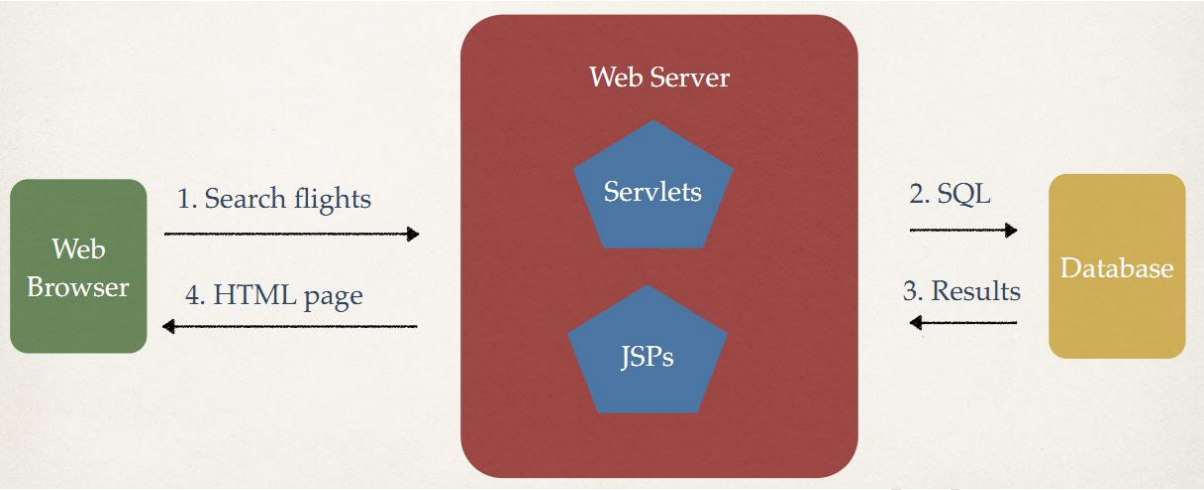
Contents

Chapter 1. JSP and Servlet Overview

1.1. JSP and Servlet overview



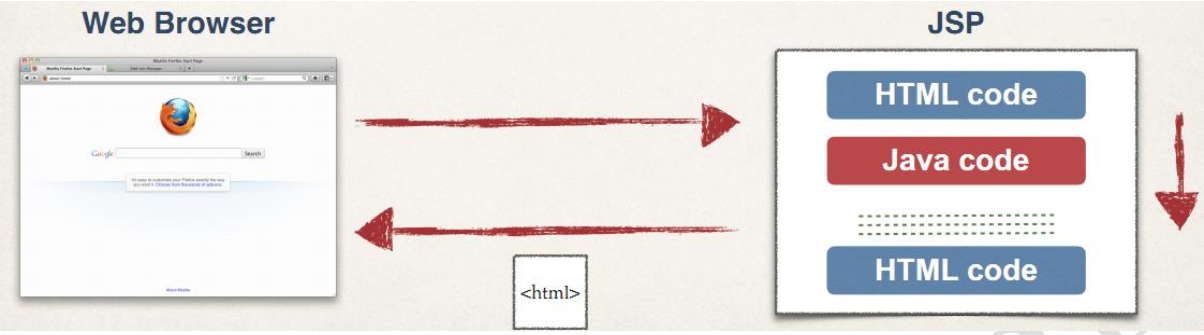
1.2. Setup Environment

1. Java Application Server - **Tomcat**

2. Java Integrated Development Environment (IDE) - **Eclipse**

Chapter 2. JSP Fundamentals

2.1. JSP overview



JSP Helloworld

helloworld.jsp

```
<html>

<body>

<h3>Hello World of Java!</h3>

The time on the server is <%= new java.util.Date() %>

</body>

</html>
```

2.2. JSP Expression

Element	Syntax
JSP Expression	<%= some Java expression %>
JSP Scriptlet	<% some Java code: 1 to many lines %>
JSP Declaration	<%! variable or method declaration %>

expression-test.jsp

```
<html>
```

```
<body>

Converting a string to uppercase: <%= new String("Hello World").toUpperCase() %>

<br/><br/>

25 multiplied by 4 equals <%= 25*4 %>

<br/><br/>

Is 75 less than 69? <%= 75 < 69 %>

</body>

</html>
```

2.3. JSP Scriplets

scriptlet-test.jsp

```
<html>

<body>

<h3>Hello World of Java</h3>

<%
	for (int i=1; i <=5; i++) {
		out.println("<br/>I really luv2code: " + i);
	}
%>
</body>

</html>
```

2.4. JSP Declarations

declaration-test.jsp

```
<html>

<body>
<%!
	String makeItLower(String data) {
		return data.toLowerCase();
	}
%>

Lower case "Hello World": <%= makeItLower("Hello World") %>

</body>
```

```
</html>
```

2.5. Calling a Java Class from JSP

FunUtils.java

```
package com.luv2code.jsp;

public class FunUtils {

	public static String makeItLower(String data) {
		return data.toLowerCase();
	}

}
```

fun-test.jsp

```
<%@ page import="com.luv2code.jsp.*" %>
<html>

<body>

Let's have some fun: <%= FunUtils.makeItLower("FUN FUN FUN") %>

</body>

</html>
```

2.6. JSP Built-in Objects

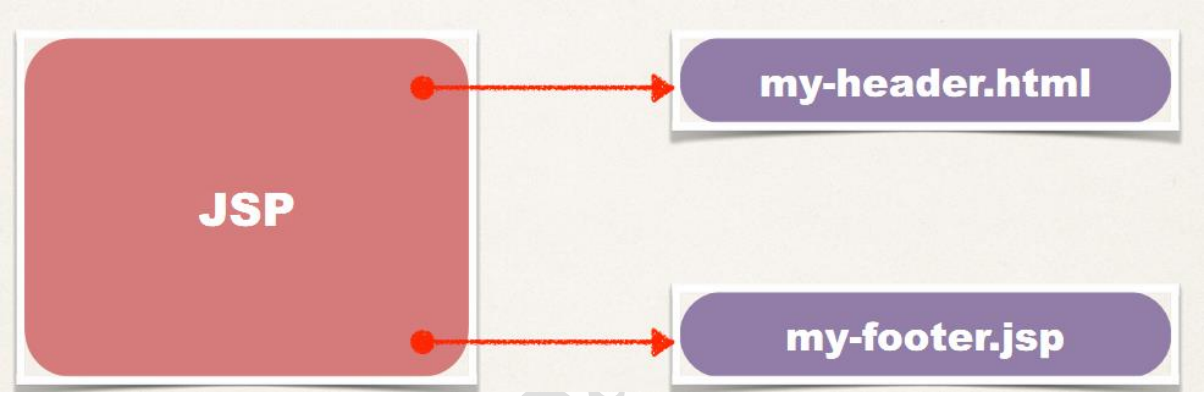
Object	Description
request	Contains HTTP request headers and form data
response	Provides HTTP support for sending response
out	JspWriter for including content in HTML page
session	Unique session for each user of the web application
application	Shared data for all users of the web application

builtin-test.jsp

```
<html>
```

```
<body>

<h3>JSP Built-In Objects</h3>

Request user agent: <%= request.getHeader("User-Agent") %>

<br/><br/>

Request language: <%= request.getLocale() %>

</body>

</html>
```

2.7. Including Files in JSP



my-header.html

```
<h1 align="center">JSP Tutorial</h1>
```

my-footer.jsp

```
<p align="center">
Last updated: <%= new java.util.Date() %>
</p>
```

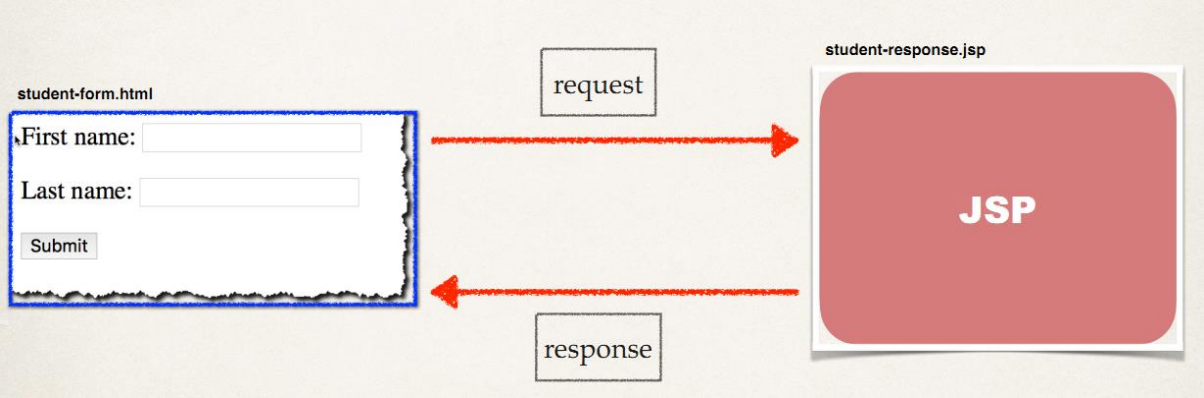
homepage.jsp

```
html>

<body>

<jsp:include page="my-header.html" />

Blah blah blah .... <br/> <br/>
Blah blah blah .... <br/> <br/>
```

```
Blah blah blah .... <br/> <br/>

<jsp:include page="my-footer.jsp" />

</body>

</html>
```

2.8. Reading HTML Form Data from JSP

Overview



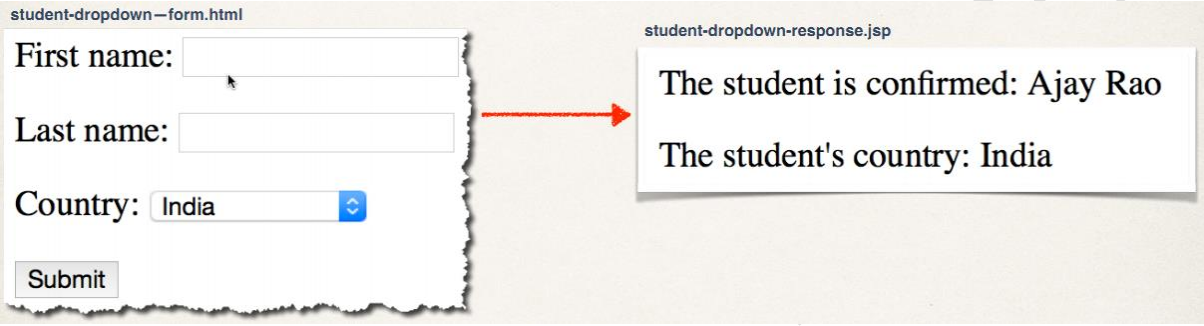
student-form.html

```
<html>

<head><title>Student Registration Form</title></head>

<body>

<form action="student-response.jsp">

        First name: <input type="text" name="firstName" />

        <br/><br/>

        Last name: <input type="text" name="lastName" />

        <br/><br/>

        <input type="submit" value="Submit" />

</form>

</body>
</html>
```

student-response.jsp

```
<html>

<head><title>Student Confirmation Title</title></head>

<body>

	The student is confirmed: ${param.firstName} ${param.lastName}

</body>
</html>
```

Drop-Down Lists



student-dropdown-form.html

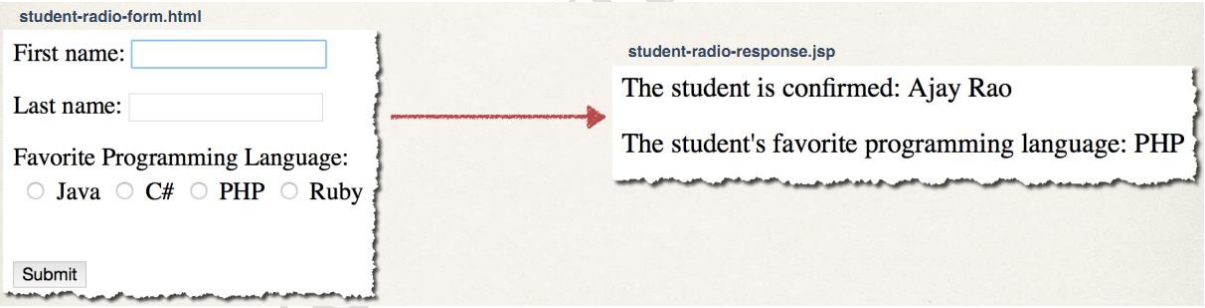
```
<html>

<head><title>Student Registration Form</title></head>

<body>

<form action="student-dropdown-response.jsp">

	First name: <input type="text" name="firstName" />

	<br/><br/>

	Last name: <input type="text" name="lastName" />

	<br/><br/>

	<select name="country">
		<option>Brazil</option>
		<option>France</option>
		<option>Germany</option>
		<option>India</option>
		<option>Turkey</option>
		<option>United Kingdom</option>
		<option>United States of America</option>
	</select>

	<br/><br/>
```

```
        <input type="submit" value="Submit" />

</form>

</body>
</html>
```

student-dropdown-response.jsp

```
<html>

<head><title>Student Confirmation Title</title></head>

<body>

        The student is confirmed: ${param.firstName} ${param.lastName}

        <br/><br/>

        The student's country: ${param.country}
</body>
</html>
```

Radio Buttons



student-radio-form.html

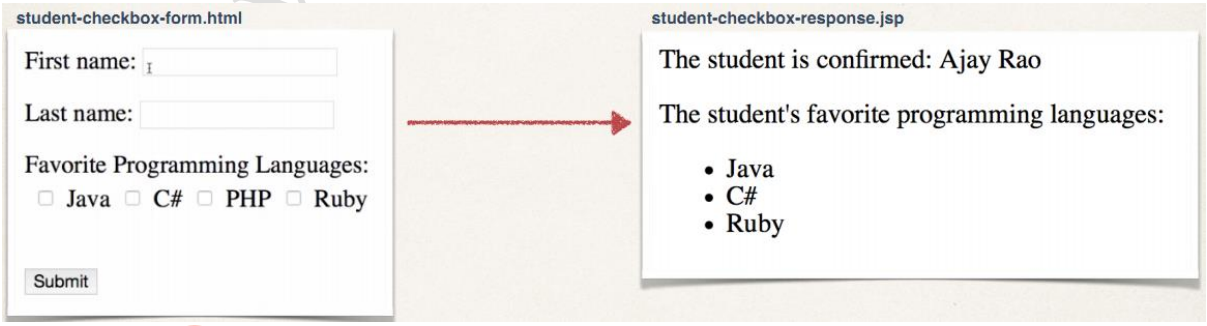
```
<html>

<head><title>Student Registration Form</title></head>

<body>

<form action="student-radio-response.jsp">

        First name: <input type="text" name="firstName" />

        <br/><br/>

        Last name: <input type="text" name="lastName" />
```

```
        <br/><br/>

        Favorite Programming Language: <br/>

        <input type="radio" name="favoriteLanguage" value="Java"> Java

        <input type="radio" name="favoriteLanguage" value="C#"> C#

        <input type="radio" name="favoriteLanguage" value="PHP"> PHP

        <input type="radio" name="favoriteLanguage" value="Ruby"> Ruby

        <br/><br/>

        <input type="submit" value="Submit" />

</form>

</body>
</html>
```

student-radio-response.jsp

```
<html>

<head><title>Student Confirmation Title</title></head>

<body>

        The student is confirmed: ${param.firstName} ${param.lastName}

        <br/> <br/>

        The student's favorite programming language: ${param.favoriteLanguage}

</body>
</html>
```

Checkboxes



student-checkbox-form.html

```
<html>

<head><title>Student Registration Form</title></head>

<body>
```

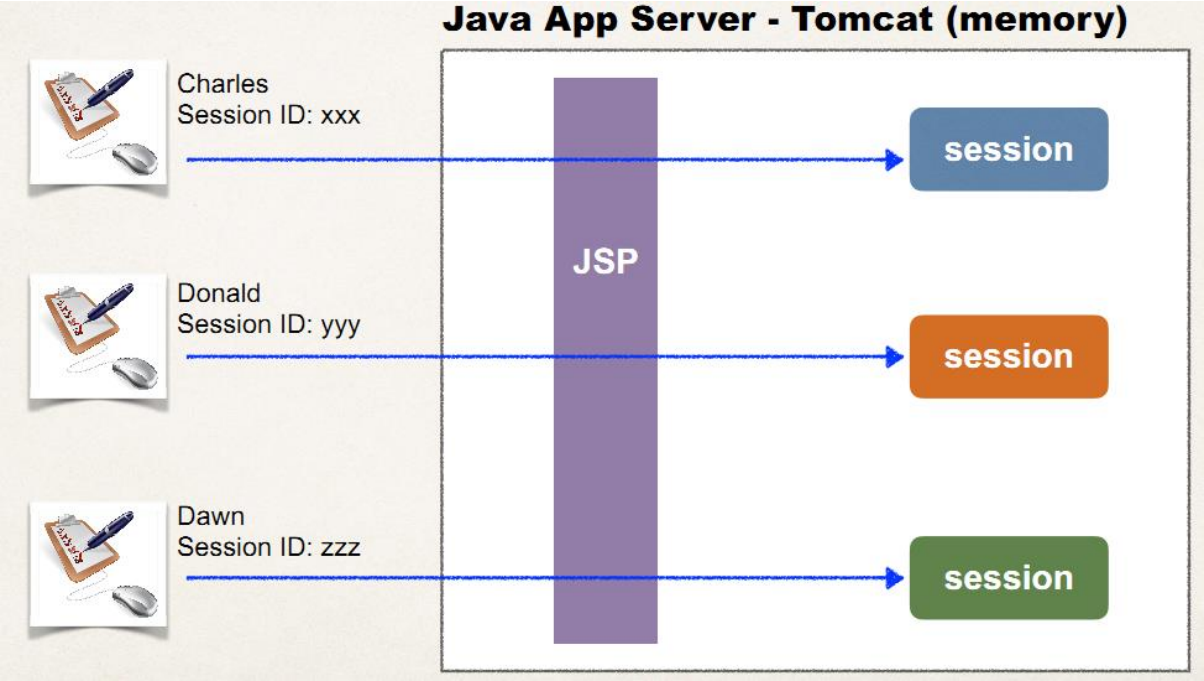
```
<form action="student-checkbox-response.jsp">

        First name: <input type="text" name="firstName" />

        <br/><br/>

        Last name: <input type="text" name="lastName" />

        <br/><br/>

        <input type="checkbox" name="favoriteLanguage" value="Java"> Java

        <input type="checkbox" name="favoriteLanguage" value="C#"> C#

        <input type="checkbox" name="favoriteLanguage" value="PHP"> PHP

        <input type="checkbox" name="favoriteLanguage" value="Ruby"> Ruby

        <br/><br/>

        <input type="submit" value="Submit" />

</form>

</body>
</html>
```

student-checkbox-response.jsp

```
<html>

<head><title>Student Confirmation Title</title></head>

<body>

        The student is confirmed: ${param.firstName} ${param.lastName}

        <br/><br/>

        Favorite Programming Languages: <br/>

        <!-- display list of "favoriteLanguage" -->
        <ul>
                <%
                        String[] langs =
request.getParameterValues("favoriteLanguage");

                        for (String tempLang : langs) {
                                out.println("<li>" + tempLang + "</li>");
                        }
                %>
        </ul>
</body>
</html>
```

Chapter 3. Session and Cookies

3.1. Session



JSP Session methods

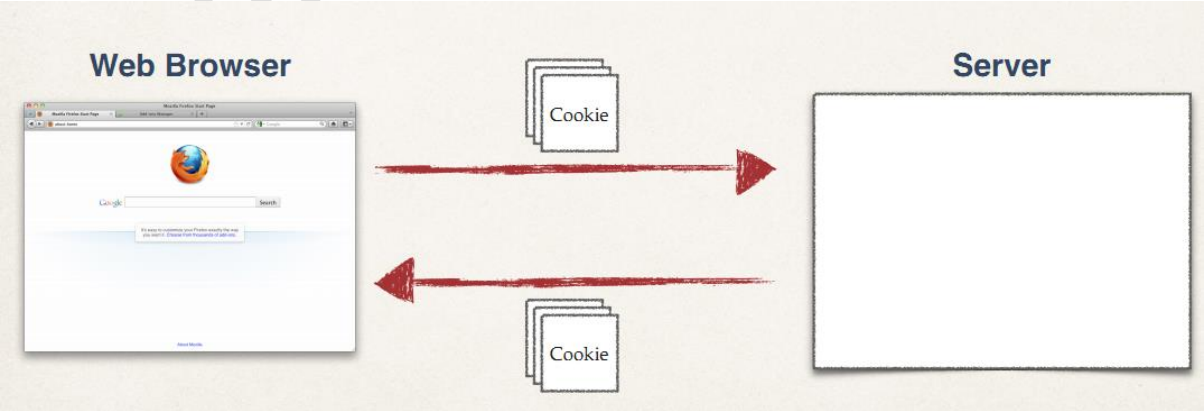
Method	Description
isNew() : boolean	Returns true if the session is new
getId() : String	Returns the session id
invalidate() : void	Invalidates this session and unbinds any object associated with it
setMaxInactiveInterval(long mills) : void	Sets the idle time for a session to expire. The value is supplied in milliseconds

todo-demo.jsp

```
<%@ page import="java.util.*" %>

<html>

<body>

<!-- Step 1: Create HTML form -->
```

```
<form action="todo-demo.jsp">
	Add new item: <input type="text" name="theItem" />

	<input type="submit" value="Submit" />
</form>

<!-- Step 2: Add new item to "To Do" list -->
<%
	// get the TO DO items from the session
	List<String> items = (List<String>) session.getAttribute("myToDoList");

	// if the TO DO items doesn't exist, then create a new one
	if (items == null) {
		items = new ArrayList<String>();
		session.setAttribute("myToDoList", items);
	}

	// see if there is form data to add
	String theItem = request.getParameter("theItem");
	if (theItem != null) {
		items.add(theItem);
	}
%>

<!-- Step 3: Display all "To Do" item from session -->
<hr>
<b>To List Items:</b> <br/>

<ol>
<%
	for (String temp : items) {
		out.println("<li>" + temp + "</li>");
	}
%>
</ol>

</body>

</html>
```

3.2. Cookies



cookies-personalize-form.html

```
<html>
<head>
	<title>Personalize The Site</title>
</head>

<body>

<form action="cookies-personalize-response.jsp">
	Select your Favorite Programming Language
	<select name="favoriteLanguage">
			<option>Java</option>
			<option>C#</option>
			<option>PHP</option>
			<option>Ruby</option>
	</select>

	<br/><br/>

	<input type="submit" value="Submit" />
</form>


</body>
```

cookies-personalize-response.jsp

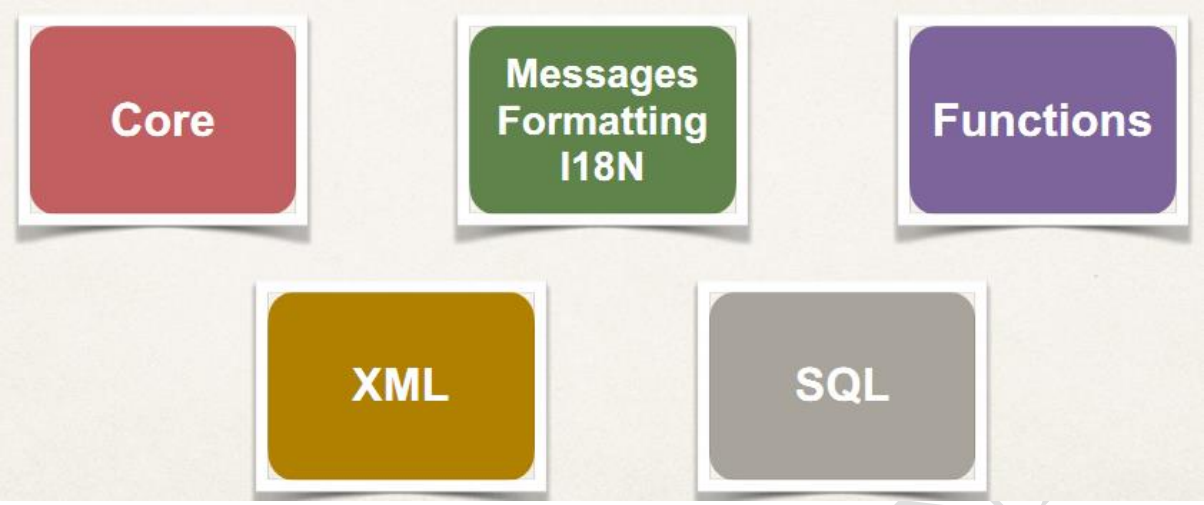
```
<html>

<head><title>Confirmation</title></head>

<%
	// read form data
	String favLang = request.getParameter("favoriteLanguage");

	// create the cookie
	Cookie theCookie = new Cookie("myApp.favoriteLanguage", favLang);

	// set the life span ... total number of seconds (yuk!)
	theCookie.setMaxAge(60*60*24*365);	// <-- for one year

	// send cookie to browser
	response.addCookie(theCookie);
%>
<body>

	Thanks! We set your favorite language to: ${param.favoriteLanguage}

	<br/><br/>

	<a href="cookies-homepage.jsp">Return to homepage.</a>

</body>
```

```
</html>
```

cookies-homepage.jsp

```
<html>

<body>
<h3>Training Portal</h3>

<!-- read the favorite programming language cookie -->
<%
	// the default ... if there are no cookies
	String favLang = "Java";

	// get the cookies from the browser request
	Cookie[] theCookies = request.getCookies();

	// find our favorite language cookie
	if (theCookies != null) {

		for (Cookie tempCookie : theCookies) {

			if ("myApp.favoriteLanguage".equals(tempCookie.getName())) {
				favLang = tempCookie.getValue();
				break;
			}
		}
	}
%>

<!-- now show a personalized page ... use the "favLang" variable -->

<!-- show new books for this lang -->
<h4>New Books for <%= favLang %></h4>
<ul>
	<li>blah blah blah</li>
	<li>blah blah blah</li>
</ul>

<h4>Latest News Reports for <%= favLang %></h4>
<ul>
	<li>blah blah blah</li>
	<li>blah blah blah</li>
</ul>

<h4>Hot Jobs for <%= favLang %></h4>
<ul>
	<li>blah blah blah</li>
	<li>blah blah blah</li>
</ul>

<hr>
<a href="cookies-personalize-form.html">Personalize this page</a>
</body>

</html>
```

Chapter 4. JSP Standard Tag Library (JSTL)



4.1. Install JSTL

```
1. In Eclipse, select File > New > Dynamic Web Project

2. For project name, enter: tagdemo

3. Keep all other defaults and click Finish.

Next, you need to install the JSTL JAR files

4. Visit http://www.luv2code.com/downloadjstl

5. Save the zip file to your computer

6. Unzip the file

7. Copy the two JAR files
- javax.servlet.jsp.jstl-1.2.1.jar
- javax.servlet.jsp.jstl-api-1.2.1.jar

8. In Eclipse, paste the two JAR files to your tagdemo project directory:
WebContent/WEB-INF/lib
```

test.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<html>

<body>

<c:set var="stuff" value="<%= new java.util.Date() %>" />
```

```
Time on the server is ${stuff}

</body>

</html>
```

4.2. JSTL Core Tags

ForEach

foreach-simple-test.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%
	// just create some sample data ... normally provided by MVC
	String[] cities = {"Mumbai", "Singapore", "Philadelphia"};

	pageContext.setAttribute("myCities", cities);
%>

<html>
<body>
	<c:forEach var="tempCity" items="${myCities}">

		${tempCity} <br/>

	</c:forEach>
</body>
</html>
```

```
JSTL Core Tags - ForEach - Building HTML Table
```

Student.java

```
package com.luv2code.jsp.tagdemo;

public class Student {

	private String firstName;
	private String lastName;
	private boolean goldCustomer;

	public Student(String firstName, String lastName, boolean goldCustomer) {
		super();
		this.firstName = firstName;
		this.lastName = lastName;
		this.goldCustomer = goldCustomer;
	}
```

```
	public String getFirstName() {
		return firstName;
	}

	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}

	public boolean isGoldCustomer() {
		return goldCustomer;
	}

	public void setGoldCustomer(boolean goldCustomer) {
		this.goldCustomer = goldCustomer;
	}



}
```

foreach-student-test.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ page import="java.util.*,com.luv2code.jsp.tagdemo.Student" %>

<%
	// just create some sample data ... normally provided by MVC
	List<Student> data = new ArrayList<>();

	data.add(new Student("John", "Doe", false));
	data.add(new Student("Maxwell", "Johnson", false));
	data.add(new Student("Mary", "Public", true));

	pageContext.setAttribute("myStudents", data);
%>

<html>

<body>
	<table border="1">

	<tr>
		<th>First Name</th>
		<th>Last Name</th>
		<th>Gold Customer</th>
	</tr>
```

```
        <c:forEach var="tempStudent" items="${myStudents}">

                <tr>
                        <td>${tempStudent.firstName}</td>
                        <td>${tempStudent.lastName}</td>
                        <td>${tempStudent.goldCustomer}</td>
                </tr>

        </c:forEach>

        </table>

</body>

</html>
```

If

Student.java

if-student-test.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ page import="java.util.*,com.luv2code.jsp.tagdemo.Student" %>

<%
        // just create some sample data ... normally provided by MVC
        List<Student> data = new ArrayList<>();

        data.add(new Student("John", "Doe", false));
        data.add(new Student("Maxwell", "Johnson", false));
        data.add(new Student("Mary", "Public", true));

        pageContext.setAttribute("myStudents", data);
%>

<html>

<body>
        <table border="1">

        <tr>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Gold Customer</th>
        </tr>

        <c:forEach var="tempStudent" items="${myStudents}">

                <tr>
                        <td>${tempStudent.firstName}</td>
                        <td>${tempStudent.lastName}</td>
```

```
                <td>
                        <c:if test="${tempStudent.goldCustomer}">
                                Special Discount
                        </c:if>

                        <c:if test="${not tempStudent.goldCustomer}">
                                -
                        </c:if>

                </td>
            </tr>

        </c:forEach>

        </table>

</body>

</html>
```

Choose

Student.java

choose-student-test.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ page import="java.util.*,com.luv2code.jsp.tagdemo.Student" %>

<%
        // just create some sample data ... normally provided by MVC
        List<Student> data = new ArrayList<>();

        data.add(new Student("John", "Doe", false));
        data.add(new Student("Maxwell", "Johnson", false));
        data.add(new Student("Mary", "Public", true));

        pageContext.setAttribute("myStudents", data);
%>

<html>

<body>
        <table border="1">

        <tr>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Gold Customer</th>
        </tr>
```

```
        <c:forEach var="tempStudent" items="${myStudents}">

                <tr>
                        <td>${tempStudent.firstName}</td>
                        <td>${tempStudent.lastName}</td>
                        <td>
                                <c:choose>

                                        <c:when test="${tempStudent.goldCustomer}">
                                                Special Discount
                                        </c:when>

                                        <c:otherwise>
                                                no soup for you!
                                        </c:otherwise>

                                </c:choose>


                        </td>
                </tr>

        </c:forEach>

        </table>

</body>

</html>
```

4.3. JSTL Function Tags

Length - toUpperCase - startsWith

function-test.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>

<html>

<body>

<c:set var="data" value="luv2code" />

Length of the string <b>${data}</b>: ${fn:length(data)}

<br/><br/>

Uppercase version of the string <b>${data}</b>: ${fn:toUpperCase(data)}

<br/><br/>
```

```
Does the string <b>${data}</b> start with <b>luv</b>?: ${fn:startsWith(data,
"luv")}

</body>

</html>
```

Split and join

split-join-test.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>

<html>
<body>

<c:set var="data" value="Singapore,Toyko,Mumbai,London" />

<h3>Split Demo</h3>

<c:set var="citiesArray" value="${fn:split(data, ',')}" />

<c:forEach var="tempCity" items="${citiesArray}" >
      ${tempCity} <br/>
</c:forEach>

<h3>Join Demo</h3>

<c:set var="fun" value="${fn:join(citiesArray, '*')}" />

Result of joining: ${fun}


</body>
</html>
```

4.4. JSTL Formatting Tags

Multi - Lingual Application

Formatting Messages

labels / placeholders

[greet]

[first_name]: John
[last_name]: Doe

[welcome_message]

English (US) | Spanish (ES) | German (DE)

Hola

Nombre de pila *John*
Apellido *Doe*

Bienvenidos a la clase de formación.

Selected locale: es_ES

To do list

- Step 1: Create Resource Files
  - Translated versions of your labels
- Step 2: Create JSP Page with labels
- Step 3: Update JSP page to change locale based on user selection

Step 1

- File name must follow specific format
  - **<your project file name>_LANGUAGECODE_COUNTRYCODE.properties**
- Examples:
  - mylabels_es_ES.properties
  - mylabels_de_DE.properties
  - mylabels_en_GB.properties

- Here's an example for the locale: **Spanish - Spain**

File: mylabels_es_ES.properties

```
label.greeting=Hola
label.firstname=Nombre de pila
label.lastname=Apellido
label.welcome=Bienvenidos a la clase de formación.
```

mylabels.properties

```
label.greeting=Howdy
label.firstname=First Name
label.lastname=Last Name
label.welcome=Welcome to the training class.
```

mylabels_es_ES.properties

```
label.greeting=Hola
label.firstname=Nombre de pila
label.lastname=Apellido
label.welcome=Bienvenidos a la clase de formacion.
```

mylabels_de_DE.properties

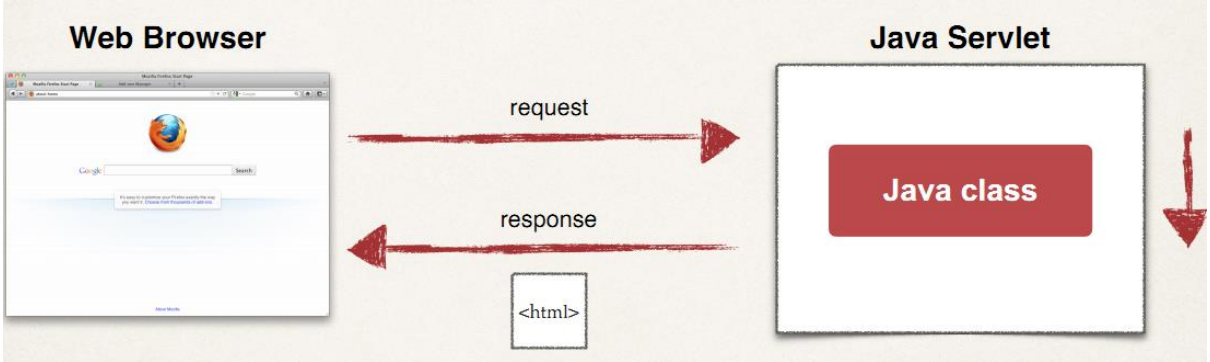
```
label.greeting=Hallo
label.firstname=Vorname
label.lastname=Nachname
label.welcome=Willkomen in der Ausbildung Klasse.
```

i18n-messages-test.jsp

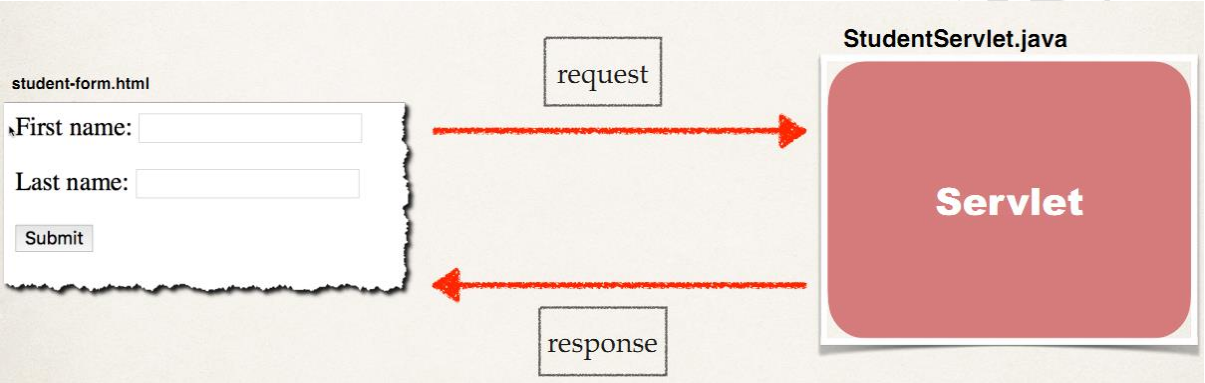
```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>

<c:set var="theLocale"
value="${not empty param.theLocale ? param.theLocale :
pageContext.request.locale}"
scope="session" />
```

```
<fmt:setLocale value="${theLocale}" />

<fmt:setBundle basename="com.luv2code.jsp.tagdemo.i18n.resources.mylabels" />

<html>

<body>

<a href="i18n-messages-test.jsp?theLocale=en_US">English (US)</a>
 |
<a href="i18n-messages-test.jsp?theLocale=es_ES">Spanish (ES)</a>
 |
<a href="i18n-messages-test.jsp?theLocale=de_DE">German (DE)</a>

<hr>

<fmt:message key="label.greeting" /> <br/> <br/>

<fmt:message key="label.firstname" /> <i>John</i> <br/>

<fmt:message key="label.lastname" /> <i>Doe</i> <br/><br/>

<fmt:message key="label.welcome" /> <br/>

<hr>

Selected locale: ${theLocale}

</body>

</html>
```

Chapter 5. Servlet



5.1. HTML Form Data with Servlets



student-form.html

```
<html>
<body>
<form action="StudentServlet" method="GET">
        First name: <input type="text" name="firstName" />
        <br/><br/>
        Last name: <input type="text" name="lastName" />
        <br/><br/>
        <input type="submit" value="Submit" />
</form>
</body>
</html>
```

StudentServlet.java

```java
package com.luv2code.servletdemo;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class StudentServlet
 */
@WebServlet("/StudentServlet")
public class StudentServlet extends HttpServlet {
      private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public StudentServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

      /**
       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
       */
      protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

            // Step 1: set content type
            response.setContentType("text/html");

            // Step 2: get the printwriter
            PrintWriter out = response.getWriter();

            // Step 3: generate the HTML content
            out.println("<html><body>");

            out.println("The student is confirmed: "
                              + request.getParameter("firstName") + " "
                              + request.getParameter("lastName"));

            out.println("</body></html>");

      }

      /**
       * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
       */
      protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
            // TODO Auto-generated method stub
            doGet(request, response);
      }
```

```
}
```

5.2. Servlet Parameters

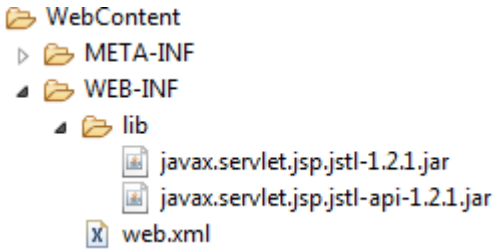
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">
  <display-name>servletdemo</display-name>
  <context-param>
    <param-name>max-shopping-cart-size</param-name>
    <param-value>99</param-value>
  </context-param>
  <context-param>
    <param-name>project-team-name</param-name>
    <param-value>The Coding Gurus</param-value>
  </context-param>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

TestParamServlet.java

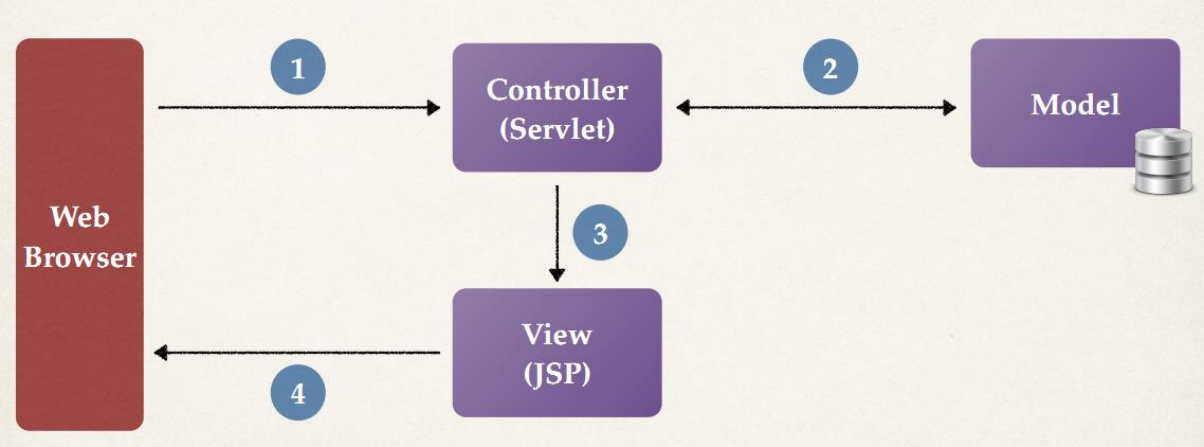
```
package com.luv2code.servletdemo;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class TestParamServlet
 */
@WebServlet("/TestParamServlet")
public class TestParamServlet extends HttpServlet {
      private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
```

```java
    public TestParamServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

      /**
       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
       */
      protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

            // Step 1: set content type
            response.setContentType("text/html");

            // Step 2: get printwriter
            PrintWriter out = response.getWriter();

            // Step 3: read configuration params
            ServletContext context = getServletContext(); // inherit from
HttpServlet

            String maxCartSize = context.getInitParameter("max-shopping-cart-
size");
            String teamName = context.getInitParameter("project-team-name");

            // Step 4: generate HTML content
            out.println("<html><body>");

            out.println("Max cart: " + maxCartSize);
            out.println("<br/><br/>");
            out.println("Team name: " + teamName);

            out.println("</body></html>");
      }

      /**
       * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
       */
      protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
            // TODO Auto-generated method stub
            doGet(request, response);
      }

}
```



Chapter 6. MVC with Servlets and JSP

6.1. Overview



MvcDemoServlet.java

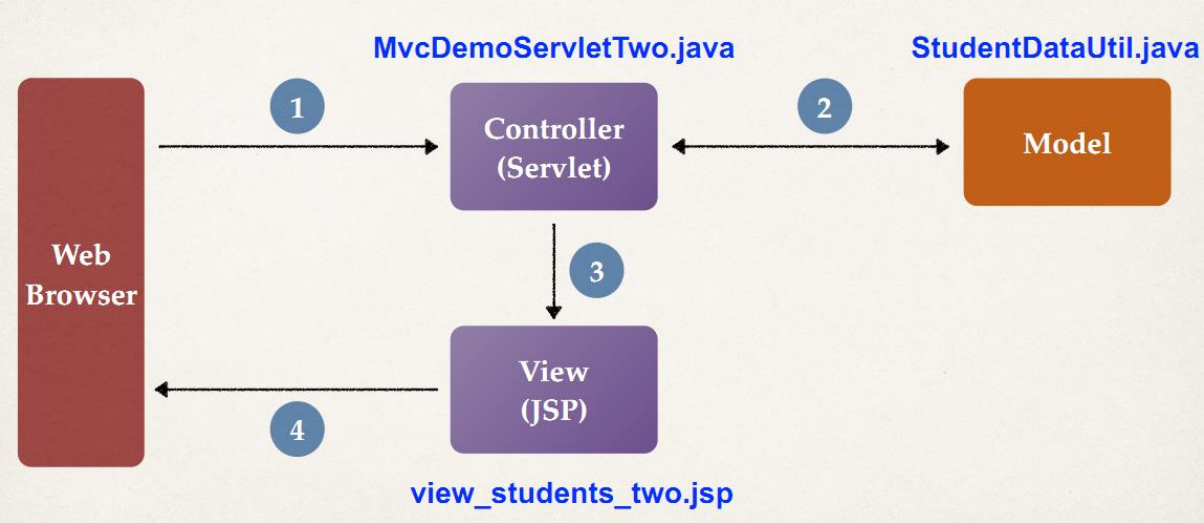
```
ackage com.luv2code.servletdemo;

import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class MvcDemoServlet
 */
@WebServlet("/MvcDemoServlet")
public class MvcDemoServlet extends HttpServlet {
      private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public MvcDemoServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

       /**
        * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
        */
```

```
	protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

		// Step 0: Add data
		String[] students = {"Susan", "Anil", "Mohamed", "Trupti"};
		request.setAttribute("student_list", students);

		// Step 1: get request dispatcher
		RequestDispatcher dispatcher =

	request.getRequestDispatcher("/view_students.jsp");

		// Step 2: forward the request to JSP
		dispatcher.forward(request, response);
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		doGet(request, response);
	}

}
```

view_students.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<html>
<body>

	<c:forEach var="tempStudent" items="${student_list}">

		${tempStudent} <br/>

	</c:forEach>

</body>
</html>
```

6.2. MVC with Servlets and JSP - More Detail



Student.java

```
package com.luv2code.servletdemo.mvctwo;

public class Student {

	private String firstName;
	private String lastName;
	private String email;

	public Student(String firstName, String lastName, String email) {
		super();
		this.firstName = firstName;
		this.lastName = lastName;
		this.email = email;
	}

	public String getFirstName() {
		return firstName;
	}

	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}

	public String getEmail() {
		return email;
	}
```

```
	public void setEmail(String email) {
		this.email = email;
	}


}
```

StudentDataUtil.java

```
package com.luv2code.servletdemo.mvctwo;

import java.util.ArrayList;
import java.util.List;

public class StudentDataUtil {

	public static List<Student> getStudents() {

		// create an empty list
		List<Student> students = new ArrayList<>();

		// add sample data
		students.add(new Student("Mary", "Public", "mary@luv2code.com"));
		students.add(new Student("John", "Doe", "john@luv2code.com"));
		students.add(new Student("Ajay", "Rao", "ajay@luv2code.com"));

		// return the list
		return students;
	}
}
```

MvcDemoServletTwo.java

```
ackage com.luv2code.servletdemo.mvctwo;

import java.io.IOException;
import java.util.List;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class MvcDemoServletTwo
 */
@WebServlet("/MvcDemoServletTwo")
public class MvcDemoServletTwo extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
```

```
     */
    public MvcDemoServletTwo() {
        super();
        // TODO Auto-generated constructor stub
    }

      /**
       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
       */
      protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

            // step 1: get the student data from helper class (model)
            List<Student> students = StudentDataUtil.getStudents();

            // step 2: add students to request object
            request.setAttribute("student_list", students);

            // step 3: get request dispatcher
            RequestDispatcher dispatcher =
                        request.getRequestDispatcher("view_students_two.jsp");


            // step 4: now forward to JSP
            dispatcher.forward(request, response);
      }

      /**
       * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
       */
      protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
            // TODO Auto-generated method stub
            doGet(request, response);
      }

}
```

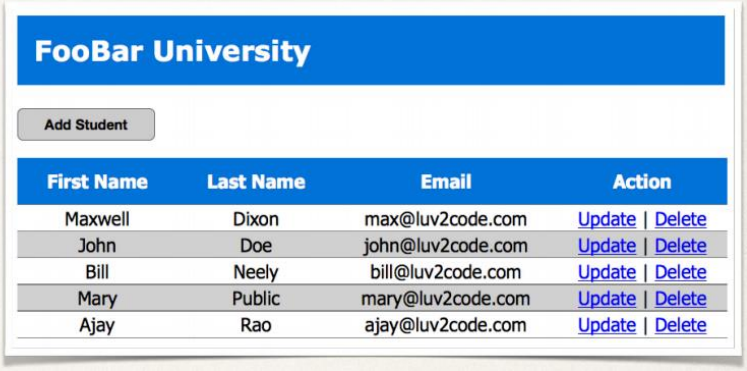
view_students_two.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<html>
<body>
<h2>Student Table Demo</h2>
<hr>
<br/>

<table border="1">

      <tr>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email Name</th>
      </tr>
```

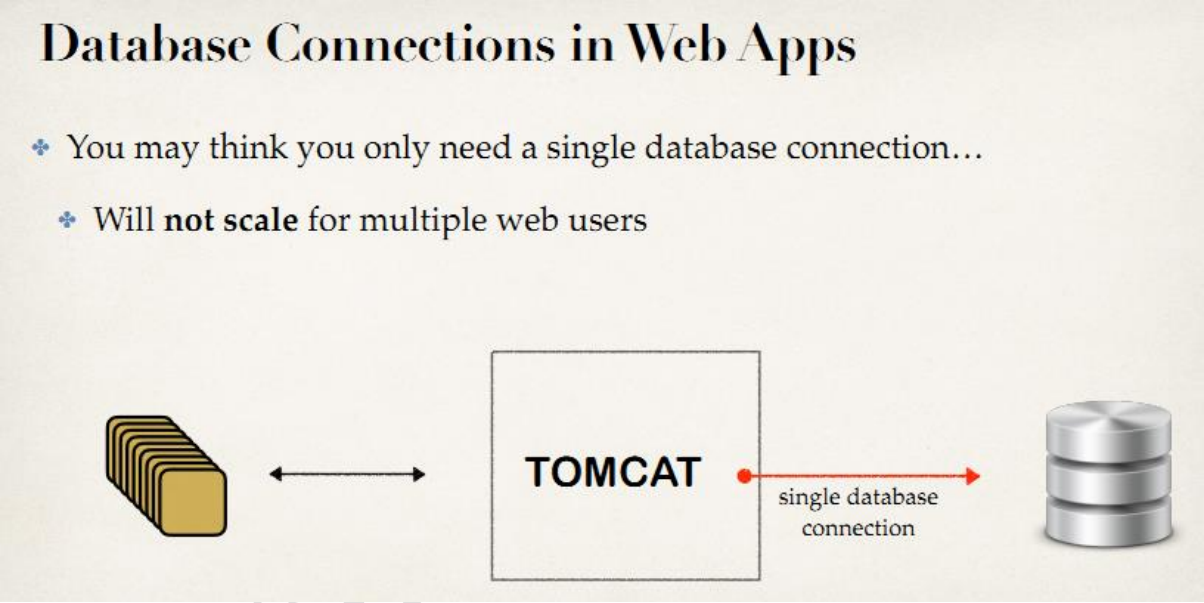
```
        <c:forEach var="tempStudent" items="${student_list}">

                <tr>
                        <td>${tempStudent.firstName}</td>
                        <td>${tempStudent.lastName}</td>
                        <td>${tempStudent.email}</td>
                </tr>

        </c:forEach>

</table>

</body>
</html>
```

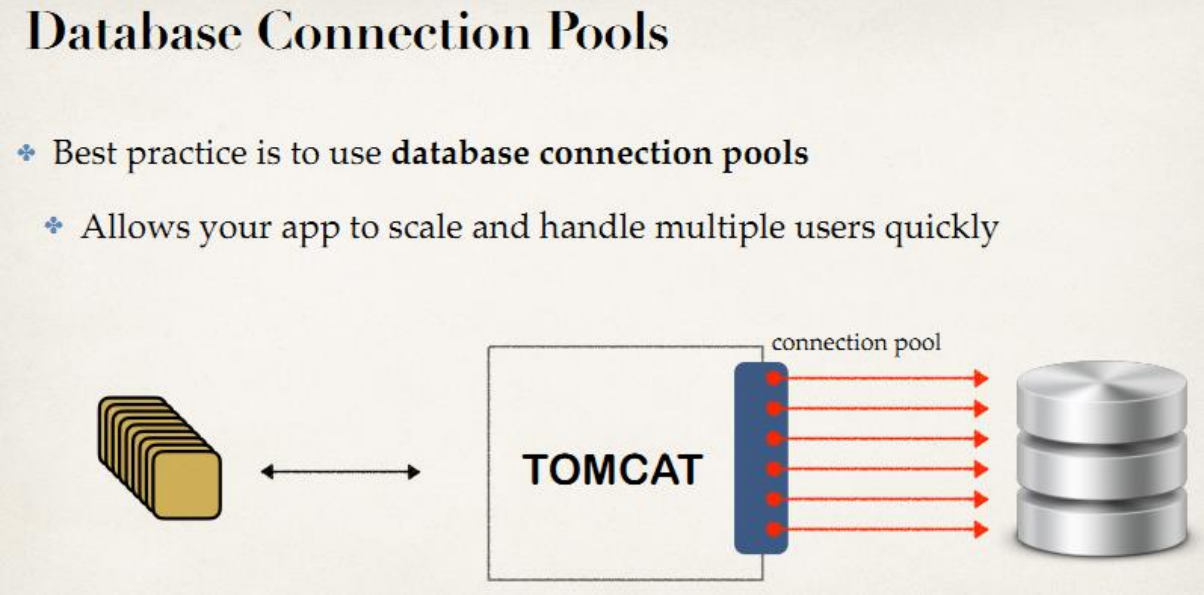
Build a complete Database Application with JDBC

- List Students
- Add a new Student
- Update a Student
- Delete a Student



Database Connection Pooling





Database Connection in Tomcat

1. Download JDBC Driver JAR file
2. Define connection pool in **`META-INF/context.xml`**
3. Get connection pool reference in Java code

Step 1

https://dev.mysql.com/downloads/

Step 2

context.xml

```
<Context>

  <Resource name="jdbc/web_student_tracker"
                  auth="Container" type="javax.sql.DataSource"
               maxActive="20" maxIdle="5" maxWait="10000"
               username="webstudent" password="webstudent"
               driverClassName="com.mysql.jdbc.Driver"

url="jdbc:mysql://localhost:3306/web_student_tracker?useSSL=false"/>

</Context>
```

Step 3

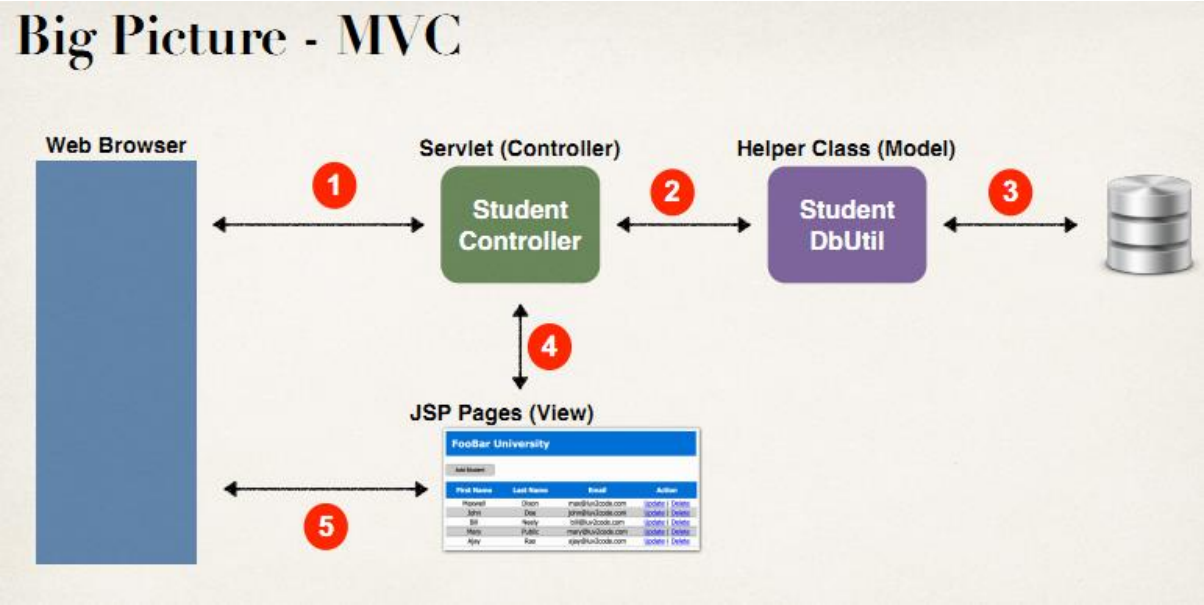
Test Tomcat Connection Pooling

TestServlet.java

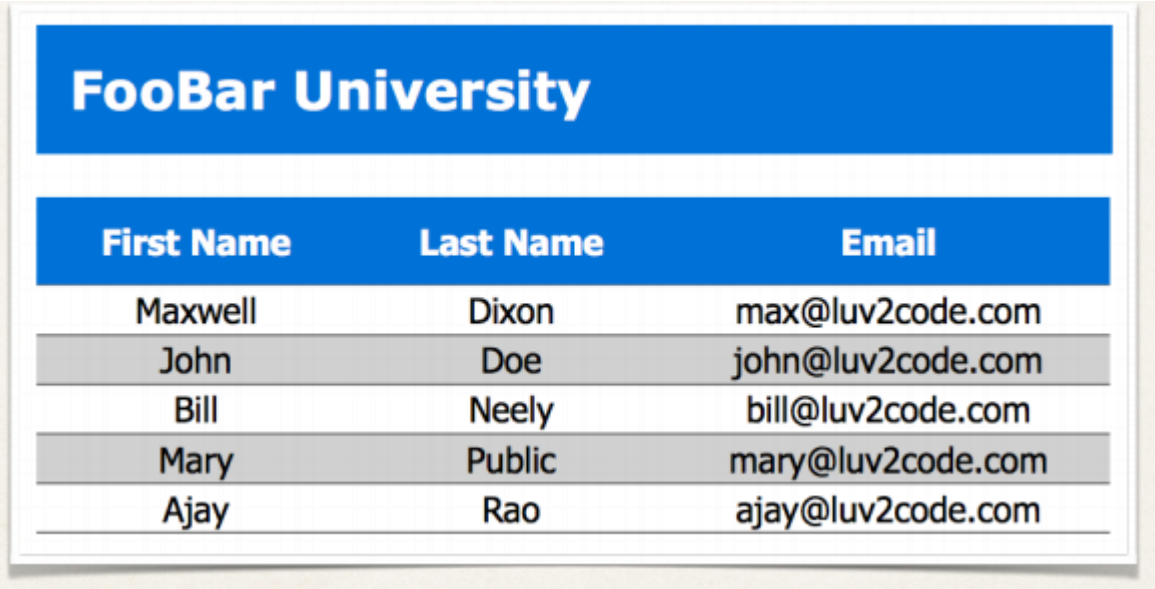
```
package com.luv2code.web.jdbc;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
```

```java
import javax.annotation.Resource;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;

/**
 * Servlet implementation class TestServlet
 */
@WebServlet("/TestServlet")
public class TestServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

	// Define datasource/connection pool for Resource Injection
	@Resource(name="jdbc/web_student_tracker")
	private DataSource dataSource;


	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

		// Step 1:  Set up the printwriter
		PrintWriter out = response.getWriter();
		response.setContentType("text/plain");

		// Step 2:  Get a connection to the database
		Connection myConn = null;
		Statement myStmt = null;
		ResultSet myRs = null;

		try {
			myConn = dataSource.getConnection();

			// Step 3:  Create a SQL statements
			String sql = "select * from student";
			myStmt = myConn.createStatement();

			// Step 4:  Execute SQL query
			myRs = myStmt.executeQuery(sql);

			// Step 5:  Process the result set
			while (myRs.next()) {
				String email = myRs.getString("email");
				out.println(email);
			}
		}
		catch (Exception exc) {
			exc.printStackTrace();
		}
	}

}
```

MVC Application Architecture

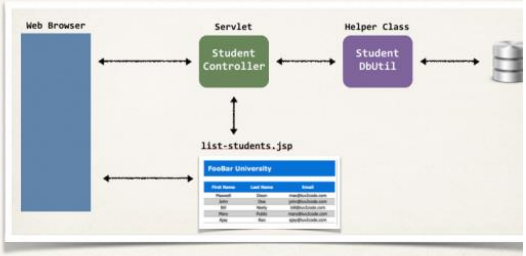


List Students



To do list

1. Create **Student.java**
2. Create **StudentDBUtil.java**
3. Create **StudentControllerServlet.java**
4. Create JSP page: **list-students.jsp**



Student.java

```
package com.luv2code.web.jdbc;

public class Student {

	private int id;
	private String firstName;
	private String lastName;
	private String email;

	public Student(String firstName, String lastName, String email) {
		this.firstName = firstName;
		this.lastName = lastName;
```

```
		this.email = email;
	}

	public Student(int id, String firstName, String lastName, String email) {
		this.id = id;
		this.firstName = firstName;
		this.lastName = lastName;
		this.email = email;
	}

	public int getId() {
		return id;
	}

	public void setId(int id) {
		this.id = id;
	}

	public String getFirstName() {
		return firstName;
	}

	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}

	public String getEmail() {
		return email;
	}

	public void setEmail(String email) {
		this.email = email;
	}

	@Override
	public String toString() {
		return "Student [id=" + id + ", firstName=" + firstName + ",
lastName=" + lastName + ", email=" + email + "]";
	}
}
```

StudentDbUtil.java

```
package com.luv2code.web.jdbc;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
```

```java
import java.util.List;

import javax.sql.DataSource;

public class StudentDbUtil {

	private DataSource dataSource;

	public StudentDbUtil(DataSource theDataSource) {
		dataSource = theDataSource;
	}

	public List<Student> getStudents() throws Exception {

		List<Student> students = new ArrayList<>();

		Connection myConn = null;
		Statement myStmt = null;
		ResultSet myRs = null;

		try {
			// get a connection
			myConn = dataSource.getConnection();

			// create sql statement
			String sql = "select * from student order by last_name";

			myStmt = myConn.createStatement();

			// execute query
			myRs = myStmt.executeQuery(sql);

			// process result set
			while (myRs.next()) {

				// retrieve data from result set row
				int id = myRs.getInt("id");
				String firstName = myRs.getString("first_name");
				String lastName = myRs.getString("last_name");
				String email = myRs.getString("email");

				// create new student object
				Student tempStudent = new Student(id, firstName,
lastName, email);

				// add it to the list of students
				students.add(tempStudent);
			}

			return students;
		}
		finally {
			// close JDBC objects
			close(myConn, myStmt, myRs);
		}
	}

	private void close(Connection myConn, Statement myStmt, ResultSet myRs) {
```

```
		try {
			if (myRs != null) {
				myRs.close();
			}

			if (myStmt != null) {
				myStmt.close();
			}

			if (myConn != null) {
				myConn.close();	// doesn't really close it ... just
puts back in connection pool
			}
		}
		catch (Exception exc) {
			exc.printStackTrace();
		}
	}


}
```

StudentControllerServlet.java

```
package com.luv2code.web.jdbc;

import java.io.IOException;
import java.util.List;

import javax.annotation.Resource;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;

/**
 * Servlet implementation class StudentControllerServlet
 */
@WebServlet("/StudentControllerServlet")
public class StudentControllerServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

	private StudentDbUtil studentDbUtil;

	@Resource(name="jdbc/web_student_tracker")
	private DataSource dataSource;

	@Override
	public void init() throws ServletException {
		super.init();
```

```
		// create our student db util ... and pass in the conn pool /
datasource
		try {
			studentDbUtil = new StudentDbUtil(dataSource);
		}
		catch (Exception exc) {
			throw new ServletException(exc);
		}
	}

	protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

		try {
			// read the "command" parameter
			String theCommand = request.getParameter("command");

			// if the command is missing, then default to listing students
			if (theCommand == null) {
				theCommand = "LIST";
			}

			// route to the appropriate method
			switch (theCommand) {

			case "LIST":
				listStudents(request, response);
				break;

			case "ADD":
				addStudent(request, response);
				break;

			case "LOAD":
				loadStudent(request, response);
				break;

			case "UPDATE":
				updateStudent(request, response);
				break;

			case "DELETE":
				deleteStudent(request, response);
				break;

			default:
				listStudents(request, response);
			}

		}
		catch (Exception exc) {
			throw new ServletException(exc);
		}

	}

	private void deleteStudent(HttpServletRequest request, HttpServletResponse
response)
		throws Exception {
```

```
	}

	private void updateStudent(HttpServletRequest request, HttpServletResponse
response)
		throws Exception {
	}

	private void loadStudent(HttpServletRequest request, HttpServletResponse
response)
		throws Exception {
	}

	private void addStudent(HttpServletRequest request, HttpServletResponse
response)
			throws Exception {
	}

	private void listStudents(HttpServletRequest request, HttpServletResponse
response)
		throws Exception {

		// get students from db util
		List<Student> students = studentDbUtil.getStudents();

		// add students to the request
		request.setAttribute("STUDENT_LIST", students);

		// send to JSP page (view)
		RequestDispatcher dispatcher = request.getRequestDispatcher("/list-
students.jsp");
		dispatcher.forward(request, response);
	}

}
```

list-students.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>
<html>

<head>
	<title>Student Tracker App</title>

	<link type="text/css" rel="stylesheet" href="css/style.css">
</head>

<body>

	<div id="wrapper">
		<div id="header">
			<h2>FooBar University</h2>
		</div>
	</div>

	<div id="container">
```

```
            <div id="content">

                <table>

                    <tr>
                        <th>First Name</th>
                        <th>Last Name</th>
                        <th>Email</th>
                    </tr>

                    <c:forEach var="tempStudent" items="${STUDENT_LIST}">


                        <tr>
                            <td> ${tempStudent.firstName} </td>
                            <td> ${tempStudent.lastName} </td>
                            <td> ${tempStudent.email} </td>

                        </tr>

                    </c:forEach>

                </table>

            </div>

        </div>
</body>


</html>
```

style.css

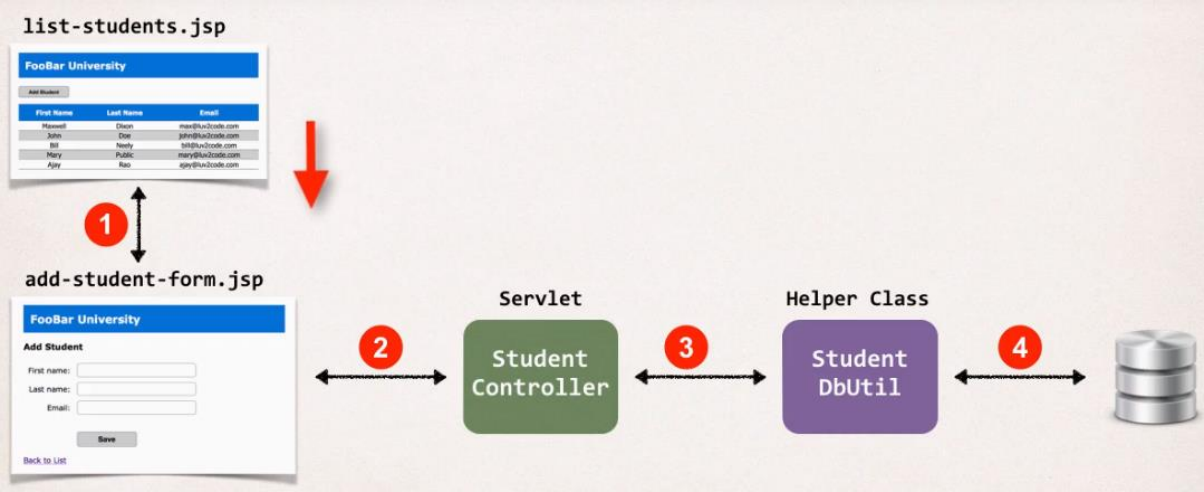
```
html, body{
        margin-left:15px; margin-right:15px;
        padding:0px;
        font-family:Verdana, Arial, Helvetica, sans-serif;
}

table {
        border-collapse:collapse;
        border-bottom:1px solid gray;
        font-family: Tahoma,Verdana,Segoe,sans-serif;
        width:72%;
}

th {
        border-bottom:1px solid gray;
        background:none repeat scroll 0 0 #0775d3;
        padding:10px;
        color: #FFFFFF;
}

tr {
        border-top:1px solid gray;
        text-align:center;
```

```
}

tr:nth-child(even) {background: #FFFFFF}
tr:nth-child(odd) {background: #BBBBBB}

#wrapper {width: 100%; margin-top: 0px; }
#header {width: 72%; background: #0775d3; margin-top: 0px; padding:15px 0px 15px
0px;}
#header h2 {width: 100%; margin:auto; color: #FFFFFF;}
#container {width: 100%; margin:auto}
#container h3 {color: #000;}
#container #content {margin-top: 20px;}

.add-student-button {
	border: 1px solid #666;
	border-radius: 5px;
	padding: 4px;
	font-size: 12px;
	font-weight: bold;
	width: 120px;
	padding: 5px 10px;

	margin-bottom: 15px;
	background: #cccccc;
}
```

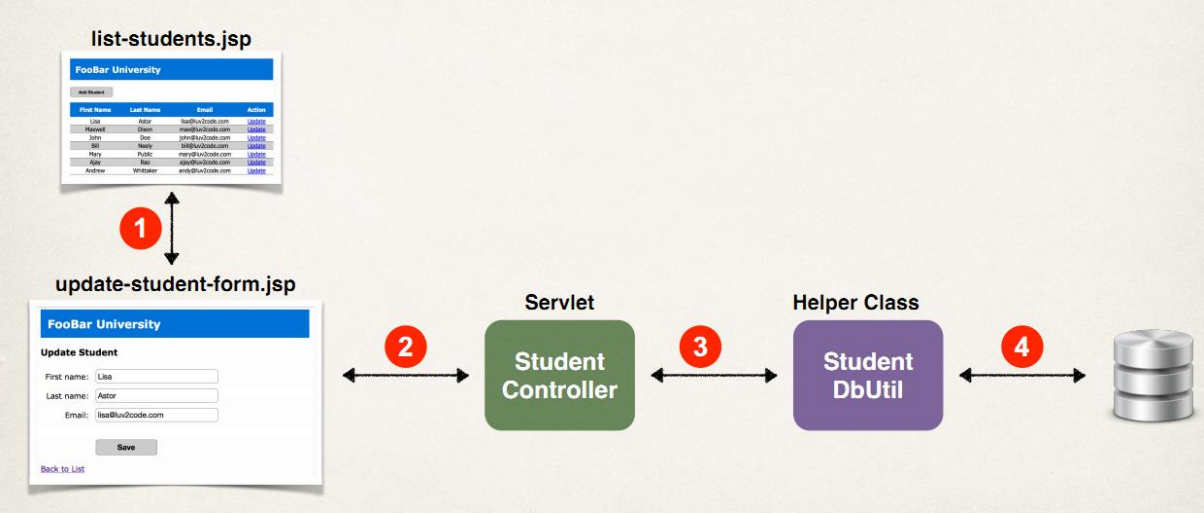
web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">
  <display-name>web-student-tracker-solution</display-name>
  <welcome-file-list>
    <welcome-file>StudentControllerServlet</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
</web-app>
```

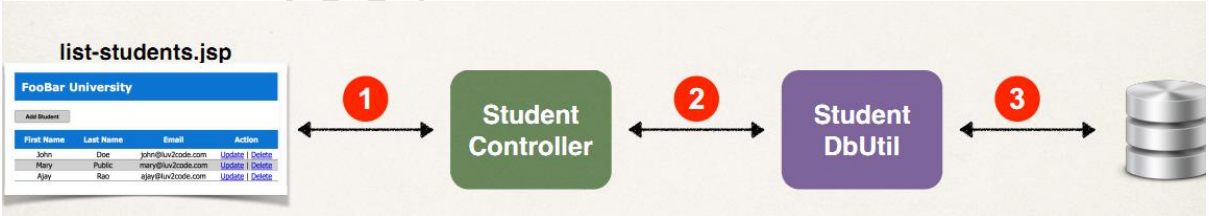
Add Students



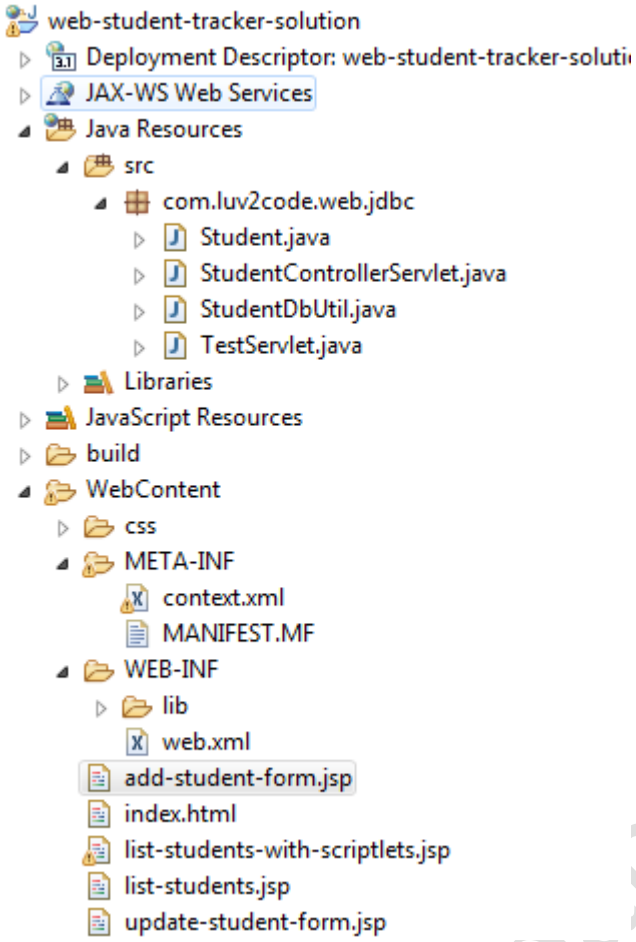
Update Students



Delete Student



Full code



Student.java

```
package com.luv2code.web.jdbc;

public class Student {

	private int id;
	private String firstName;
	private String lastName;
	private String email;

	public Student(String firstName, String lastName, String email) {
		this.firstName = firstName;
		this.lastName = lastName;
		this.email = email;
	}

	public Student(int id, String firstName, String lastName, String email) {
		this.id = id;
		this.firstName = firstName;
		this.lastName = lastName;
		this.email = email;
```

```
	}

	public int getId() {
		return id;
	}

	public void setId(int id) {
		this.id = id;
	}

	public String getFirstName() {
		return firstName;
	}

	public void setFirstName(String firstName) {
		this.firstName = firstName;
	}

	public String getLastName() {
		return lastName;
	}

	public void setLastName(String lastName) {
		this.lastName = lastName;
	}

	public String getEmail() {
		return email;
	}

	public void setEmail(String email) {
		this.email = email;
	}

	@Override
	public String toString() {
		return "Student [id=" + id + ", firstName=" + firstName + ",
lastName=" + lastName + ", email=" + email + "]";
	}
}
```

StudentDbUtil.java

```
package com.luv2code.web.jdbc;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import javax.sql.DataSource;

public class StudentDbUtil {

	private DataSource dataSource;
```

```java
	public StudentDbUtil(DataSource theDataSource) {
		dataSource = theDataSource;
	}

	public List<Student> getStudents() throws Exception {

		List<Student> students = new ArrayList<>();

		Connection myConn = null;
		Statement myStmt = null;
		ResultSet myRs = null;

		try {
			// get a connection
			myConn = dataSource.getConnection();

			// create sql statement
			String sql = "select * from student order by last_name";

			myStmt = myConn.createStatement();

			// execute query
			myRs = myStmt.executeQuery(sql);

			// process result set
			while (myRs.next()) {

				// retrieve data from result set row
				int id = myRs.getInt("id");
				String firstName = myRs.getString("first_name");
				String lastName = myRs.getString("last_name");
				String email = myRs.getString("email");

				// create new student object
				Student tempStudent = new Student(id, firstName,
lastName, email);

				// add it to the list of students
				students.add(tempStudent);
			}

			return students;
		}
		finally {
			// close JDBC objects
			close(myConn, myStmt, myRs);
		}
	}

	private void close(Connection myConn, Statement myStmt, ResultSet myRs) {

		try {
			if (myRs != null) {
				myRs.close();
			}

			if (myStmt != null) {
				myStmt.close();
			}
```

```
                if (myConn != null) {
                        myConn.close();   // doesn't really close it ... just
puts back in connection pool
                }
        }
        catch (Exception exc) {
                exc.printStackTrace();
        }
    }

    public void addStudent(Student theStudent) throws Exception {

        Connection myConn = null;
        PreparedStatement myStmt = null;

        try {
                // get db connection
                myConn = dataSource.getConnection();

                // create sql for insert
                String sql = "insert into student "
                                + "(first_name, last_name, email) "
                                + "values (?, ?, ?)";

                myStmt = myConn.prepareStatement(sql);

                // set the param values for the student
                myStmt.setString(1, theStudent.getFirstName());
                myStmt.setString(2, theStudent.getLastName());
                myStmt.setString(3, theStudent.getEmail());

                // execute sql insert
                myStmt.execute();
        }
        finally {
                // clean up JDBC objects
                close(myConn, myStmt, null);
        }
    }

    public Student getStudent(String theStudentId) throws Exception {

        Student theStudent = null;

        Connection myConn = null;
        PreparedStatement myStmt = null;
        ResultSet myRs = null;
        int studentId;

        try {
                // convert student id to int
                studentId = Integer.parseInt(theStudentId);

                // get connection to database
                myConn = dataSource.getConnection();

                // create sql to get selected student
                String sql = "select * from student where id=?";
```

```
                // create prepared statement
                myStmt = myConn.prepareStatement(sql);

                // set params
                myStmt.setInt(1, studentId);

                // execute statement
                myRs = myStmt.executeQuery();

                // retrieve data from result set row
                if (myRs.next()) {
                        String firstName = myRs.getString("first_name");
                        String lastName = myRs.getString("last_name");
                        String email = myRs.getString("email");

                        // use the studentId during construction
                        theStudent = new Student(studentId, firstName,
lastName, email);
                }
                else {
                        throw new Exception("Could not find student id: " +
studentId);
                }

                return theStudent;
        }
        finally {
                // clean up JDBC objects
                close(myConn, myStmt, myRs);
        }
    }

    public void updateStudent(Student theStudent) throws Exception {

        Connection myConn = null;
        PreparedStatement myStmt = null;

        try {
                // get db connection
                myConn = dataSource.getConnection();

                // create SQL update statement
                String sql = "update student "
                                + "set first_name=?, last_name=?, email=?
"
                                + "where id=?";

                // prepare statement
                myStmt = myConn.prepareStatement(sql);

                // set params
                myStmt.setString(1, theStudent.getFirstName());
                myStmt.setString(2, theStudent.getLastName());
                myStmt.setString(3, theStudent.getEmail());
                myStmt.setInt(4, theStudent.getId());

                // execute SQL statement
                myStmt.execute();
```

```
		}
		finally {
			// clean up JDBC objects
			close(myConn, myStmt, null);
		}
	}

	public void deleteStudent(String theStudentId) throws Exception {

		Connection myConn = null;
		PreparedStatement myStmt = null;

		try {
			// convert student id to int
			int studentId = Integer.parseInt(theStudentId);

			// get connection to database
			myConn = dataSource.getConnection();

			// create sql to delete student
			String sql = "delete from student where id=?";

			// prepare statement
			myStmt = myConn.prepareStatement(sql);

			// set params
			myStmt.setInt(1, studentId);

			// execute sql statement
			myStmt.execute();
		}
		finally {
			// clean up JDBC code
			close(myConn, myStmt, null);
		}
	}
}
```

StudentControllerServlet.java

```
package com.luv2code.web.jdbc;

import java.io.IOException;
import java.util.List;

import javax.annotation.Resource;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;

/**
 * Servlet implementation class StudentControllerServlet
 */
@WebServlet("/StudentControllerServlet")
```

```java
public class StudentControllerServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

	private StudentDbUtil studentDbUtil;

	@Resource(name="jdbc/web_student_tracker")
	private DataSource dataSource;

	@Override
	public void init() throws ServletException {
		super.init();

		// create our student db util ... and pass in the conn pool /
datasource
		try {
			studentDbUtil = new StudentDbUtil(dataSource);
		}
		catch (Exception exc) {
			throw new ServletException(exc);
		}
	}

	protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

		try {
			// read the "command" parameter
			String theCommand = request.getParameter("command");

			// if the command is missing, then default to listing students
			if (theCommand == null) {
				theCommand = "LIST";
			}

			// route to the appropriate method
			switch (theCommand) {

			case "LIST":
				listStudents(request, response);
				break;

			case "ADD":
				addStudent(request, response);
				break;

			case "LOAD":
				loadStudent(request, response);
				break;

			case "UPDATE":
				updateStudent(request, response);
				break;

			case "DELETE":
				deleteStudent(request, response);
				break;

			default:
				listStudents(request, response);
```

```
                }

            }
            catch (Exception exc) {
                throw new ServletException(exc);
            }

        }

        private void deleteStudent(HttpServletRequest request, HttpServletResponse
response)
            throws Exception {

            // read student id from form data
            String theStudentId = request.getParameter("studentId");

            // delete student from database
            studentDbUtil.deleteStudent(theStudentId);

            // send them back to "list students" page
            listStudents(request, response);
        }

        private void updateStudent(HttpServletRequest request, HttpServletResponse
response)
            throws Exception {

            // read student info from form data
            int id = Integer.parseInt(request.getParameter("studentId"));
            String firstName = request.getParameter("firstName");
            String lastName = request.getParameter("lastName");
            String email = request.getParameter("email");

            // create a new student object
            Student theStudent = new Student(id, firstName, lastName, email);

            // perform update on database
            studentDbUtil.updateStudent(theStudent);

            // send them back to the "list students" page
            listStudents(request, response);

        }

        private void loadStudent(HttpServletRequest request, HttpServletResponse
response)
            throws Exception {

            // read student id from form data
            String theStudentId = request.getParameter("studentId");

            // get student from database (db util)
            Student theStudent = studentDbUtil.getStudent(theStudentId);

            // place student in the request attribute
            request.setAttribute("THE_STUDENT", theStudent);

            // send to jsp page: update-student-form.jsp
            RequestDispatcher dispatcher =
```

```
                        request.getRequestDispatcher("/update-student-
form.jsp");
            dispatcher.forward(request, response);
        }

        private void addStudent(HttpServletRequest request, HttpServletResponse
response) throws Exception {

            // read student info from form data
            String firstName = request.getParameter("firstName");
            String lastName = request.getParameter("lastName");
            String email = request.getParameter("email");

            // create a new student object
            Student theStudent = new Student(firstName, lastName, email);

            // add the student to the database
            studentDbUtil.addStudent(theStudent);

            // send back to main page (the student list)
            listStudents(request, response);
        }

        private void listStudents(HttpServletRequest request, HttpServletResponse
response)
            throws Exception {

            // get students from db util
            List<Student> students = studentDbUtil.getStudents();

            // add students to the request
            request.setAttribute("STUDENT_LIST", students);

            // send to JSP page (view)
            RequestDispatcher dispatcher = request.getRequestDispatcher("/list-
students.jsp");
            dispatcher.forward(request, response);
        }

}
```

list-students.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>
<html>

<head>
        <title>Student Tracker App</title>

        <link type="text/css" rel="stylesheet" href="css/style.css">
</head>

<body>

        <div id="wrapper">
```

```
            <div id="header">
                  <h2>FooBar University</h2>
            </div>
      </div>

      <div id="container">

            <div id="content">

                  <!-- put new button: Add Student -->

                  <input type="button" value="Add Student"
                              onclick="window.location.href='add-student-
form.jsp'; return false;"
                              class="add-student-button"
                  />

                  <table>

                        <tr>
                              <th>First Name</th>
                              <th>Last Name</th>
                              <th>Email</th>
                              <th>Action</th>
                        </tr>

                        <c:forEach var="tempStudent" items="${STUDENT_LIST}">

                              <!-- set up a link for each student -->
                              <c:url var="tempLink"
value="StudentControllerServlet">
                                    <c:param name="command" value="LOAD" />
                                    <c:param name="studentId"
value="${tempStudent.id}" />
                              </c:url>

                              <!--  set up a link to delete a student -->
                              <c:url var="deleteLink"
value="StudentControllerServlet">
                                    <c:param name="command" value="DELETE" />
                                    <c:param name="studentId"
value="${tempStudent.id}" />
                              </c:url>


                              <tr>
                                    <td> ${tempStudent.firstName} </td>
                                    <td> ${tempStudent.lastName} </td>
                                    <td> ${tempStudent.email} </td>
                                    <td>
                                          <a href="${tempLink}">Update</a>
                                           |
                                          <a href="${deleteLink}"
                                          onclick="if (!(confirm('Are you sure
you want to delete this student?'))) return false">
                                          Delete</a>
                                    </td>
                              </tr>
```

```
                        </c:forEach>

                    </table>

                </div>

        </div>
</body>


</html>
```

add-student-form.jsp

```
<!DOCTYPE html>
<html>

<head>
        <title>Add Student</title>

        <link type="text/css" rel="stylesheet" href="css/style.css">
        <link type="text/css" rel="stylesheet" href="css/add-student-style.css">
</head>

<body>
        <div id="wrapper">
                <div id="header">
                        <h2>FooBar University</h2>
                </div>
        </div>

        <div id="container">
                <h3>Add Student</h3>

                <form action="StudentControllerServlet" method="GET">

                        <input type="hidden" name="command" value="ADD" />

                        <table>
                                <tbody>
                                        <tr>
                                                <td><label>First name:</label></td>
                                                <td><input type="text" name="firstName"
/></td>
                                        </tr>

                                        <tr>
                                                <td><label>Last name:</label></td>
                                                <td><input type="text" name="lastName"
/></td>
                                        </tr>

                                        <tr>
                                                <td><label>Email:</label></td>
                                                <td><input type="text" name="email"
/></td>
                                        </tr>
```

```
                                <tr>
                                        <td><label></label></td>
                                        <td><input type="submit" value="Save"
class="save" /></td>
                                </tr>

                        </tbody>
                </table>
        </form>

        <div style="clear: both;"></div>

        <p>
                <a href="StudentControllerServlet">Back to List</a>
        </p>
    </div>
</body>

</html>
```

update-student-form.jsp

```
<!DOCTYPE html>
<html>

<head>
    <title>Update Student</title>

    <link type="text/css" rel="stylesheet" href="css/style.css">
    <link type="text/css" rel="stylesheet" href="css/add-student-style.css">
</head>

<body>
    <div id="wrapper">
        <div id="header">
                <h2>FooBar University</h2>
        </div>
    </div>

    <div id="container">
        <h3>Update Student</h3>

        <form action="StudentControllerServlet" method="GET">

                <input type="hidden" name="command" value="UPDATE" />

                <input type="hidden" name="studentId"
value="${THE_STUDENT.id}" />

                <table>
                        <tbody>
                                <tr>
                                        <td><label>First name:</label></td>
                                        <td><input type="text" name="firstName"

value="${THE_STUDENT.firstName}" /></td>
                                </tr>
```

```
                                <tr>
                                        <td><label>Last name:</label></td>
                                        <td><input type="text" name="lastName"

value="${THE_STUDENT.lastName}" /></td>
                                </tr>

                                <tr>
                                        <td><label>Email:</label></td>
                                        <td><input type="text" name="email"

value="${THE_STUDENT.email}" /></td>
                                </tr>

                                <tr>
                                        <td><label></label></td>
                                        <td><input type="submit" value="Save"
class="save" /></td>
                                </tr>

                        </tbody>
                    </table>
            </form>

            <div style="clear: both;"></div>

            <p>
                    <a href="StudentControllerServlet">Back to List</a>
            </p>
        </div>
</body>

</html>
```