Web Technologies

Java Server Pages (JSP)

Web
Technologies

Introduction

JSP Basics

JSP Scripting
Elements

MCS

Web
Technologies

Introduction

JSP Basics

JSP Scripting
Elements

Introduction

- Different technologies have been developed by Java to support Web and distributed computing
- J2EE (Java 2 Enterprise Edition) covers various technologies covering Web, services and communication technologies
- Our focus will be on Web components of the J2EE architecture
- Java supports a platform independent approach to enterprise application development
- Three main components for Web technologies are:
  - Servlets
  - Java Server Pages
  - Enterprise Java Beans (EJBs)

Servlets

- Servlets are server-side programs that allow application logic to be embedded in HTTP request-response process.
- Servlets provide means to extend the functionality of a Web server to enable dynamic content in HTML, XML, or other web languages
- Java Servlets are not invoked directly by the users
- Web container in which a Web application is deployed, invokes the relevant servlets of the Web application
- The presentation and application logic both are embedded in the servlets
- Presentation components are produced programmatically by the servlet.

MCS

Web Technologies

Introduction

JSP Basics

JSP Scripting Elements

5 / 13

Java Server Pages (JSP)

- Dynamic content generation can be achieved in two ways; programmatic content generation or template based content generation.
- Java servlets fall in the programmatic content generation
- JSP belongs to template based content generation
- HTML, XML, XHTML tags allow to define the static structure and content of a page.
- JSP tags and scriptlets written in the Java programming language are used to encapsulate the logic that generates the dynamic content
- It enables to keep the content design and development activities loosely coupled with the design and development of the application logic.

Enterprise JavaBeans

- EJB technology is a distributed component model for developing secure, scalable, transactional and multi-user components.
- These are reusable software units containing business logic

JSP Fundamentals

- A simple JSP might look like this:

```
<%@ page import=''java.util.Date'' %>
<html>
<body>
The current time is <%= new Date().toString() %>
</body>
</html>
```

- A JSP page is compiled by the web container into a servlet.
- The above code will display the current time of the system.
- Basic JSP tags fall into following categories:
  - Directives: These affect the overall structure of the servlet that results from translation
  - Scripting elements: These allow to use Java code in the JSP page
  - Actions: These are special tags available to affect the runtime behaviour of the JSP

Web
Technologies

Introduction

JSP Basics

JSP Scripting
Elements

```
<html>
<body>
Good
<%
java.util.Calendar currTime = new
java.util.GregorianCalendar();
if (currTime.get(currTime.HOUR_OF_DAY) < 12)
{ %>
Morning!
<% }
else if (currTime.get(currTime.HOUR_OF_DAY) < 18)
{ %>
Afternoon!
<% } else { %>
Evening!
<% } %>
</body>
</html>
```

MCS

Web
Techologies

Introduction

JSP Basics

JSP Scripting
Elements

8 / 13

JSP Directives

- These serve as JSP messages sent to the JSP container from the JSP
- Directives provide Tomcat with information it needs to translate the JSP page
- These are used to set the global value such as class declarations, output content type etc.
- 3 main directives:
    - Page directive
    - include directive
    - taglib directive
- Directives start with <%@ and end with %>.

Page Directive

- The page directive is used to define and manipulate a number of important page-dependent attributes that affect the whole JSP (entire compiled class file).
- General syntax of the page directive is:
  <%@page ATTRIBUTES %>

MCS

Web
Technologies

Introduction

JSP Basics

JSP Scripting
Elements

Page Directives

- Attributes are name value pairs as described below:
  - language : Defines the scripting language to be used. Default value: Java
  - import : comma separated list of packages or classes just like import state in usual java code
  - session : Specifies whether the page participates in an HTTP session. When "true" the implicit object named session (referring to javax.servlet.http.HttpSession) is available for the page. If "false" the page does not participate in a session and the implicit session object is unavailable.

Include Directive

- The include directive instructs the container to include the content of the resource in the current JSP.
- Resources can be includes at runtime.
- <%@ include file=''Filename'' %>
- The included file can be either a static file (such as an HTML file) or another JSP

```
<html>
<head><title>Include directive test page</title> </head>
<body>
<h1>Include directive test page</h1>
<%@ include file=''copyright.html'' %>
</body>

</html>
```

JSP Scripting Elements

- JSP scripting elements allow Java code insertion - variable or method declarations, scriptlets (arbitrary Java code) and expressions
- Variables, loops, decisions can be inserted into JSP
- Different Java objects can also be used by using these scripting elements

Declaration

- A declaration is a block of Java code in a JSP that is used to define class-wide variables and methods in the generated servlet.
- A declaration is enclosed within <%! and %> tags.
- In this case, a new variable will be created for each new instance of the compiled JSP (which is a servlet).
- By default, Tomcat server creates one instance of each servlet and allows access to the instance by using multithreading.
- Thus, if variables are declared in the declaration block, these will be available to all requests and these can modify the variable values.

```
<%!
int visits=0;
%>
No.  of times this page visited:  <b><%= i=i+1 %></b>
```

- Output of this code will increment the value of i each time the page is visited.

Scriptlet

- A scriptlet is a block of Java code that is executed during the request-processing time
- It is enclosed within <% and %> tags
- The code in the scriptlet is added to the service() method of the compiled servlet. It is therefore, processes for every request that the servlet receives.

```
<%
for (int i=0;i<10;i++) {
%>
This is line no: <b><%= i %> </b> <br>
<% } %>
```