Module 5

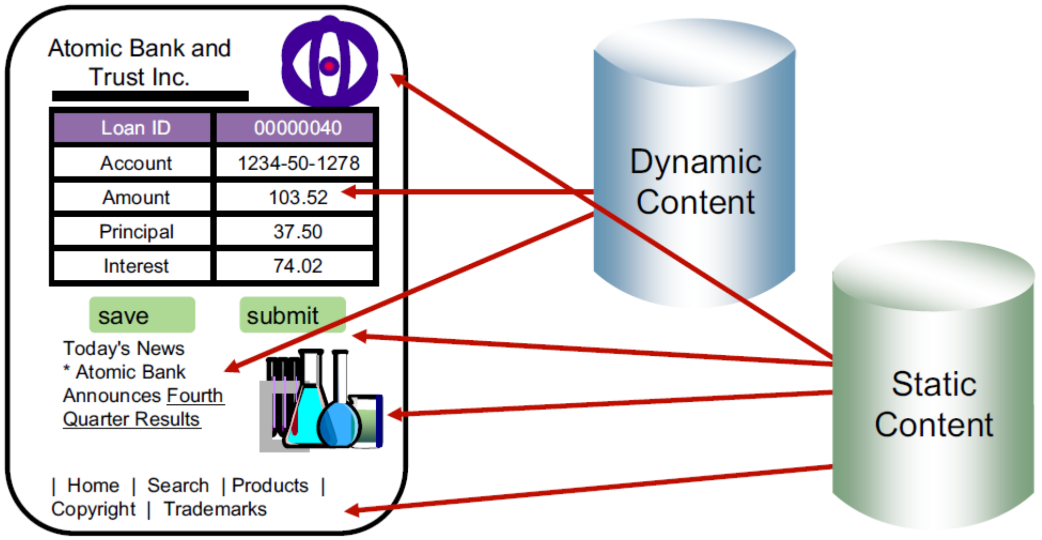
JSP Overview

Course Objectives

- After completing this module, you should be able to:
  - Explain JavaServer Pages technology
  - Identify the role of JSP pages within Web applications
  - Describe the JSP execution model
  - Discuss the JSP lifecycle

Content within a Web Page

- Content delivered to a client is composed of:
  - Static or non-customized content
  - Customized content
- Page layout and style are managed through HTML and XSL



What is JavaServer Pages?

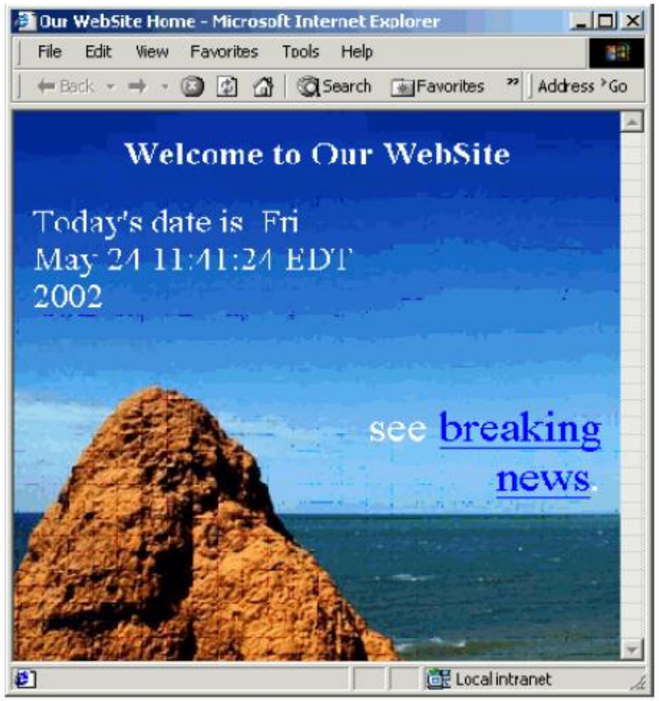
- JavaServer Pages technology that lets you mix static HTML with dynamically-generated HTML
- JSP technology allows server-side scripting:
  - Static tags are HTML, XML, or another markup language.
  - Dynamic content generated by scripting code
- A JSP file (with an extension of .jsp) contains any combination of:
  - JSP syntax
  - Markup tags such as HTML or XML

JSP example

```
<HTML>
<HEAD><TITLE>Our WebSite
   Home</TITLE></HEAD>
<BODY background="image.jpg"
    text="#ffffff">
<TABLE>
<TR><TD>
<H1>Welcome to Our WebSite</H1>
</TD></TR><TR><TD>
<H3>Today's date is

<%= new java.util.Date() %>

</H3></TD>
<TD>see <A href="breaking.html">
breaking news</A>.
</TD></TR>
</TABLE>
</BODY>
</HTML>
```


Our WebSite Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Address Go
Local intranet

Welcome to Our WebSite

Today's date is Fri May 24 11:41:24 EDT 2002

see breaking news.

JSP syntax elements

- JSP 2.0 syntax consists of:
  - Directives
    - Instructions to the JSP engine or compiler
  - Scripting
    - Declarations – additional methods and variables to be generated into the JSP servlet
    - Scriptlets – inline Java code
    - Expressions – Java code that resolves to Strings
  - Actions available within the JSP servlet
    - Standard actions for bean usage and flow control
    - Custom actions can be added

JSP or Servlet?

- Use servlets to:
  - Determine what processing is needed to satisfy the request
  - Validate input
  - Work with business objects to access the data and perform the processing needed to satisfy the request
  - Control the flow through a Web application
- Use JSP pages for displaying the content generated by your Web application

JSP Benefits (1 of 2)

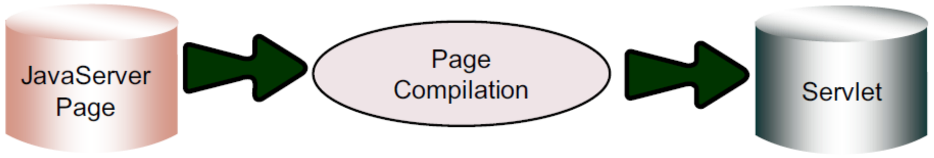
- Separation of static from dynamic content
  - The logic to generate the dynamic content is kept separate from the static presentation by encapsulating it within external (JavaBeans) components
  - Separation of workload
- Write once, run anywhere
  - Easily moved between platforms, no rewriting necessary
- Java EE-Compliant
  - The Java EE Blueprint recommends using JSP pages over servlets for the presentation of dynamic data

JSP Benefits (2 of 2)

- Leverages the Servlet API
  - The JSP specification is a standard extension defined on top of the Servlet API
- Reuse of components and tag libraries
  - JSP technology emphasizes the use of reusable components such as JavaBeans, Enterprise JavaBeans, and tag libraries
- High-quality tool support
  - One goal of the JSP design is to enable the creation of JSP development tools, such as Page Designer in Application Developer

JSP execution model (1 of 2)

- A JSP page is executed in a Web container
  - The Web container delivers client requests to the JSP page, and returns the page's response to the client
- The JSP page is converted into a servlet (JSP servlet) and executed
- This process is known as *page compilation*:
  - JSP source is parsed
  - Java servlet code is generated
  - This JSP servlet is compiled, loaded, and run

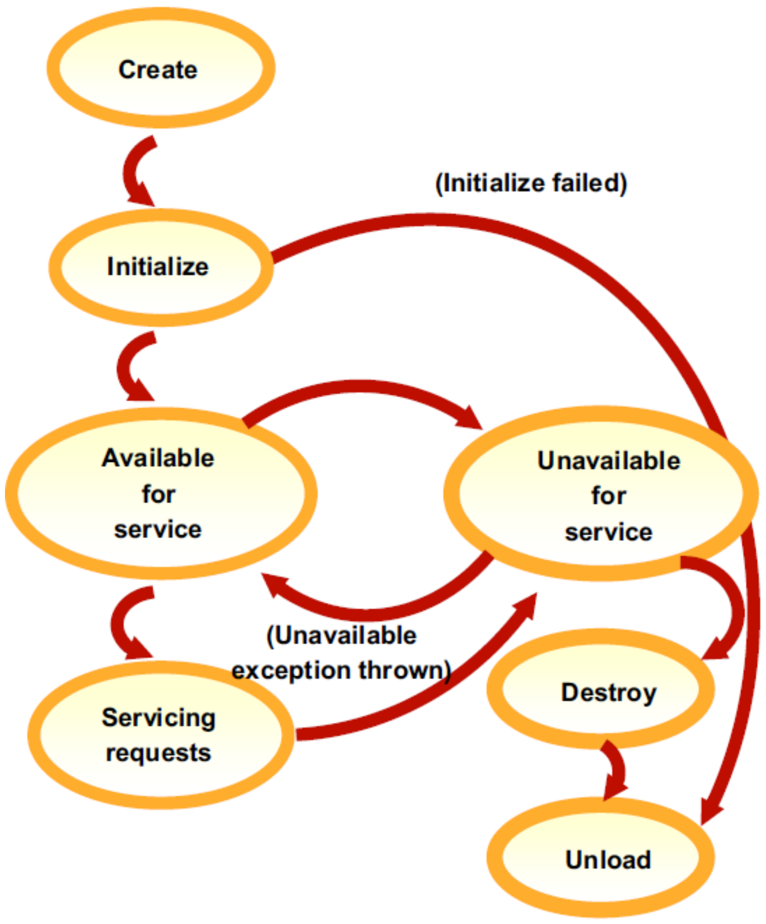


JSP execution model (2 of 2)

- Compilation is only performed as needed:
  - No class file exists, or
  - JSP has been updated since last compilation
- Precompilation
  - The JSP 2.0 specification requires support by the container for precompilation
- All request parameters starting with jsp_ are reserved
  - JSP pages should ignore parameters starting with jsp_

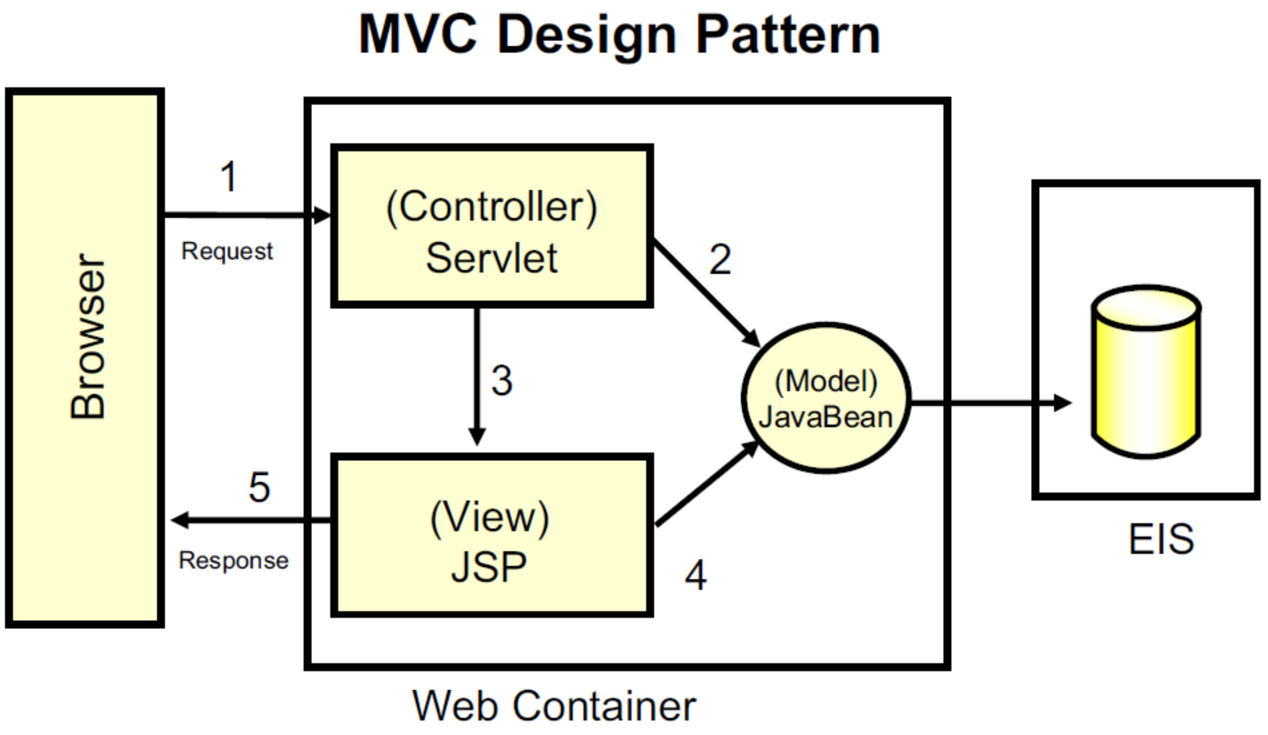
The JSP lifecycle

- JSP files are compiled into servlets and have the same lifecycle as all other servlets



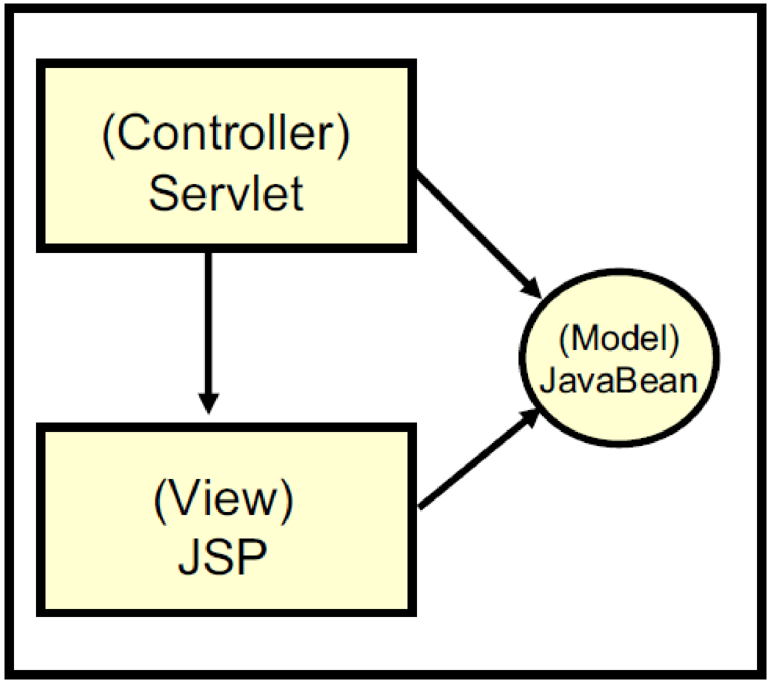
Typical JSP access model

- The request is sent to a servlet that generates dynamic content, and calls a JSP page to send the content to the browser, as shown:



Subsequent JSP requests

- Next time the JSP page is requested:
  - The JSP Web container determines whether the .jsp file has changed since it was loaded
    - If changed, the Web container recompiles the page, loads the newly generated servlet, and then invokes the service method
    - If it has not changed, the current instance is used, and the service method is called
  - The rest of the processing is unchanged



JSP servlet class

- Class (servlet) that is created by the Page Compiler must implement the HttpJspPage interface
- This interface defines the _jspService method where script content is written

```
void _jspService( HttpServletRequest request,
                  HttpServletResponse response)
                        throws IOException, ServletException
```

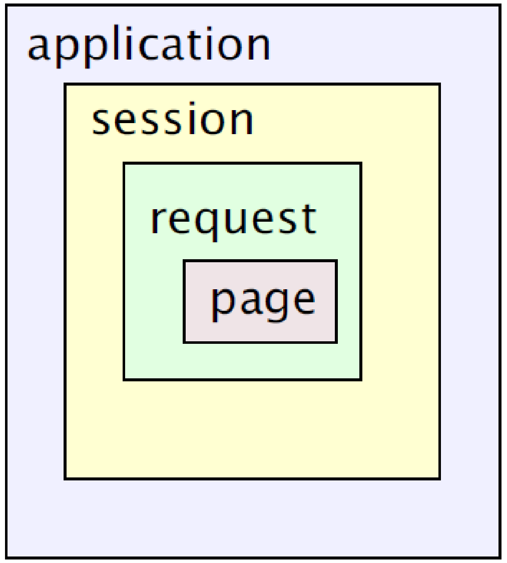
- A JSP page may include initialization and cleanup routines

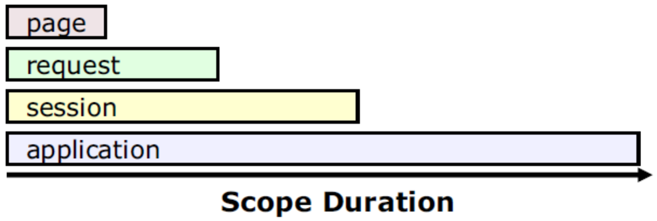
```
public void jspInit()
public void jspDestroy()
```

- Output is written to a JspWriter, which supports buffer management

Scope attributes

- A JSP page can access objects at run time via one of four different scopes (or holder objects)
  - Page
    - The current JSP page, used with Custom Actions
  - Request
    - The current HttpServletRequest object
  - Session
    - The current HttpSession object
  - Application
    - The current ServletContext object





Page Scope

- Objects are available only within the page where they are created
- References to these objects are released after the response is sent back to the client, or when the request is forwarded to somewhere else
- Use setAttribute(String, Object) to set and getAttribute(String) to retrieve
- References to objects with page scope are stored in the pageContext object

Request Scope

- Objects are available within the page where they are created, and pages to which the current request is forwarded
- References to these objects are released after the response is sent back to the client
- References to objects with request scope are stored in the request object
- To store objects in the request context:
  - Use req.setAttribute(String, Object) in the servlet
    - req is the parameter of type HttpServletRequest that is passed to the servlet
  - In a JSP page, use request.getAttribute(name) to retrieve values

Session Scope

- Objects are available from servlets and JSP pages processing requests that are in the same user session
- References to the object are released after the associated session ends
- References to objects with session scope are stored in the session object
- To store objects in the session context:
  - In the servlet, use code such as:

```
HttpSession session = req.getSession;
session.setAttribute(name, object);
```

  - In a JSP page, use `session.getAttribute(name)` to retrieve values

Application Scope

- Objects are available from servlets and pages that are processing requests in the same application
- References to the object are released when the run-time environment reclaims the ServletContext object
- References to objects with application scope are stored in the application object
- To store objects in the application context:
  - In the servlet, use code such as:

```
ServletContext context = getServletContext;
context.setAttribute(name, object);
```

  - In a JSP page, use `application.getAttribute(name)` to retrieve values

JSP positioning

- Advantages:
  - Can use page authoring tools to develop presentation
  - Allows HTML framework to manage composition of content
  - Full support for Java on server-side
- Warnings:
  - Potential to write too much Java code in the HTML document
  - Potential to use for more than just presentation (view)

Checkpoint

1. What types of content are most Web pages composed of?
2. What are the disadvantages of using servlets for Web pages?
3. What executes when a JSP page is invoked?

Module Summary

- This unit covered the following topics:
  - Explain JavaServer Pages technology
  - Identify the role of JSP pages within Web applications
  - Describe the JSP execution model
  - Discuss the JSP lifecycle