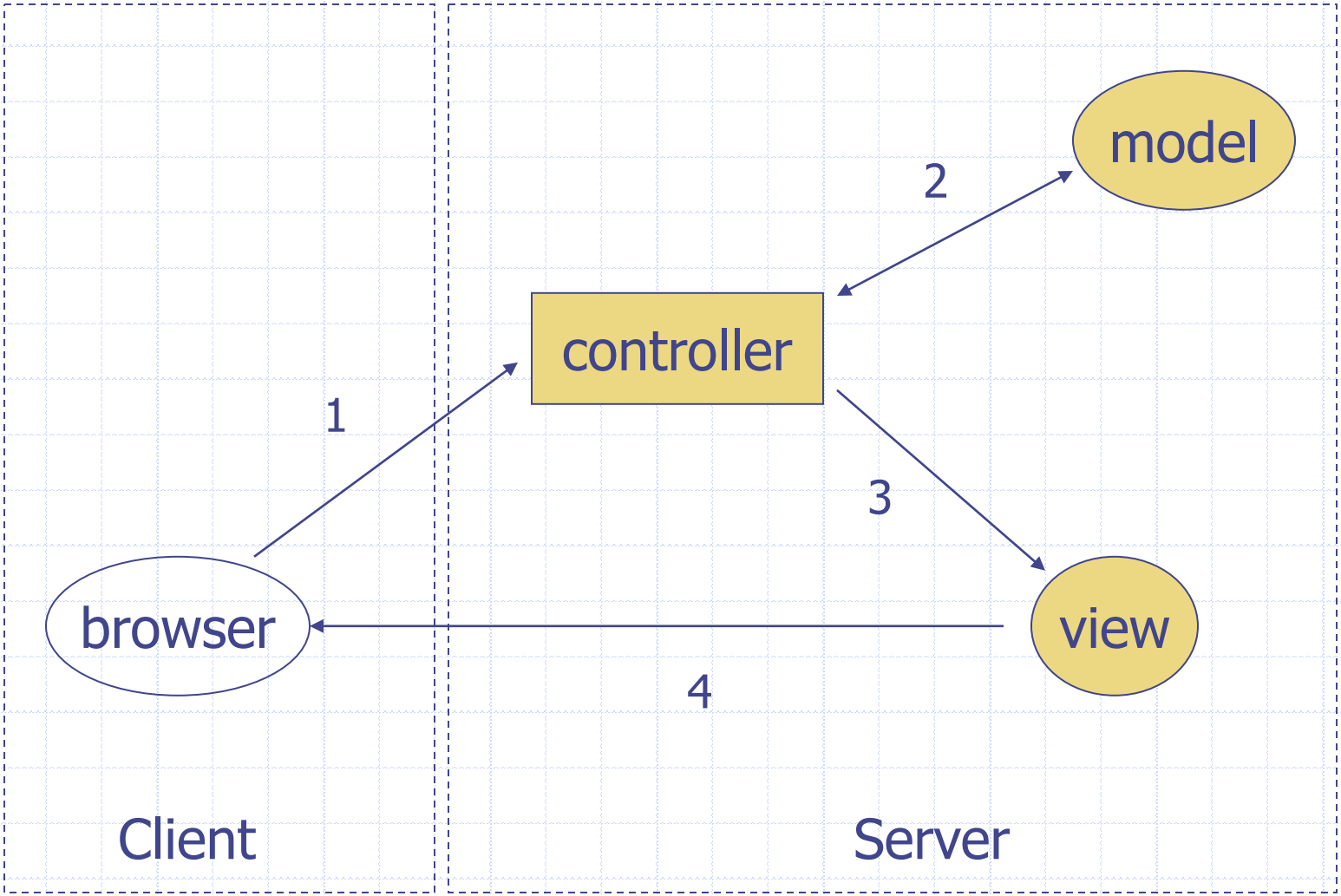
JSP As A Template Engine

- A template engine combines templates with data to produce documents
- In other words, JSP is used to *display data*, no more and no less
  - *Who's going to do the processing?*
  - *Where does data come from?*
  - *How to display data without using Java?*

The MVC Architecture

- A software architecture commonly used for web and GUI applications
- Model-View-Controller (MVC)
  - Model: data
  - Controller: for processing
  - View: for display

MVC in a Web Application ...



... MVC in a Web Application

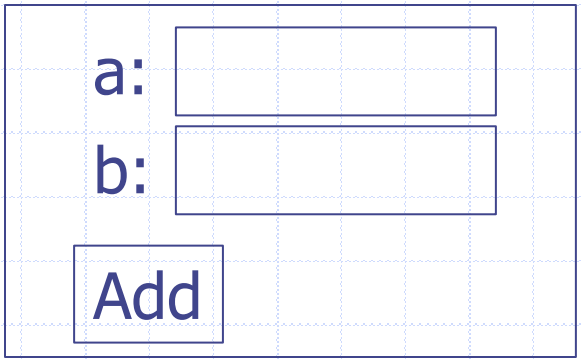
1. Browser sends a request to controller
2. Controller processes the request, updates some data
3. Controller forwards the request and data to view
4. View generates the response that is sent back to the client

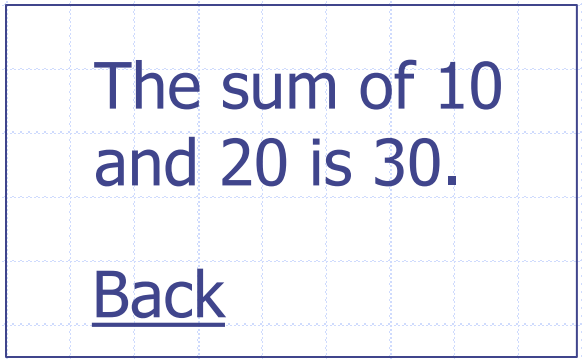
MVC in Java Web Application

- Model: regular Java classes (a.k.a. bean, or POJO – Plain Old Java Object)
  - E.g. `GuestBookEntry`
- Controller: servlet
- View: JSP

Add Example Using MVC …

- Data
  - Integer `a`, `b`, and `sum`
- Operations
  - Display form
  - Process input and display result





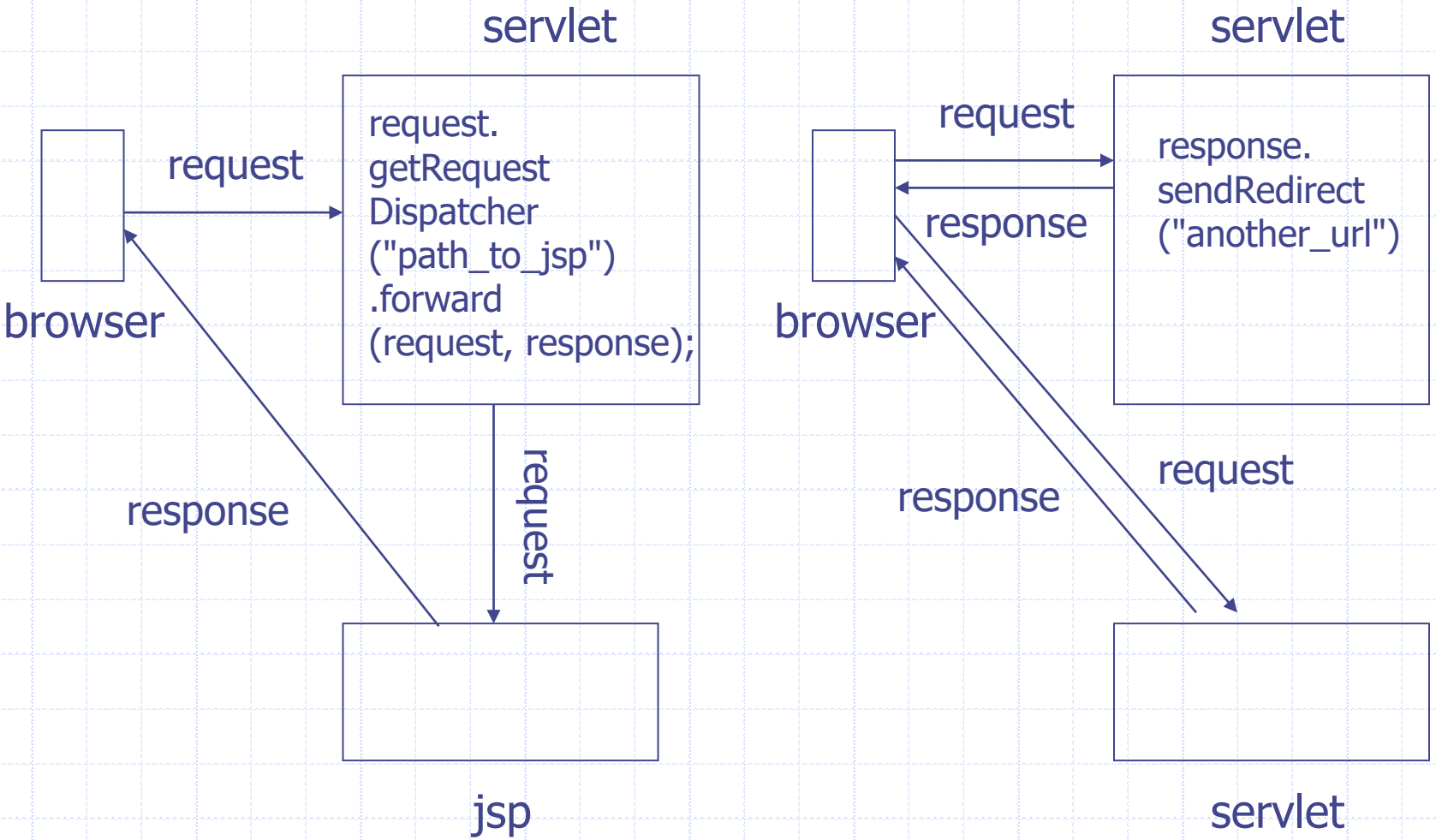
... Add Example Using MVC

- Web Application = Data + Operations
  - Create data model classes when necessary
  - One servlet per operation, except for form processing operations
    - `doGet()` to display form
    - `doPost()` to process form submission
  - One JSP per view (i.e. page)

Forward Request From Controller to View


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

Forward vs. Redirect



Send Data From Controller to View

- Data in application/session/request scope is kept for the duration of the application/session/request
- Request scope is usually used for passing data from controller to view

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

More About the MVC Example

- Requests should always go to controllers first
  - Hide JSPs under `/WEB-INF` so users cannot access them directly
- Controllers do not generate HTML
  - No `out.println()`
- JSPs are only used for display
  - No Java code in JSP