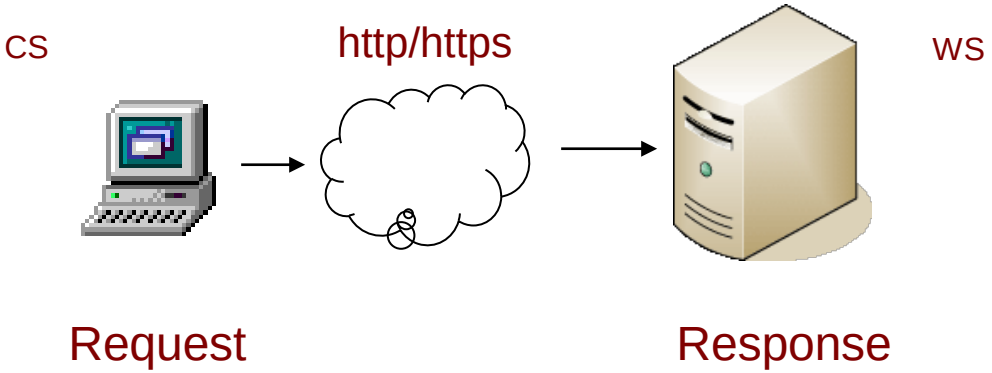
Introduction to JSP

Agenda

- Web Application
- Servlet
- JSP
- JSP Lifecycle
- Tag
- Custom Tag

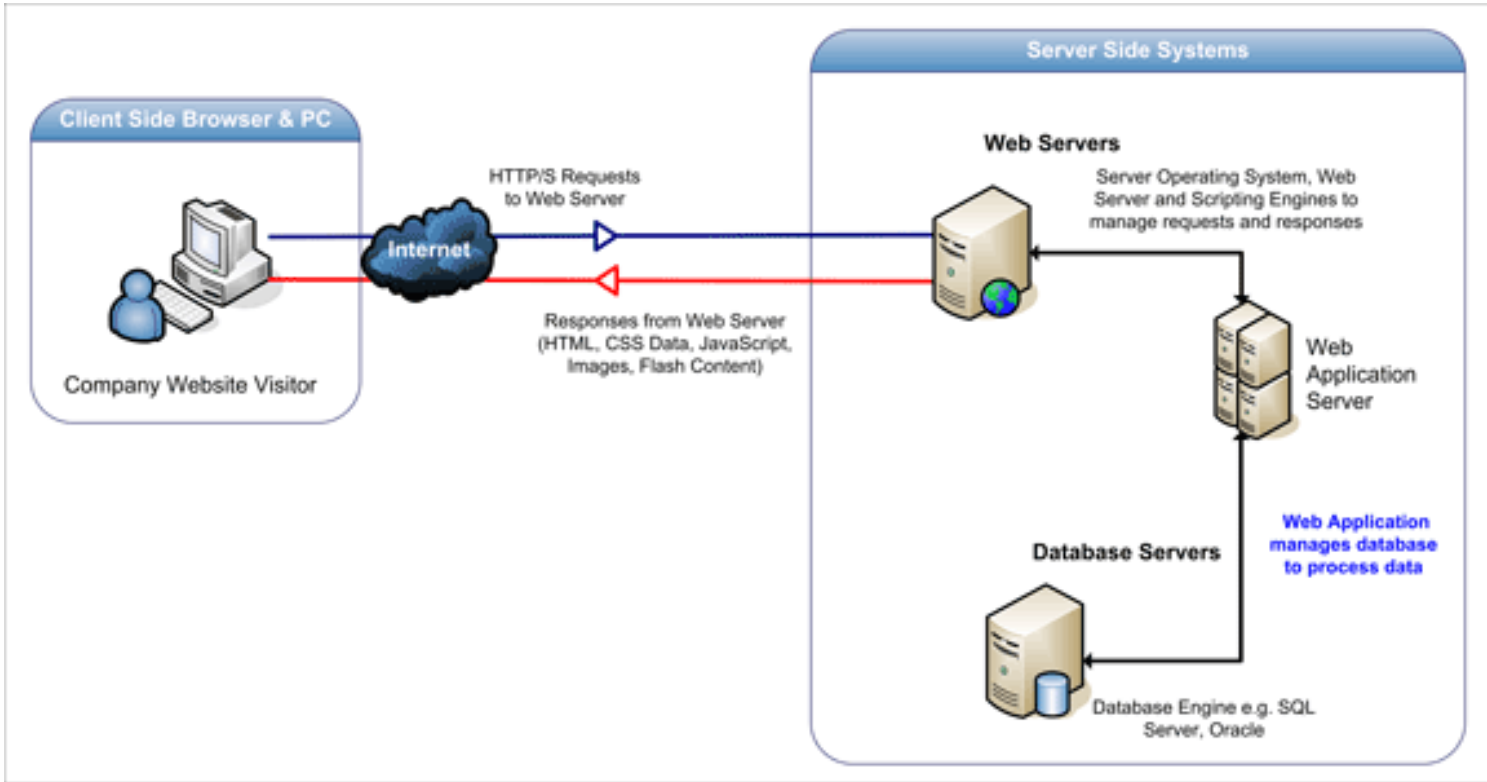
Web Application

- Web Request & Response
  - HTTP/HTTPS Protocol



Web Application

- How do web application work?



Servlet

Servlet

- What is Servlet?
  - A java programming language class to extend and enhance web servers
  - Run on web server and build web page
  - Server side components
    - Provide dynamic content
    - Process and/or store the data
    - Manage information state

Servlet

- Specification

Tomcat	Servlet / JSP	JDK
3.3.x	2.2 / 1.1	1.1
4.1.x	2.3 / 1.2	1.3
5.5.x	2.4 / 2.0	1.4
6.0.x	2.5 / 2.1	1.5
7.0.x	3.0 / 2.2	1.6
8.0.x	3.1 / 2.3	1.7
9.0.x	4.0 / 2.4	1.8

Glassfish	Servlet / JSP
2.1.x	2.5 / 2.1
3.0.x	3.0 / 2.2
3.1.x	3.0 / 2.2
4.0.x	3.1 / 2.3

Servlet

- Basic Servlet Structure

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SomeServlet extends HttpServlet {
        public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

                // Use "request" to read incoming HTTP headers (cookies)
                // and HTML form data (data the user entered and submitted)

                // Use "response" to specify the HTTP response and headers

                PrintWriter out = response.getWriter();
                // Use "out" to send content to browser
        }
}
```

Servlet

- Servlet Generate Plain Text

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
                public void doGet(HttpServletRequest request,
                                HttpServletResponse response)
                                throws ServletException, IOException {

                                PrintWriter out = response.getWriter();
                                out.println("Hello World");
                }
}
```

Servlet

- Servlet Generate HTML

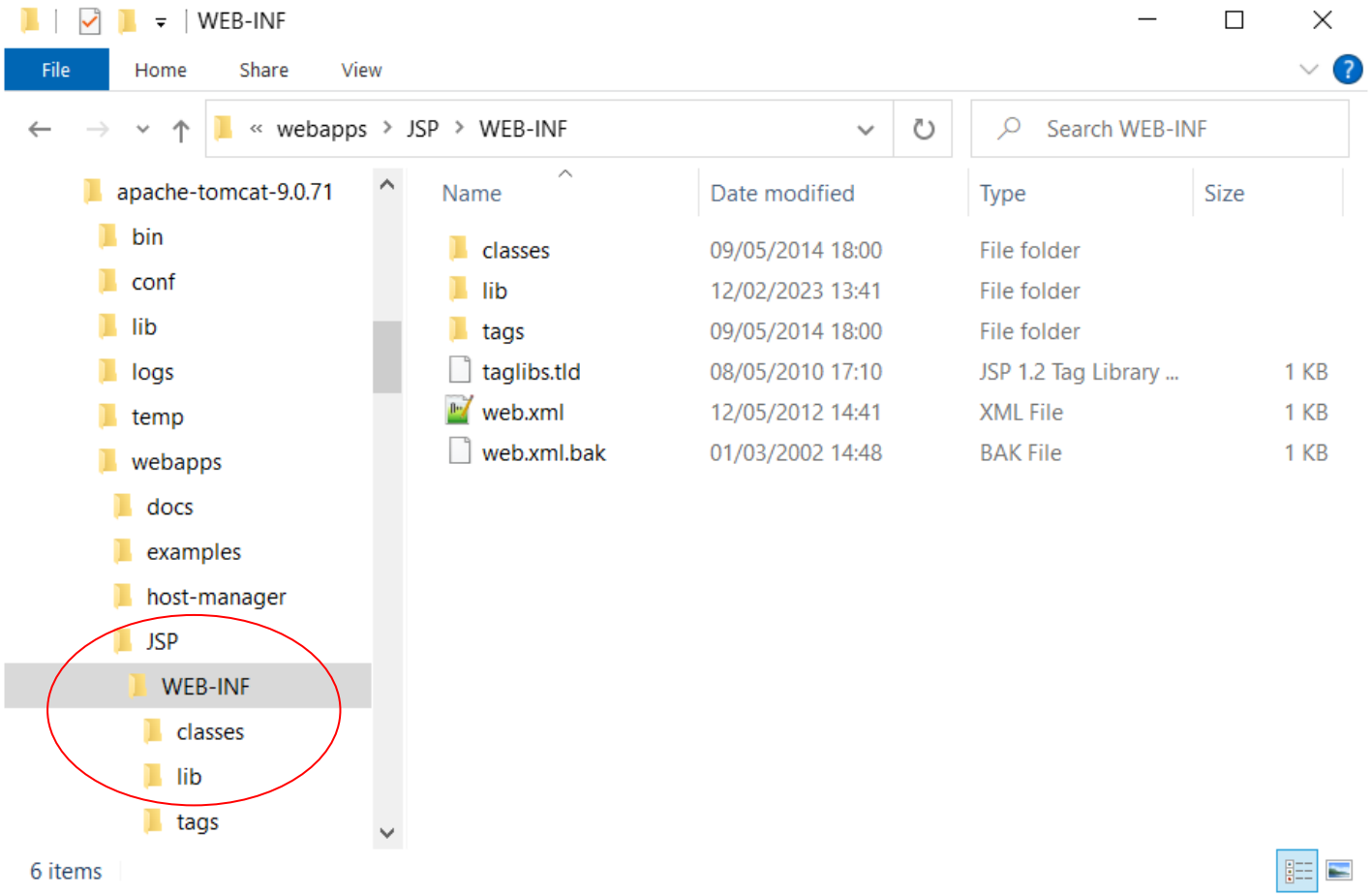
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class  HelloServlet extends HttpServlet {
                public void service(HttpServletRequest request, HttpServletResponse
response)   throws ServletException, IOException {

                                PrintWriter out = response.getWriter();
                                out.println("<HTML>");
                                out.println("<TITLE>Hello Servlet</TITLE>");
                                out.println("<HEAD></HEAD>");
                                out.println("<BODY>Hello World</BODY>");
                                out.println("</HTML>");
                                out.flush();
                                out.close();
                }
}
```

Servlet

- Install Servlet
  - download & install tomcat
  - create project under tomcat webapps
    - ex. C:\apache-tomcat-9.0.71\webapps\JSP
  - create WEB-INF folder under project
    - ex. C:\apache-tomcat-9.0.71\webapps\JSP\WEB-INF
  - create classes & lib folder

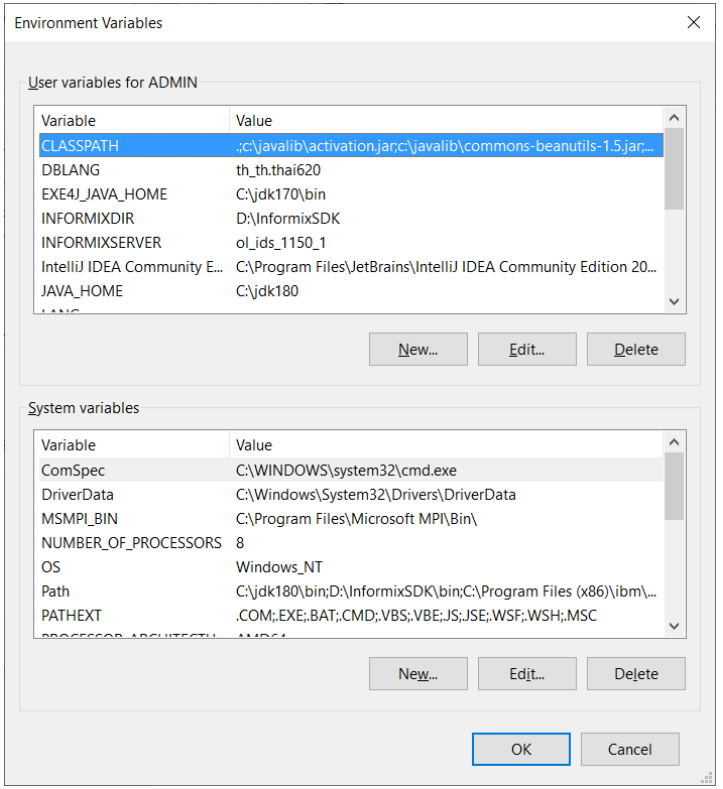
Servlet

- Install Servlet



Servlet

- Compile Servlet
  - set CLASSPATH with servlet API
    - set CLASSPATH=.;C:\apache-tomcat-9.0.71\lib\servlet-api.jar;
    - set environment variables



Servlet

- Compile Servlet
  - compile java class

Servlet

- Config Servlet
  - WEB-INF/web.xml

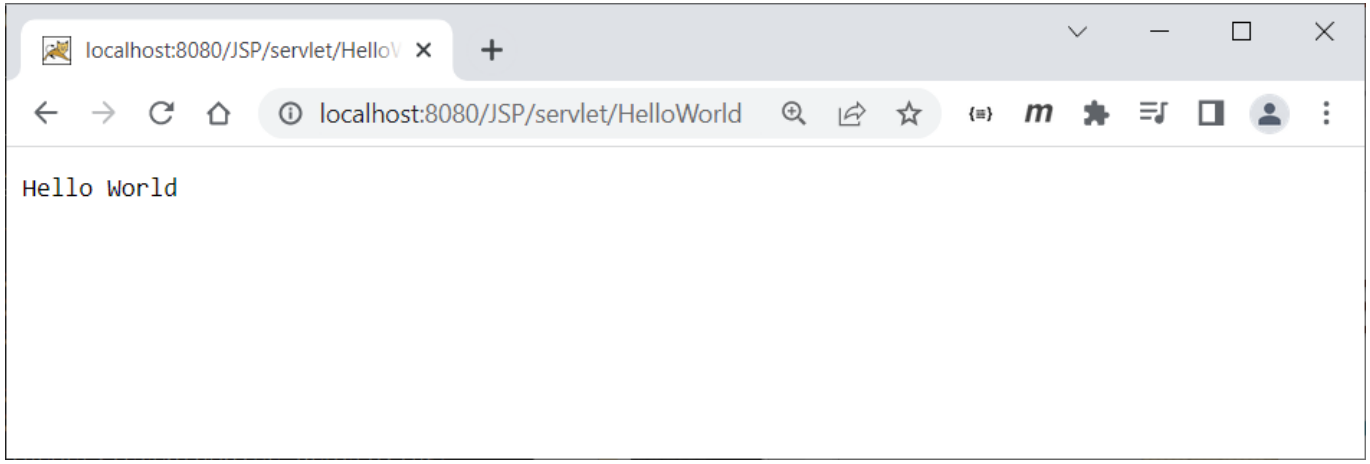
```
<servlet>
        <servlet-name>HelloWorld</servlet-name>
        <servlet-class>HelloWorld</servlet-class>
</servlet>

<servlet-mapping>
        <servlet-name>HelloWorld</servlet-name>
        <url-pattern>/servlet/HelloWorld</url-pattern>
</servlet-mapping>
```

Servlet

- Running Servlet
  - set JAVA_HOME environment variable
    - set JAVA_HOME=c:\jdk180
  - start tomcat
    - cd C:\apache-tomcat-9.0.71\bin
    - startup

Servlet

- Running Servlet
  - access browser go to http://localhost:8080/JSP/servlet/HelloWorld



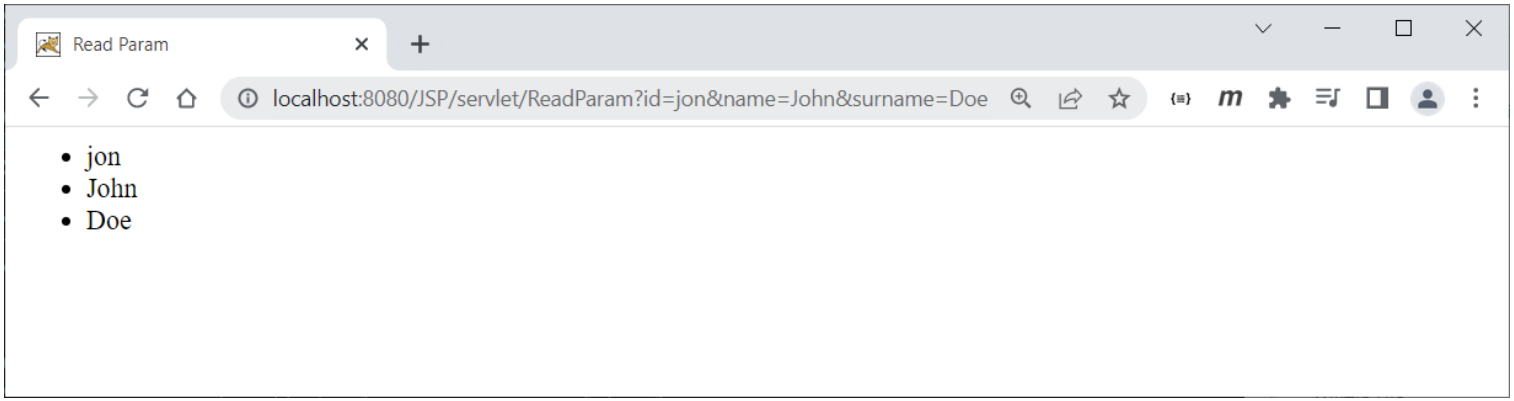
Servlet

- Reading Parameters

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ReadParam extends HttpServlet {
        public void doGet(HttpServletRequest request,HttpServletResponse response)
                throws ServletException, IOException {
                        response.setContentType("text/html");
                        PrintWriter out = response.getWriter();
                        out.println("<HTML>");
                        out.println("<TITLE>Read Param</TITLE>");
                        out.println("<HEAD></HEAD>");
                        out.println("<BODY><UL>");
                        out.println("<LI>"+request.getParameter("id"));
                        out.println("<LI>"+request.getParameter("name"));
                        out.println("<LI>"+request.getParameter("surname"));
                        out.println("</UL></BODY>");
                        out.println("</HTML>");
        }
        public void doPost(HttpServletRequest request,HttpServletResponse response)
                throws ServletException, IOException {
                doGet(request, response);
        }
}
```

Servlet

- Reading Parameters
  - http://localhost:8080/JSP/servlet/ReadParam?id=jon&name=John&surname=Doe



Servlet

- Listing Data Form - I

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

public class ListingParam extends HttpServlet {
        public void doGet(HttpServletRequest request,HttpServletResponse response)
                throws ServletException, IOException {
                response.setContentType("text/html");
                PrintWriter out = response.getWriter();
                out.println("<HTML>");
                out.println("<TITLE>Listing Param</TITLE>");
                out.println("<HEAD></HEAD>");
                out.println("<TABLE BORDER=1 ALIGN=CENTER>");
                out.println("<TR><TH>Parameter Name</TH><TH>Parameter
Value(s)</TH></TR>");
```

Servlet

- Listing Data Form - II

```
                Enumeration paramNames = request.getParameterNames();
                for(;paramNames.hasMoreElements();) {
                        String paramName = (String)paramNames.nextElement();
                        out.println("<TR><TD>" + paramName + "<TD>");
                        String[] paramValues = request.getParameterValues(paramName);
                        if (paramValues.length == 1) {
                                String paramValue = paramValues[0];
                                if (paramValue.length() == 0) out.print("<I>No Value</I>");
                                else out.print(paramValue);
                        } else {
                                out.println("<UL>");
                                for(int i=0; i<paramValues.length; i++) {
                                        out.println("<LI>" + paramValues[i]);
                                }
                                out.println("</UL>");
                        }
                }
                out.println("</TABLE></BODY></HTML>");
  }
  public void doPost(HttpServletRequest request,HttpServletResponse response)
                        throws ServletException, IOException {
                                doGet(request, response);
                }
}
```

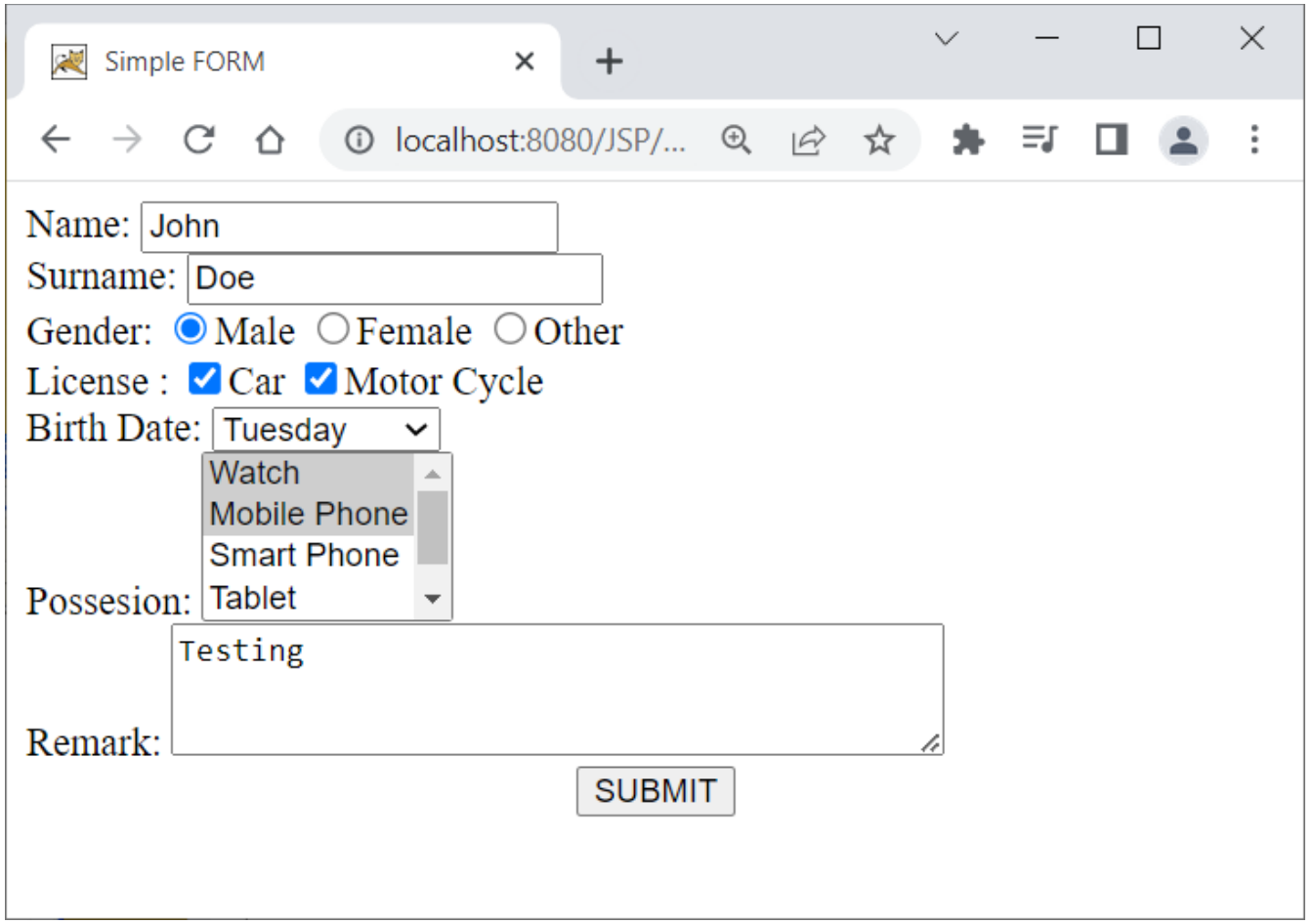
Servlet

- Listing Data Form – SimpleForm.html

```
<HTML>
<HEAD><TITLE>Simple FORM</TITLE></HEAD>
<BODY>
<FORM ACTION="servlet/ListingParam" METHOD="POST">
Name: <INPUT TYPE="TEXT" NAME="name"></INPUT><BR>
Surname: <INPUT TYPE="TEXT" NAME="surname"></INPUT><BR>
Gender:  <INPUT TYPE="RADIO" NAME="gender" VALUE="Male" checked>Male</INPUT>
                    <INPUT TYPE="RADIO" NAME="gender" VALUE="Male">Female</INPUT>
                    <INPUT TYPE="RADIO" NAME="gender" VALUE="Other">Other</INPUT><BR>
License : <INPUT TYPE="CHECKBOX" NAME="car" VALUE="permit">Car</INPUT>
                    <INPUT TYPE="CHECKBOX" NAME="motorcycle" VALUE="permit">Motor Cycle</INPUT><BR>
Birth Date: <SELECT NAME="birthdate">
                                        <OPTION value="sunday">Sunday</OPTION>
                                        <OPTION value="monday" selected>Monday</OPTION>
                                        <OPTION value="tuesday">Tuesday</OPTION>
                                        <OPTION value="wednesday">Wednesday</OPTION>
                                        <OPTION value="thursday">Thursday</OPTION>
                                        <OPTION value="friday">Friday</OPTION>
                                        <OPTION value="saturday">Saturday</OPTION>
                                        </SELECT><BR>
Possesion: <SELECT NAME="possesion" MULTIPLE="TRUE">
                                        <OPTION value="watch" selected>Watch</OPTION>
                                        <OPTION value="mobile" selected>Mobile Phone</OPTION>
                                        <OPTION value="smartphone">Smart Phone</OPTION>
                                        <OPTION value="tablet">Tablet</OPTION>
                                        <OPTION value="notebook">Note Book</OPTION>
                                        </SELECT><BR>
Remark: <TEXTAREA NAME="remark" ROWS=3 COLS=40></TEXTAREA><BR>
<CENTER><INPUT TYPE="SUBMIT" VALUE="SUBMIT"></CENTER>
</FORM>
</BODY>
</HTML>
```

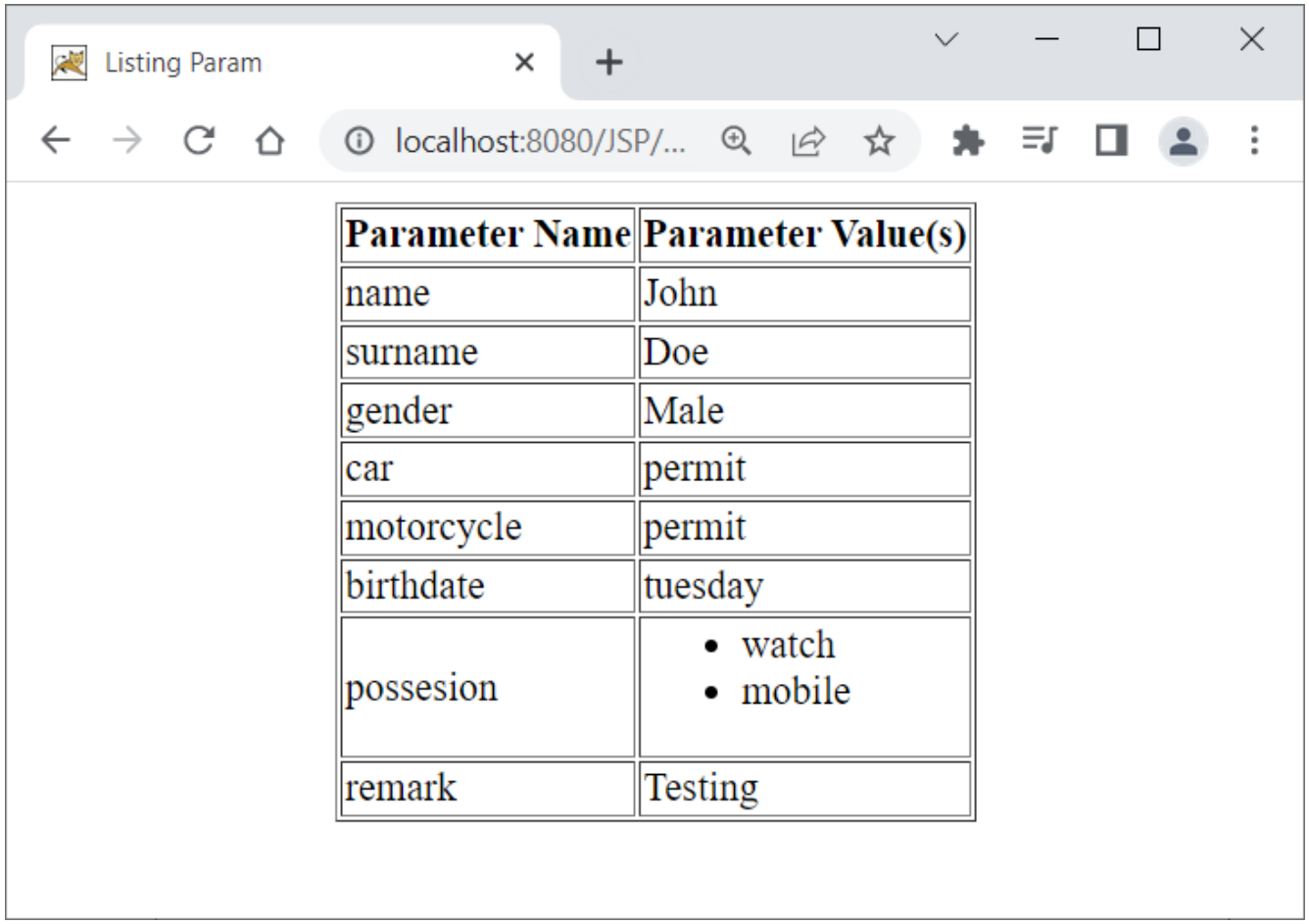
Servlet

- Listing Data Form



Servlet

- Listing Data Form



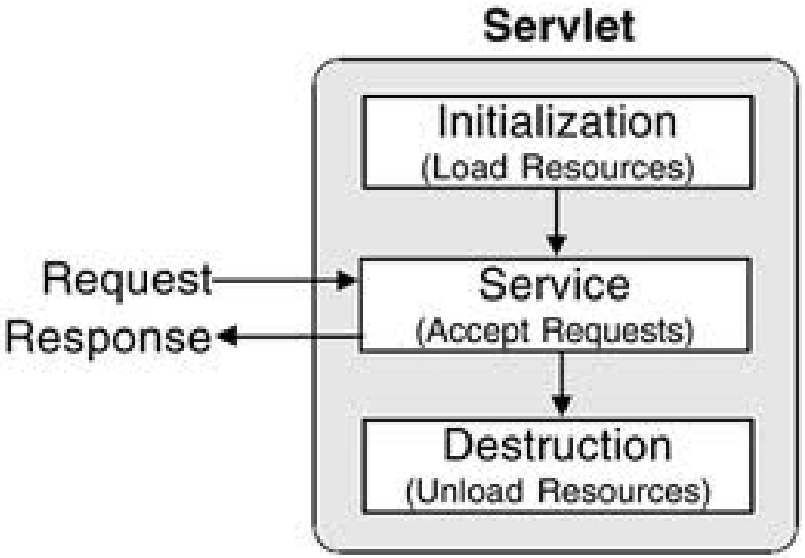
Parameter Name	Parameter Value(s)
name	John
surname	Doe
gender	Male
car	permit
motorcycle	permit
birthdate	tuesday
possesion	• watch • mobile
remark	Testing

Servlet

- Servlet Container
  - The main function of the container is to load, initialize and execute servlets
    - wait for HTTP request
    - construct a ServletRequest object and a ServletResponse object
    - load the servlet class and invoke service method, passing the ServletRequest and ServletResponse objects
    - calls the destroy method and unload when the servlet class is shut down

Servlet

- Servlet Life Cycle



Servlet

- Servlet Life Cycle
  - public void init(ServletConfig config) throws ServletException
  - public void service(ServletRequest request, ServletResponse response) throws ServletException, IOException
  - public void destroy()

Servlet

- Servlet Configuration - I

```
package com.fs.dev;

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class TheServlet extends HttpServlet implements java.io.Serializable {
public TheServlet() {
		super();
}
public void init(ServletConfig config) throws ServletException {
		super.init(config);
		System.out.println("----------------------------------------------------------------");
		System.out.println(getClass().getName()+" servlet initialize ...");
		System.out.println("----------------------------------------------------------------");
		System.out.println("init on "+config.getServletContext().getRealPath(""));
		java.util.Enumeration en = config.getInitParameterNames();
		for(;en.hasMoreElements();) {
				String key = (String)en.nextElement();
				System.out.println(key+"="+config.getInitParameter(key));
		}
}
```

Servlet

- Servlet Configuration - II

```
public void service(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
		response.setContentType("text/xml; charset=windows-874");
		java.util.Enumeration en = request.getParameterNames();
		for(;en.hasMoreElements();) {
				String key = (String)en.nextElement();
				String value = request.getParameter(key);
				System.out.println(key+"="+value);
		}
		String type = request.getParameter("type");
		if(type.equals("reset")) {
				StringBuffer buf = new StringBuffer();
				buf.append("<message type=\"reset\">");
				buf.append("<body>ok</body>");
				buf.append("</message>");
				sendResponse(response,buf.toString());
		} else {
				StringBuffer buf = new StringBuffer();
				buf.append("<message type=\"none\">");
				buf.append("<body>Unknown request</body>");
				buf.append("</message>");
				sendResponse(response,buf.toString());
		}
}
```

Servlet

- Servlet Configuration - III

```
public void destroy() {
		System.out.println(this+" destroying.");
}

protected void sendResponse(HttpServletResponse response, String text) {
		try {
				String header = "";
				if(!text.trim().equals("")) header = "<?xml version=\"1.0\"
encoding=\"windows-874\"?>";
				response.setDateHeader("Expires",System.currentTimeMillis());
				response.getOutputStream().println(header+text);
				response.getOutputStream().flush();
				response.getOutputStream().close();
		} catch (Exception ex) {
		}
}

}
```

Servlet

- Servlet Configuration – web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
   PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
   "http://java.sun.com/j2ee/dtds/web-app_2_3.dtd">
<web-app>
<servlet>
                <servlet-name>TheServlet</servlet-name>
                <servlet-class>com.fs.dev.TheServlet</servlet-class>
                <init-param>
                                <param-name>dburl</param-name>
                                <param-value>jdbc:mysql://127.0.0.1:3306/refdb</param-value>
                </init-param>
                <init-param>
                                <param-name>dbuser</param-name>
                                <param-value>root</param-value>
                </init-param>
                <init-param>
                                <param-name>dbpassword</param-name>
                                <param-value>root</param-value>
                </init-param>
                <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
                <servlet-name>TheServlet</servlet-name>
                <url-pattern>/servlet/TheServlet</url-pattern>
</servlet-mapping>
</web-app>
```

Servlet

- Servlet Life Cycle Events

Object	Event	Listener Interface and Event Class
Web context	Initialization and destruction	`javax.servlet.ServletContextListener` and `ServletContextEvent`
	Attribute added, removed, or replaced	`javax.servlet.ServletContextAttributeListener` and `ServletContextAttributeEvent`
Session	Creation, invalidation, activation, passivation, and timeout	`javax.servlet.http.HttpSessionListener`, `javax.servlet.http.HttpSessionActivationListener` and `HttpSessionEvent`
	Attribute added, removed, or replaced	`javax.servlet.http.HttpSessionAttributeListener` and `HttpSessionBindingEvent`
Request	A servlet request has started being processed by web components	`javax.servlet.ServletRequestListener` and `ServletRequestEvent`
	Attribute added, removed, or replaced	`javax.servlet.ServletRequestAttributeListener` and `ServletRequestAttributeEvent`

Servlet

- Servlet Life Cycle Events

```
package com.fs.dev;

import javax.servlet.*;
import javax.servlet.http.*;

public class TheSession implements javax.servlet.http.HttpSessionListener {
		private static int sessions;

		public void sessionCreated(javax.servlet.http.HttpSessionEvent e) {
				sessions++;
				System.out.println("session created : "+e+", id="+e.getSession().getId());
		}

		public void sessionDestroyed(javax.servlet.http.HttpSessionEvent e) {
				if(sessions>0) sessions--;
				System.out.println("session destroyed : "+e+", id="+e.getSession().getId());
				java.util.Enumeration en = e.getSession().getAttributeNames();
				for(;en.hasMoreElements();) {
						Object key = en.nextElement();
						System.out.println("destroy("+e.getSession().getId()+")
"+key+"="+e.getSession().getAttribute(key.toString()));
				}
		}
}
```

Servlet

- Servlet Life Cycle Events – web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE web-app
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/j2ee/dtds/web-app_2_3.dtd">

<web-app>

<listener>
        <listener-class>com.fs.dev.TheSession</listener-class>
</listener>


</web-app>
```

Servlet

- Servlet Context
  - An abstraction and mapping to the document root of web application and the resources.
  - Allow to get/set and change web application scope attribute values.

Servlet

- Servlet Context
  - javax.servlet.ServletContext

```
ServletContext context = getServletContext();

ServletContext context = request.getSession().getServletContext();
```

```
context.setAttribute("someValue", "aValue");

Object attribute = context.getAttribute("someValue");
```

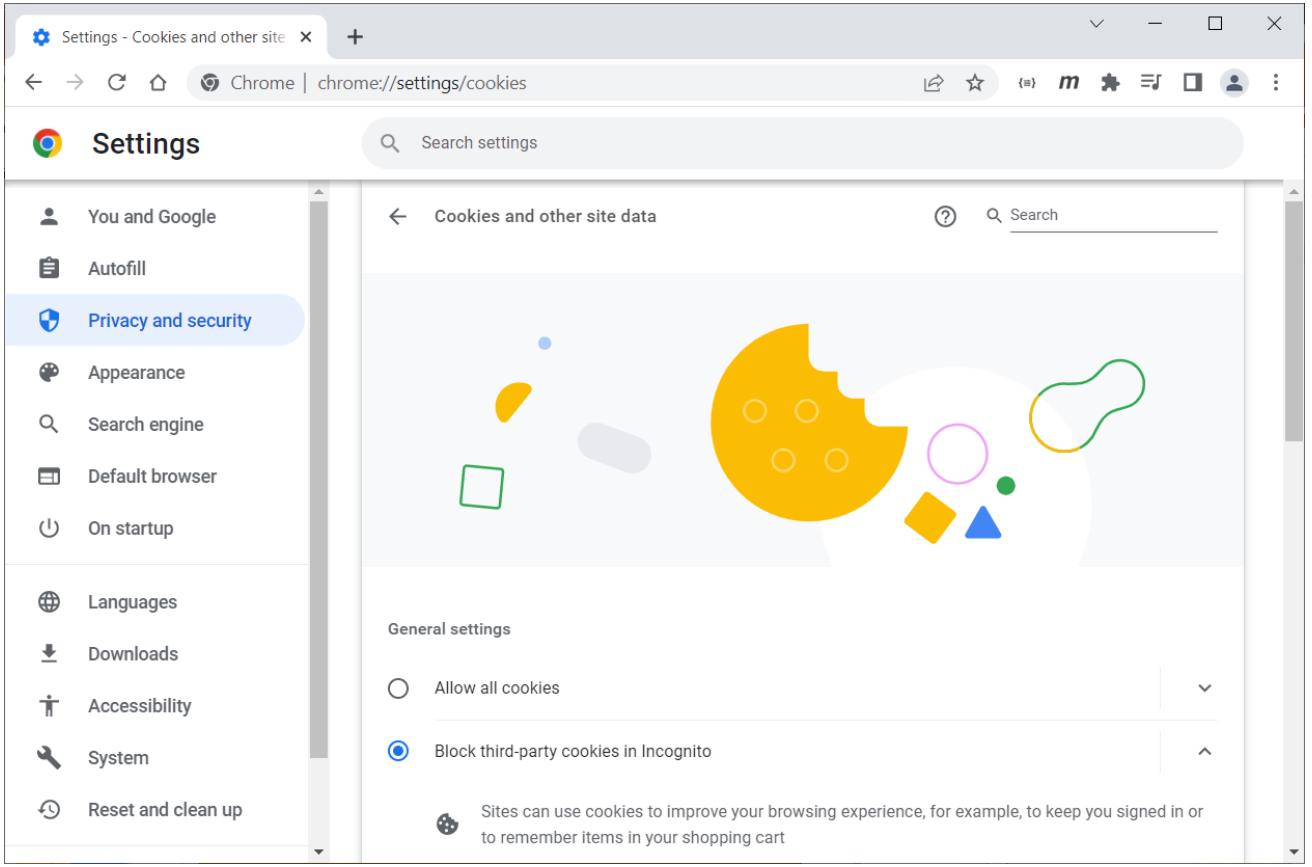
```
context.getRealPath("")

RequestDispatcher dispatcher = context.getRequestDispatcher("/index.html");
dispatcher.forward(request,response);
```

Servlet

- Cookies
  - A cookie is a piece of data stored by a website within a browser and then subsequently sent back to the same website by the browser.
    - Cookies are usually limited to 4096 bytes
    - At least 20 cookies per unique host or domain name

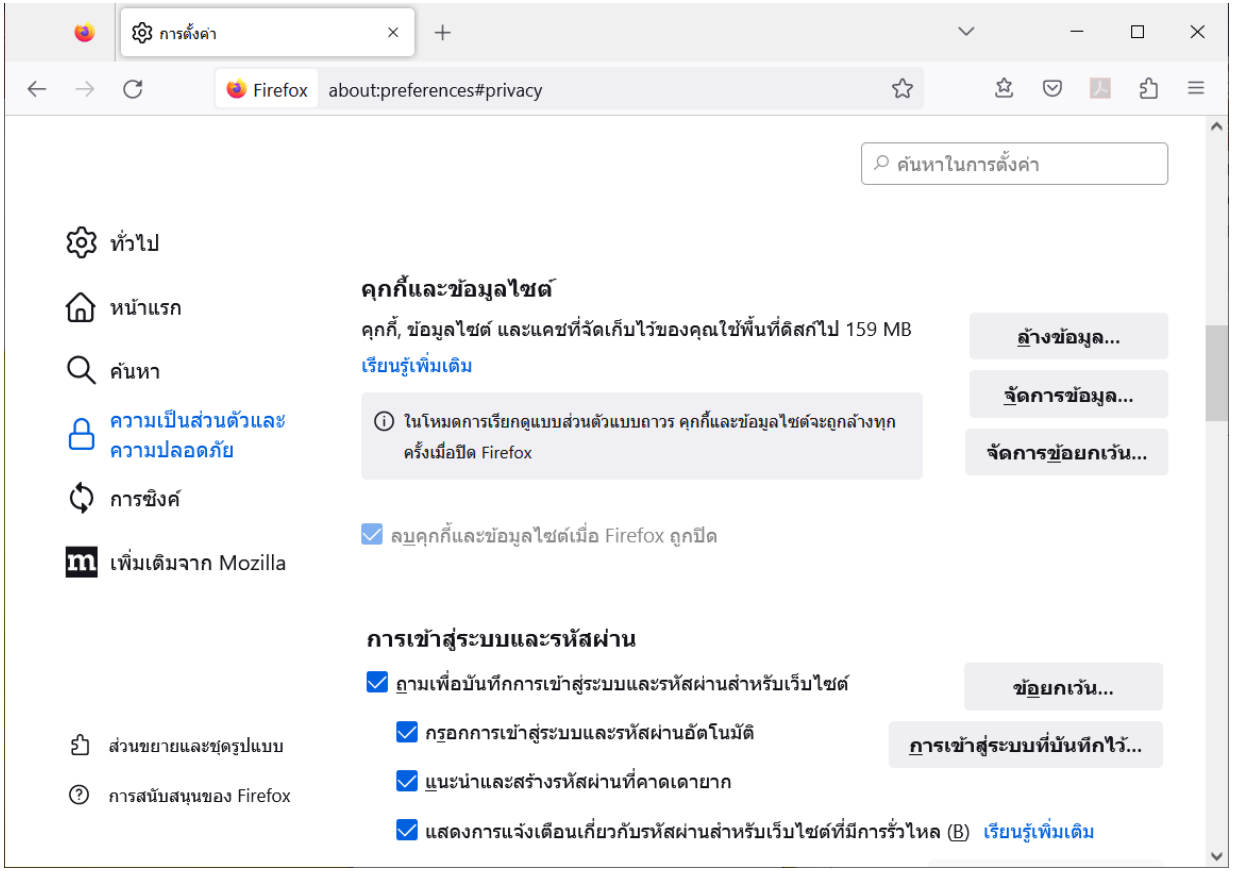
Servlet

- Enable Cookies
  - Chrome



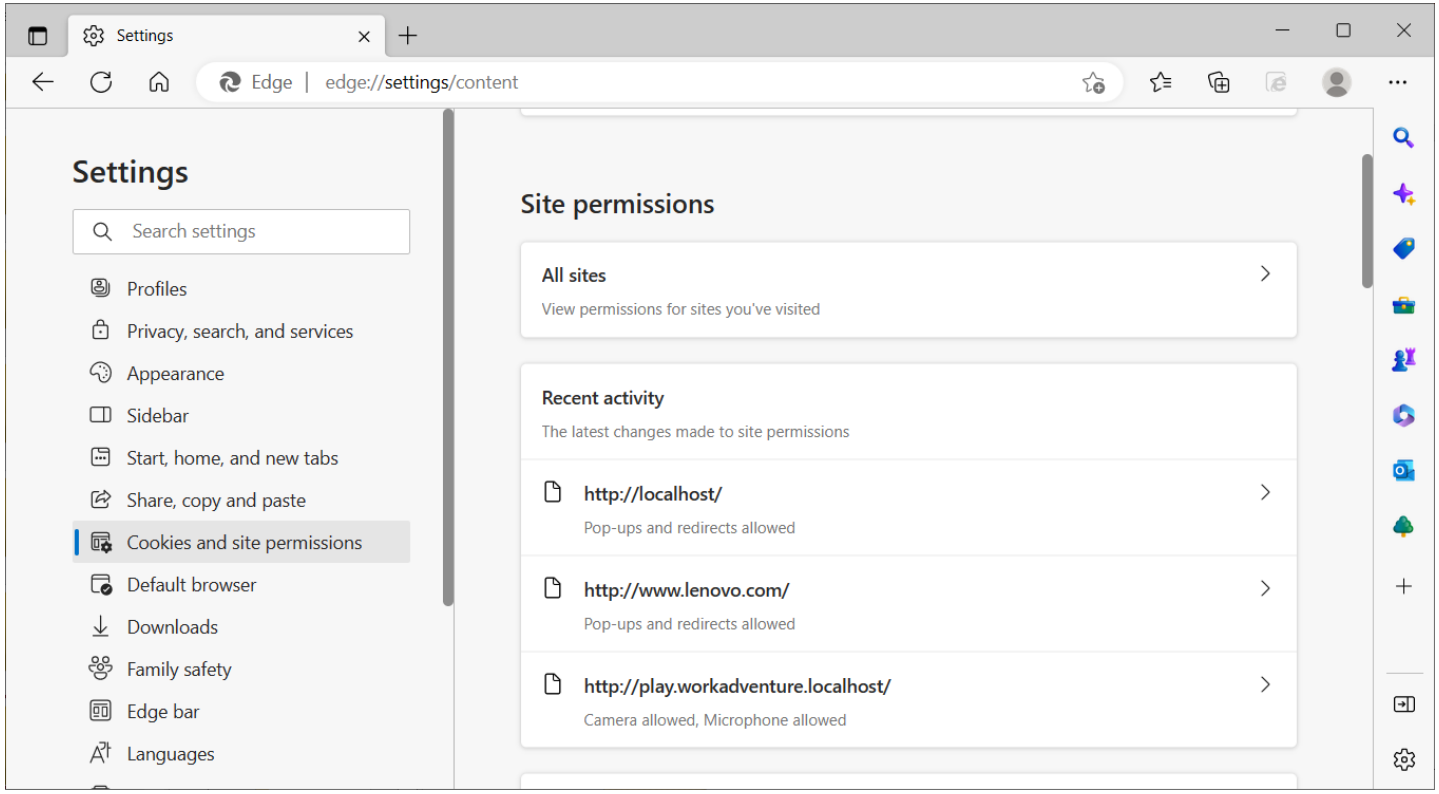
Servlet

- Enable Cookies
  - Firefox



Servlet

- Enable Cookies
  - Edge



Servlet

- Cookie Attributes
  - getComment/setComment
  - getDomain/setDomain
  - getMaxAge/setMaxAge
  - getName/setName
  - getPath/setPath
  - getSecure/setSecure
  - getValue/setValue
  - getVersion/setVersion

Servlet

- Cookies

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class CounterCookie extends HttpServlet {
                public void doGet(HttpServletRequest request, HttpServletResponse response)
                                throws ServletException, IOException {
                                PrintWriter out = response.getWriter();
                                out.println("Listing Cookies");
                                int counter = 0;
                                Cookie[] cookies = request.getCookies();
                                if(cookies!=null) {
                                                for(int i=0; i<cookies.length; i++) {
                                                                Cookie cookie = cookies[i];
                                                                out.println(cookie.getName()+" = "+cookie.getValue());
                                                                if("counter".equals(cookie.getName())) {
                                                                                if(cookie.getValue()!=null) {
                                                                                counter =
Integer.parseInt(cookie.getValue());
                                                                                }
                                                                }
                                                }
                                }
                                Cookie cookie = new Cookie("counter", ""+(++counter));
                                response.addCookie(cookie);
                }
}
```

Servlet

- Session Tracking
  - 3 Type Problem
    - Cookie
    - URL Rewriting
    - Hidden Form Field
  - Session Tracking API
    - HttpSession

Servlet

- Session Tracking

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class CounterSession extends HttpServlet {
        public void doGet(HttpServletRequest request,HttpServletResponse response)
                throws ServletException, IOException {
                HttpSession session = request.getSession(true);
                PrintWriter out = response.getWriter();
                Integer counter = new Integer(0);
                String heading = "";
                if (session.isNew()) {
                        heading = "Welcome, Newcomer";
                } else {
                        heading = "Welcome Back";
                        Integer oldCounter = (Integer)session.getAttribute("counter");
                        if(oldCounter!=null) {
                                counter = new Integer(oldCounter.intValue() + 1);
                        }
                }
                session.setAttribute("counter", counter);
                out.println(heading);
                out.println("Session ID = "+session.getId());
                out.println("Number of Access = "+counter);
        }
}
```

Servlet

- Servlet & JDBC - I

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class QueryServlet extends HttpServlet {
                public void doPost(HttpServletRequest request,HttpServletResponse response)
                                throws ServletException, IOException {
                                                doGet(request, response);
                }
                public void doGet(HttpServletRequest request, HttpServletResponse response)
                                throws ServletException, IOException {
                                PrintWriter out = response.getWriter();
                                String sql = request.getParameter("sql");
                                System.out.println("sql : "+sql);
                                if(sql!=null && !sql.trim().equals("")) {
                                                try {
                                                String driver = "com.mysql.jdbc.Driver";
                                                String url = "jdbc:mysql://127.0.0.1:3306/refdb";
                                                String user = "root";
                                                String password  = "root";
                                                Class.forName(driver);
```

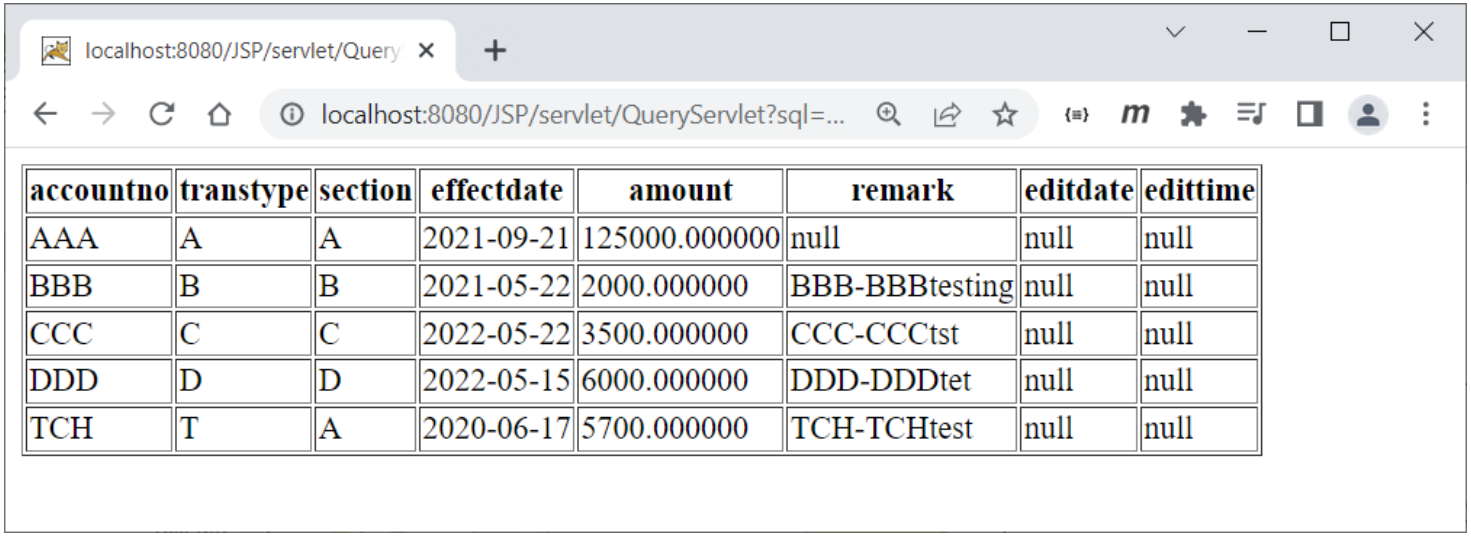
Servlet

- Servlet & JDBC - II

```
                                        java.sql.Connection conn =
java.sql.DriverManager.getConnection(url,user,password);
                                        java.sql.Statement stm = conn.createStatement();
                                        java.sql.ResultSet rs = stm.executeQuery(sql);
                                        java.sql.ResultSetMetaData met = rs.getMetaData();
                                        out.println("<table border=1>");
                                        out.println("<tr>");
                                        for(int i=1,isz=met.getColumnCount();i<=isz;i++) {
                                                String colname = met.getColumnName(i);
                                                out.println("<th>"+colname+"</th>");
                                        }
                                        out.println("</tr>");
                                        while(rs.next()) {
                                                out.println("<tr>");
                                                for(int i=1,isz=met.getColumnCount();i<=isz;i++) {
                                                        String colname = met.getColumnName(i);

        out.println("<td>"+rs.getString(colname)+"</td>");
                                                }
                                                out.println("</tr>");
                                        }
                                        out.println("</table>");
                                        } catch(Exception ex) { ex.printStackTrace(); }
                        }
        }
}
```

Servlet

- Servlet & JDBC
  - http://localhost:8080/JSP/servlet/QueryServlet?sql=select%20*%20from%20credit



accountno	transtype	section	effectdate	amount	remark	editdate	edittime
AAA	A	A	2021-09-21	125000.000000	null	null	null
BBB	B	B	2021-05-22	2000.000000	BBB-BBBtesting	null	null
CCC	C	C	2022-05-22	3500.000000	CCC-CCCtst	null	null
DDD	D	D	2022-05-15	6000.000000	DDD-DDDtet	null	null
TCH	T	A	2020-06-17	5700.000000	TCH-TCHtest	null	null

JSP

JSP

- What is JSP?
  - JSP stands for Java Server Page is a technology from Sun that enables the java programmers to generate HTML, XML or other types of documents to server the web client
  - Allow the programmers to embed Java code into html (.jsp) page

JSP

- Advantage of JSP
  - JSP translate and compile into java servlet but easier to develop
  - JSP uses simplified scripting language based syntax for embedding HTML
  - JSP containers provide easy way for accessing standard objects and actions
  - JSP use HTTP as default request / response communication paradigm

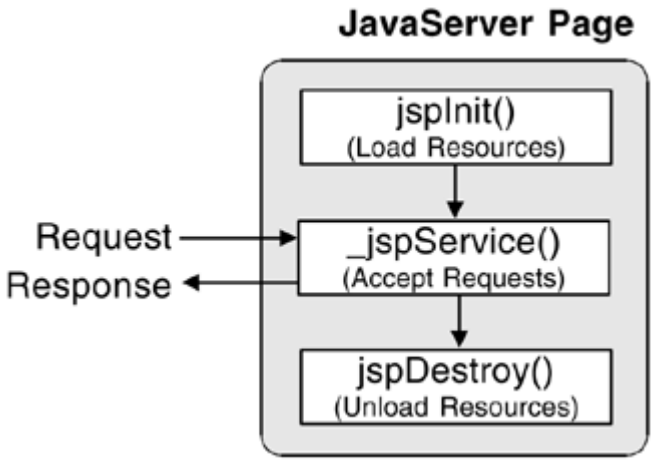
JSP

- hello.jsp

```
<%
            String hello = "Hello World";
%>
<html>
<head>
  <title>Hello JSP</title>
</head>
<body>
<%=hello%>
</body>
</html>
```

JSP

- JSP Lifecycle
  - jspInit
    - Invoked one time
  - jspService
    - Request and Response
  - jspDestroy
    - When shutdown



JSP

- JSP Lifecycle
  - cycle.jsp

```
<%!
public void jspInit() {
                System.out.println(getClass().getName()+" jsp init ...");
}
public void jspDestroy() {
                System.out.println(getClass().getName()+" jsp destroy ...");
}
%>
<%
                String hello = "Hello World";
                System.out.println(hello);
%>
<html>
<head>
  <title>Hello JSP</title>
</head>
<body>
<%=hello%>
</body>
</html>
```

JSP

- Java Server Page
  - Call Java code directly.
    - Place all Java code in JSP page. Appropriate only for very small amounts of code.
  - Call Java code indirectly.
    - Develop separate utility classes. Insert into JSP page only the Java code needed to invoke the utility classes.

JSP

- Java Server Page
  - Use beans
    - Develop separate utility classes structured as beans. Use jsp:useBean, jsp:getProperty, and jsp:setProperty to invoke the code.
  - Use the MVC architecture
    - Have a servlet respond to original request, look up data, and store results in beans. Forward to a JSP page to present results. JSP page uses beans.

JSP

- Predefined Variables
  - request
    - The HttpServletRequest (1st argument to service/doGet)
  - response
    - The HttpServletResponse (2nd arg to service/doGet)
  - out
    - The Writer (a buffered version of type JspWriter) used to the client

JSP

- Predefined Variables
  - session
    - The HttpSession associated with the request (unless disabled with the session attribute of the page directive)
  - application
    - The ServletContext (for sharing data) as obtained via getServletContext().

JSP

- Predefined Variables
  - config
    - This is the ServletConfig object for this page.
  - page
    - This is simply a synonym for this.
  - pageContext
    - A new class called PageContext to encapsulate use of server-specific features like higher performance JspWriter

JSP

- Predefined Variables
  - index.jsp

```
<%
		String hello = "Hello JSP";
		String who = request.getParameter("who");
		if(who!=null) hello = "Hello "+who;
%>
<html>
<head>
  <title>JSP Examples</title>
</head>
<body bgcolor="#FFFFFF">
		<% out.print(hello); %>
</body>
</html>
```

JSP

- Expressions
  - Format
    - <%= Java Expression %>
  - Result
    - Expression evaluated, converted to String, and placed into HTML page at the place it occurred in JSP page That is, expression placed in _jspService inside out.print
  - Examples
    - Current time: <%= new java.util.Date() %>
    - Your hostname: <%=request.getRemoteHost() %>

JSP

- Scriptlets
  - Format
    - <% Java Code %>
  - Result
    - Code is inserted verbatim into servlet's _jspService
  - Example
    - <%String queryData = request.getQueryString(); out.println("Attached GET data: " + queryData); %>
    - <% response.setContentType("text/plain"); %>

JSP

- Declarations
  - Format
    - <%! Java Code %>
  - Result
    - Code is inserted verbatim into servlet's class definition,outside of any existing methods
  - Examples
    - <%! private int someField = 5; %>
    - <%! private void someMethod(...) {...} %>

JSP

- Directives
  - <%@ page info="SCCS id: Id"%>
  - <%@ page errorPage="errorpage.jsp"%>
  - <%@ page isErrorPage="true"%>
  - <%@ page contentType="text/html; charset=windows-874"%>
  - <%@ page import="java.util.*"%>
  - <%@ page buffer="none"%>

JSP

- Directives
  - <%@ page autoFlush="true"%>
  - <%@ page isThreadSafe="true"%>
  - <%@ page session="false"%>
  - <%@ page extends="com.fs.bean.SuperJSP"%>
  - <%@ include file="includefile.jsp"%>

JSP

- Directives
  - errorpage.jsp

```
<%@ page isErrorPage="true" %>
<html>
<title>Error</title>
<head>
</head>
<body>
<br>
<center>
                <a href="javascript:window.history.back();">Go Back</a>
</center>
<br>
<div style="text-align:center;">
<%=exception%>
</div>
<br>
</body>
</html>
```

JSP

- Directives
  - error.jsp

```
<%@ page errorPage="errorpage.jsp"%>
<%
        String hello = null;
%>
<html>
<head>
  <title>Error JSP</title>
</head>
<body>
<%=hello.equals("Hello World")%>
</body>
</html>
```

JSP

- Directives Purpose
  - Give high-level information about the servlet that will result from the JSP page
  - Can control
    - Which classes are imported
    - What class the servlet extends
    - What MIME type is generated
    - How multithreading is handled
    - If the servlet participates in sessions
    - The size and behavior of the output buffer
    - What page handles unexpected error

JSP

- Standard Action Tag
  - <jsp:forward page="forward.jsp"/>
  - <jsp:include page="includefile.jsp"/>
  - <jsp:useBean id="fsUser" scope="session" class="com.fs.bean.UserBean"/>
  - <jsp:setProperty name="fsUser" property="*"/>

JSP

- Standard Action Tag
  - <jsp:useBean>
  - scope
    - page – bean can use within the jsp page
    - request – bean can use from any jsp page processing the same request.
    - session – bean can use from any jsp page in the same session.
    - application – bean can use from any jsp page in the same application

JSP

- Standard Action Tag
  - UserBean.java - I

```
package com.fs.bean;

public class  UserBean
{
		private String id = null;
		private String name = null;
		private String surname = null;
		public UserBean() {
		}
		public void setId(String id) {
			this.id = id;
		}
		public String getId() {
			return id;
		}
		public void setName(String name) {
			this.name = name;
		}
```

JSP

- Standard Action Tag
  - UserBean.java - II

```
	public String getName() {
		return name;
	}
	public void setSurname(String surname) {
		this.surname = surname;
	}
	public String getSurname() {
		return surname;
	}
	public String toString() {
		return super.toString()+"{"+id+","+name+","+surname+"}";
	}
}
```

JSP

- Standard Action Tag
  - user.jsp

```
<jsp:useBean id="fsUser" scope="session" class="com.fs.bean.UserBean"/>
<jsp:setProperty name="fsUser" property="*"/>
<%
		System.out.println(fsUser);
%>
<html>
<head>
  <title>JSP Examples</title>
</head>
<body bgcolor="#FFFFFF">
<form name="fsform" action="user.jsp" method="post">
		<table>
			<tr><td>ID</td><td><input name="id" value=""></input></td></tr>
			<tr><td>Name</td><td><input name="name" value=""></input></td></tr>
			<tr><td>Surname</td><td><input name="surname"
value=""></input></td></tr>
			<tr><td><input type="submit" value="submit"></input></td><td><input
type="reset" value="reset"></input></td></tr>
		</table>
<form>
</body>
</html>
```

JSP

- Standard Action Tag
  - userbean.jsp

```
<jsp:useBean id="fsUser" scope="session" class="com.fs.bean.UserBean"/>
<jsp:setProperty name="fsUser" property="*"/>
<%
		System.out.println(fsUser);
%>
<html>
<head>
  <title>JSP Examples</title>
</head>
<body bgcolor="#FFFFFF">
		<table>
			<tr><td>ID</td><td><jsp:getProperty name="fsUser" property="id"/></td></tr>
			<tr><td>Name</td><td><jsp:getProperty name="fsUser"
property="name"/></td></tr>
			<tr><td>Surname</td><td><jsp:getProperty name="fsUser"
property="surname"/></td></tr>
		</table>
</body>
</html>
```

JSP

- Standard Action Tag
  - usersession.jsp

```
<%
		com.fs.bean.UserBean fsUser = (com.fs.bean.UserBean)session.getAttribute("fsUser");
%>
<html>
<head>
  <title>JSP Examples</title>
</head>
<body bgcolor="#FFFFFF">
		<table>
			<tr><td>ID</td><td><%=fsUser.getId()%></td></tr>
			<tr><td>Name</td><td><%=fsUser.getName()%></td></tr>
			<tr><td>Surname</td><td><%=fsUser.getSurname()%></td></tr>
		</table>
</body>
</html>
```

JSP

- Custom Tags
  - Tag Handler Class
    - Java code to output
  - Tag Library Descriptor File
    - XML file describing tag name, attributes and tag handler class (TLD file)
  - JSP File
    - Import tag library
    - Define tag prefix
    - Use tag

JSP

- Custom Tags
  - Tag Handler Class (Select.java - I)

```
package com.fs.tag;
import java.io.*;
import java.util.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;

public class Select extends BodyTagSupport {
		private String name = null;
		private String section = null;
public Select() {
		super();
}
public String getName() {
		return name;
}
public String getSection() {
		return section;
}
```

JSP

- Custom Tags
  - Tag Handler Class (Select.java - II)

```
public void setName(String newName) {
        name = newName;
}
public void setSection(String section) {
        this.section = section;
}
public int doAfterBody() throws JspException {
        try {
                BodyContent body = getBodyContent();
                JspWriter out = body.getEnclosingWriter();
                StringBuffer outstr = new StringBuffer();
                outstr.append("<select name='");
                outstr.append(name);
                outstr.append("' ");
                outstr.append(">");
```

JSP

- Custom Tags
  - Tag Handler Class (Select.java - III)

```
BufferedReader contentReader = new
BufferedReader(body.getReader());
String rowread = "";
StringBuffer item = new StringBuffer();
while ((rowread = contentReader.readLine()) != null) {
        item.append(rowread.trim());
}
java.util.Map tree = null;
if((getSection()!=null) && !getSection().equals("")) {
        tree =
(java.util.Map)this.pageContext.getSession().getAttribute(getSection());
}
if((tree!=null) && !tree.isEmpty()) {
        java.util.Iterator it = tree.keySet().iterator();
        for(;it.hasNext();) {
                Object key = it.next();
                String value = (String)tree.get(key);
                if((key!=null) && (value!=null)) {
```

JSP

- Custom Tags
  - Tag Handler Class (Select.java - IV)

```
                        if(item.toString().equals(key.toString())) {
                        outstr.append("<option value=\""+key+"\"
selected>"+value+"</option>\n");
                        } else {
                        outstr.append("<option value=\""+key+"\">"+value+"</option>\n");
                                                                }
                                                        }
                                                }
                        }
                        outstr.append("</select>");
                        if(outstr!=null){
                                        out.println(outstr.toString());
                        }
                } catch (IOException e) {
                        throw new JspTagException(e.toString());
                }
                return SKIP_BODY;
} }
```

JSP

- Custom Tags
  - Tag Library Descriptor File – taglibs.tld - I

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE taglib PUBLIC "-//Sun Microsystems, Inc.//DTD JSP Tag Library 1.2//EN"
"http://java.sun.com/dtd/web-jsptaglibrary_1_2.dtd">
<taglib>

<tlib-version>1.0</tlib-version>
<jsp-version>1.2</jsp-version>
<short-name>input</short-name>
<uri>http://freewillsolutions.com/RD/taglibs/formcontrol</uri>
<description>Set of JSP external Tag from Freewill Solutions. Use for control a entry
form.</description>
```

JSP

- Custom Tags
  - Tag Library Descriptor File – taglibs.tld - II

```
<tag>
        <name>select</name>
        <tag-class>com.fs.tag.Select</tag-class>
        <body-content>JSP</body-content>
                <attribute>
                        <name>name</name>
                        <required>true</required>
                        <rtexprvalue>true</rtexprvalue>
                </attribute>
                <attribute>
                        <name>section</name>
                        <required>false</required>
                </attribute>
</tag>

</taglib>
```

JSP

- Custom Tags
  - tag.jsp - 1

```
<%@ taglib uri="/WEB-INF/taglibs.tld" prefix="fs"%>
<%
		java.util.Map map = new java.util.TreeMap();
		map.put("01","Male");
		map.put("02","Female");
		map.put("03","Other");
		session.setAttribute("GENDER",map);
%>
<html>
<meta http-equiv="content-type" content="text/html; charset=windows-874">
<title>Test TLD</title>
<head>
</head>
<body>
<fs:select name="gender" section="GENDER">03</fs:select>
```

JSP

- Custom Tags
  - tag.jsp - II

```
<select name="gender1">
<%
		java.util.Iterator it = map.keySet().iterator();
		for(;it.hasNext();) {
			String key = (String)it.next();
			String value = (String)map.get(key);
			%>
				<option value="<%=key%>"><%=value%></option>
			<%
		}
%>
</select>
</body>
</html>
```

JSP

- Custom Tags
  - Simple Tag File – tomcat 5 or later
    - helloWorld.jsp

```
<%@ taglib prefix="tags" tagdir="/WEB-INF/tags" %>
<html>
  <head>
    <title>Hello World Using Tag File</title>
  </head>
  <body>
    <tags:helloWorld/>
  </body>
</html>
```

    - helloWorld.tag

```
<font color=red>Hello, world!</font>
```

JSP

- JSP & JDBC
  - jdbc.jsp - 1

```
<%@ page errorPage="errorpage.jsp"%>
<%@ page contentType="text/html; charset=windows-874"%>
<%@ page import="com.fs.bean.util.*"%>
<html>
<head>
   <title>Query</title>
</head>
<body>
<form name="qform" action="jdbc.jsp" method="post">
                <table>
                                <tr><td>SQL</td><td><input name="sql"
size="30"></input></td></tr>
                                <tr><td></td><td><input type=submit name="submit"
value="Execute"></input></tr>
                </table>
</form>
<table border=1>
```

JSP

- JSP & JDBC
  - jdbc.jsp - II

```
<%
		String sql = request.getParameter("sql");
		System.out.println("sql : "+sql);
		if(sql!=null && !sql.trim().equals("")) {
			String driver = "com.mysql.jdbc.Driver";
			String url = "jdbc:mysql://127.0.0.1:3306/refdb";
			String user = "root";
			String password  = "root";
			Class.forName(driver);
			java.sql.Connection conn =
java.sql.DriverManager.getConnection(url,user,password);
			java.sql.Statement stm = conn.createStatement();
			java.sql.ResultSet rs = stm.executeQuery(sql);
			java.sql.ResultSetMetaData met = rs.getMetaData();
			out.println("<tr>");
			for(int i=1,isz=met.getColumnCount();i<=isz;i++) {
				String colname = met.getColumnName(i);
				out.println("<th>"+colname+"</th>");
			}
```

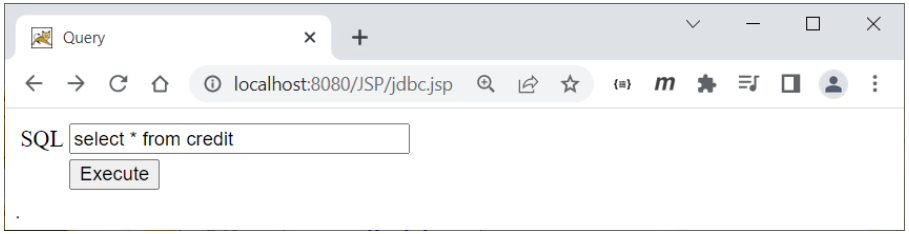
JSP

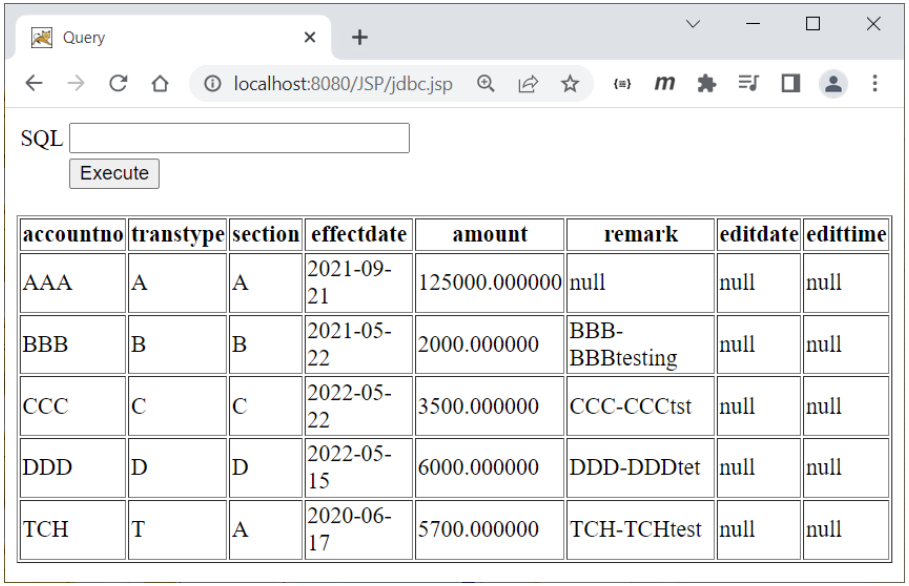
- JSP & JDBC
  - jdbc.jsp - III

```
                        out.println("</tr>");
                        while(rs.next()) {
                                out.println("<tr>");
                                for(int i=1,isz=met.getColumnCount();i<=isz;i++) {
                                        String colname = met.getColumnName(i);

        out.println("<td>"+rs.getString(colname)+"</td>");
                                                                }
                                out.println("</tr>");
                        }
        }
%>
</table>
</body>
</html>
```

JSP

- JSP & JDBC
  - http://localhost:8080/JSP/jdbc.jsp





accountno	transtype	section	effectdate	amount	remark	editdate	edittime
AAA	A	A	2021-09-21	125000.000000	null	null	null
BBB	B	B	2021-05-22	2000.000000	BBB-BBBtesting	null	null
CCC	C	C	2022-05-22	3500.000000	CCC-CCCtst	null	null
DDD	D	D	2022-05-15	6000.000000	DDD-DDDtet	null	null
TCH	T	A	2020-06-17	5700.000000	TCH-TCHtest	null	null

Reference

- https://www.javatpoint.com/servlet-tutorial
- https://www.geeksforgeeks.org/introduction-java-servlets/
- https://www.baeldung.com/intro-to-servlets
- https://www.tutorialspoint.com/jsp/index.htm
- https://www.javatpoint.com/jsp-tutorial
- https://www.geeksforgeeks.org/introduction-to-jsp/

Q & A