

Agenda:

- Basic HTML concepts
- HTML Forms and Servlets
- Tomcat Common Bean Utility library

A crash course in HTML

HTML basics

- *HyperText Markup Language (HTML)* is used to provide the user interface for web applications.
- To write and edit HTML code and JSPs, you can use a general text editor like NotePad, a text editor that's specifically designed for working with HTML, or an *Integrated Development Environment*, or *IDE*, that's designed for developing web applications.
- An *HTML document* is used to define each *HTML page* that's displayed by a web browser.
- Within an HTML document, *HTML tags* define how the page will look when it is displayed. Each of these HTML tags is coded within a set of brackets (< >).
- HTML tags aren't case sensitive.
- To make your code easier to read, you can use spaces, indentation, and blank lines.

Basic HTML tags

Tag	Description
<code><!doctype ... ></code>	Identifies the type of HTML document. This tag is often inserted automatically by the HTML editor.
<code><html> </html></code>	Marks the beginning and end of an HTML document.
<code><head> </head></code>	Marks the beginning and end of the Head section of the HTML document.
<code><title> </title></code>	Marks the title that is displayed in the title bar of the browser.
<code><body> </body></code>	Marks the beginning and end of the Body section of the HTML document.

Basic HTML tags (cont.)

Tag	Description
<code><h1> </h1></code>	Tells the browser to use the default format for a heading-1 paragraph.
<code><h2> </h2></code>	Tells the browser to use the default format for a heading-2 paragraph.
<code><p> </p></code>	Tells the browser to use the default format for a standard paragraph.
<code>
</code>	Inserts a line break.
<code> </code>	Marks text as bold.
<code><i> </i></code>	Marks text as italic.
<code><u> </u></code>	Marks text as underlined.
<code><!-- --></code>	Defines a comment that is ignored by the browser.

Anchor tags...

With URLs that are relative to the current directory

```
<a href="join.html">The Email List application 1</a><br>
<a href="email/join.html">
    The Email List application 2</a><br>
```

With relative URLs that navigate up the directory structure

```
<a href="..">Go back one directory level</a><br>
<a href="..">Go back two directory levels</a><br>
```

With URLs that are relative to the webapps directory

```
<a href="/">Go to the default root directory for the web
server</a><br>
<a href="/musicStore">Go to the root directory of the
musicStore app</a>
```

With absolute URLs

```
<a href=
    "http://www.murach.com/email">An Internet address</a>
<a href="http://64.71.179.86/email">An IP address</a>
```

The Anchor tag

Tag	Description
<code><a></code> <code></code>	Defines a link to another URL. When the user clicks on the text that's displayed by the tag, the browser requests the page that is identified by the Href attribute of the tag.

One attribute of the Anchor tag

Attribute	Description
<code>href</code>	Specifies the URL for the link.

The HTML code for a table

```
<p>Here is the information that you entered:</p>
```

```
<table cellpadding="5" cellspacing="5" border="1">
  <tr>
    <td align="right">First name:</td>
    <td>John</td>
  </tr>
  <tr>
    <td align="right">Last name:</td>
    <td>Smith</td>
  </tr>
  <tr>
    <td align="right">Email address:</td>
    <td>jsmith@hotmail.com</td>
  </tr>
</table>
```


The table displayed in a browser

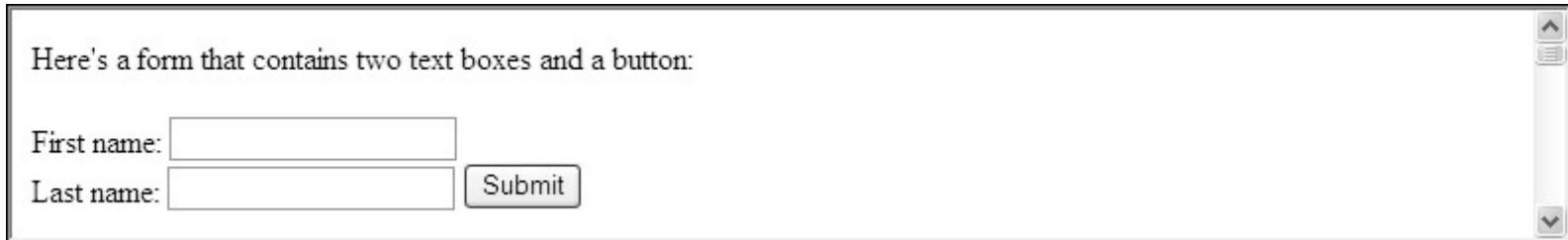
Here is the information that you entered:

First name:	John
Last name:	Smith
Email address:	jsmith@hotmail.com

The tags for working with tables

Tag	Description
<code><table> </table></code>	Marks the start and end of the table.
<code><tr> </tr></code>	Marks the start and end of each row.
<code><td> </td></code>	Marks the start and end of each data cell within a row.

A form displayed in a browser before the user enters data



Here's a form that contains two text boxes and a button:

First name:

Last name:

Description

- A *form* contains one or more *controls* such as text boxes, buttons, check boxes, and list boxes.

Web Site for HTML tags Demos

The screenshot shows the w3schools.com website in a Mozilla Firefox browser window. The page is titled "HTML Basic" and is located at www.w3schools.com/html/html_basic.asp. The website features a navigation menu with links to various web technologies: HOME, HTML, CSS, JAVASCRIPT, JQUERY, XML, ASP.NET, PHP, SQL, and MORE... There are also links to REFERENCES, EXAMPLES, FORUM, and ABOUT. The main content area is titled "HTML Basic - 4 Examples" and includes a "Previous" link and a "Next Chapter" link. The text states: "Don't worry if the examples use tags you have not learned. You will learn about them in the next chapters." The first example is "HTML Headings", which explains that HTML headings are defined with the <h1> to <h6> tags. An example code block shows:

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```

 Below the code block is a "Try it yourself" button. The second example is "HTML Paragraphs", which explains that HTML paragraphs are defined with the <p> tag. An example code block is shown below the text. The website also has a sidebar with a list of HTML topics: HTML Basic, HTML HOME, HTML Introduction, HTML Editors, HTML Elements, HTML Attributes, HTML Headings, HTML Paragraphs, HTML Formatting, HTML Links, HTML Head, HTML CSS, HTML Images, HTML Tables, HTML Lists, HTML Blocks, HTML Layout, HTML Forms, HTML Iframes, HTML Colors, HTML Colomames, and HTML Colorvalues. There are also several advertisements and promotional banners, including a "Join Google" banner, a "SHARE THIS PAGE" section, and a "WEB HOSTING" section. The footer of the page includes a "Secure Search" bar and a "McAfee" logo.

HTML Basic - Mozilla Firefox

File Edit View History Bookmarks Tools Help

HTML Basic

www.w3schools.com/html/html_basic.asp

Secure Search

w3schools.com

Select Language | Search w3schools.com: Google Custom Search

HOME HTML CSS JAVASCRIPT JQUERY XML ASP.NET PHP SQL MORE... REFERENCES | EXAMPLES | FORUM | ABOUT

SQL Server Query Tool
www.Confio.com/SQL-Server...
Improve SQL Server Performance 65%, Save on Costs, Free Trial Download!

GET YOUR BUSINESS ONLINE
Join Google
Get your business online for FREE—with help from Google.
REGISTER TODAY!
Free website includes domain name and hosting for one year.

SHARE THIS PAGE
f t e d i s u
Like 85k

WEB HOSTING
Best Web Hosting
eUK Web Hosting
UK Reseller Hosting
Domain, Hosting & Email

WEB BUILDING
XML Editor - Free Trial!
FREE Website BUILDER
FREE Website Creator
Best Website Templates

STATISTICS
Browser Statistics
OS Statistics
Display Statistics

Nissan LEAF®
MOUSE OVER TO EXPLORE

HTML Basic

HTML HOME

HTML Introduction

HTML Editors

HTML Basic

HTML Elements

HTML Attributes

HTML Headings

HTML Paragraphs

HTML Formatting

HTML Links

HTML Head

HTML CSS

HTML Images

HTML Tables

HTML Lists

HTML Blocks

HTML Layout

HTML Forms

HTML Iframes

HTML Colors

HTML Colomames

HTML Colorvalues

HTML Basic - 4 Examples

« Previous Next Chapter »

Don't worry if the examples use tags you have not learned.
You will learn about them in the next chapters.

HTML Headings

HTML headings are defined with the <h1> to <h6> tags.

Example

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```

Try it yourself »

HTML Paragraphs

HTML paragraphs are defined with the <p> tag.

Example

Web Site for HTML tags Demos

Tryit Editor v1.8 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

HTML Basic Tryit Editor v1.8

www.w3schools.com/html/tryit.asp?filename=tryhtml_headers

Unlimited Music Made Easy
play.google.com/music
Discover and play millions of songs with Google Play Music All Access.

Computer Consultant
www.AgilityNetworks.com
On-site Computer Consultant Services for Businesses in Chicago

Source Code:

```
<!DOCTYPE html>
<html>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>

</body>
</html>
```

Submit Code »

Result:

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

W3Schools.com - Try it yourself

Edit the code above and click "Submit Code" to see the result.

Secure Search McAfee

Web Site for HTML tags Demos

The screenshot shows a Mozilla Firefox browser window with the title "HTML Tables - Mozilla Firefox". The address bar displays "www.w3schools.com/html/html_tables.asp". The page content is organized into a sidebar on the left and a main content area on the right.

Left Sidebar:

- HTML Object
- HTML Audio
- HTML Video
- HTML YouTube
- HTML Examples**
 - HTML Examples
 - HTML Quiz
 - HTML5 Quiz
 - HTML Certificate
 - HTML5 Certificate
- HTML References**
 - HTML Tag List
 - HTML Attributes
 - HTML Events
 - HTML Canvas
 - HTML Audio/Video
 - HTML Doctypes
 - HTML Colomames
 - HTML Colorpicker
 - HTML Colomixer
 - HTML Character Sets
 - HTML ASCII
 - HTML ISO-8859-1
 - HTML Symbols
 - HTML URL Encode
 - HTML Lang Codes
 - HTTP Messages
 - HTTP Methods
 - Keyboard Shortcuts

Main Content Area:

HTML Table Headers

Header information in a table are defined with the <th> tag.

All major browsers display the text in the <th> element as bold and centered.

```
<table border="1">
<tr>
<th>Header 1</th>
<th>Header 2</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

How the HTML code above looks in your browser:

Header 1	Header 2
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

More Examples

- [Tables without borders](#)
How to create tables without borders.
- [Table headers](#)
How to create table headers.
- [Table with a caption](#)
How to add a caption to a table.
- [Table cells that span more than one row/column](#)
How to define table cells that span more than one row or one column.

Web Site for HTML tags Demos

Tryit Editor v1.8 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

HTML Tables Tryit Editor v1.8

www.w3schools.com/html/tryit.asp?filename=tryhtml_tables

Secure Search

Source Code:

Submit Code »

```
<!DOCTYPE html>
<html>
<body>

<p>
Each table starts with a table tag.
Each table row starts with a tr tag.
Each table data starts with a td tag.
</p>

<h4>One column:</h4>
<table border="1">
<tr>
  <td>100</td>
</tr>
</table>

<h4>One row and three columns:</h4>
<table border="1">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
</table>

<h4>Two rows and three columns:</h4>
<table border="1">
<tr>
  <td>100</td>
  <td>200</td>
  <td>300</td>
</tr>
<tr>
  <td>400</td>
  <td>500</td>
  <td>600</td>
</tr>
</table>
```

Result:

W3Schools.com - Try it yourself

Each table starts with a table tag. Each table row starts with a tr tag. Each table data starts with a td tag.

One column:

100

One row and three columns:

100	200	300
-----	-----	-----

Two rows and three columns:

100	200	300
400	500	600

Web Site for HTML tags Demos

Tryit Editor v1.8 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

HTML Tables Tryit Editor v1.8

www.w3schools.com/html/tryit.asp?filename=tryhtml_tables

Secure Search

SQL Server Query Tool
www.Confio.com/SQL-Server
Identify Bottlenecks & Wait
Types. DBA Performance
Tools. Free Trial.

\$1 A WEEK for **12 WEEKS** **THE WALL STREET JOURNAL.**
SUBSCRIBE NOW!

Source Code: Submit Code »

```
<!DOCTYPE html>
<html>
<body>

<p>
Hello World ... A. Bader
</p>

<p>
Each table starts with a table tag.
Each table row starts with a tr tag.
Each table data starts with a td tag.
</p>

<h4>One column:</h4>
<table border="1">
<tr>
<td>100</td>
</tr>
</table>

<h4>One row and three columns:</h4>
<table border="1">
<tr>
<td>100</td>
<td>200</td>
<td>300</td>
</tr>
</table>

<h4>Two rows and three columns:</h4>
<table border="1">
<tr>
<td>100</td>
<td>200</td>
<td>300</td>
</tr>
<tr>
<td>400</td>
<td>500</td>
<td>600</td>
</tr>
</table>

</body>
</html>
```

Result: W3Schools.com - Try it yourself

Each table starts with a table tag. Each table row starts with a tr tag. Each table data starts with a td tag.

One column:

100

One row and three columns:

100	200	300
-----	-----	-----

Two rows and three columns:

100	200	300
400	500	600

Edit the code above and click "Submit Code" to see the result.

Secure Search McAfee

HTML forms

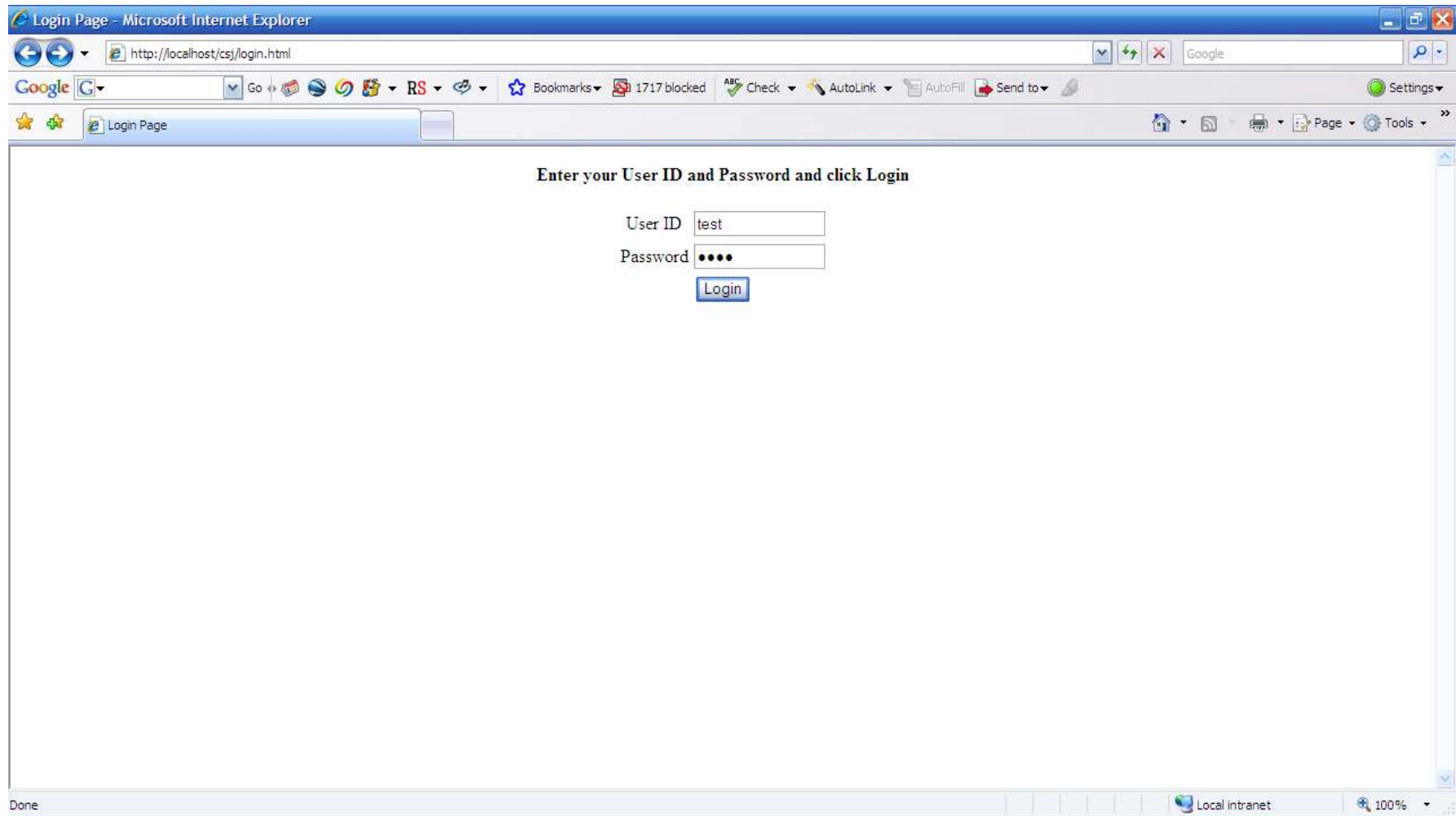
- An HTML form is identified in HTML source by the `<form>` tag

```
<form method=m action=uri>
    ... more html, including fields
</form>
```
- The **method** is the HTTP request method to use when submitting the data (POST or GET)
- The **action** attribute is the server resource that will handle the request. This points to the servlet URL that we will write to handle this form.

GET vs. POST

- The form data is submitted as part of the HTTP request as a sequence of *URL-encoded* name-value pairs.
- The GET method encodes the name-value pairs onto the request URL (i.e. Visible in the Location of the browser)
- The POST method encodes the name-value pairs in the message body of the request. (invisible to the user, but not encrypted)
- GET is the default method

A Login HTML form



The screenshot shows a Microsoft Internet Explorer browser window titled "Login Page - Microsoft Internet Explorer". The address bar displays "http://localhost/cs/login.html". The browser's toolbar includes buttons for back, forward, stop, and refresh, along with a search bar containing "Google". Below the toolbar, there are various utility buttons like "Go", "Bookmarks", "1717 blocked", "Check", "AutoLink", "AutoFill", "Send to", and "Settings". The main content area of the browser displays a login form with the following elements:

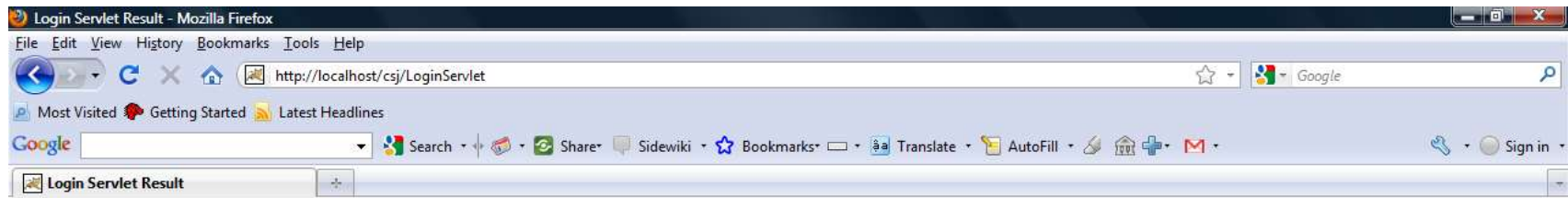
Enter your User ID and Password and click Login

User ID

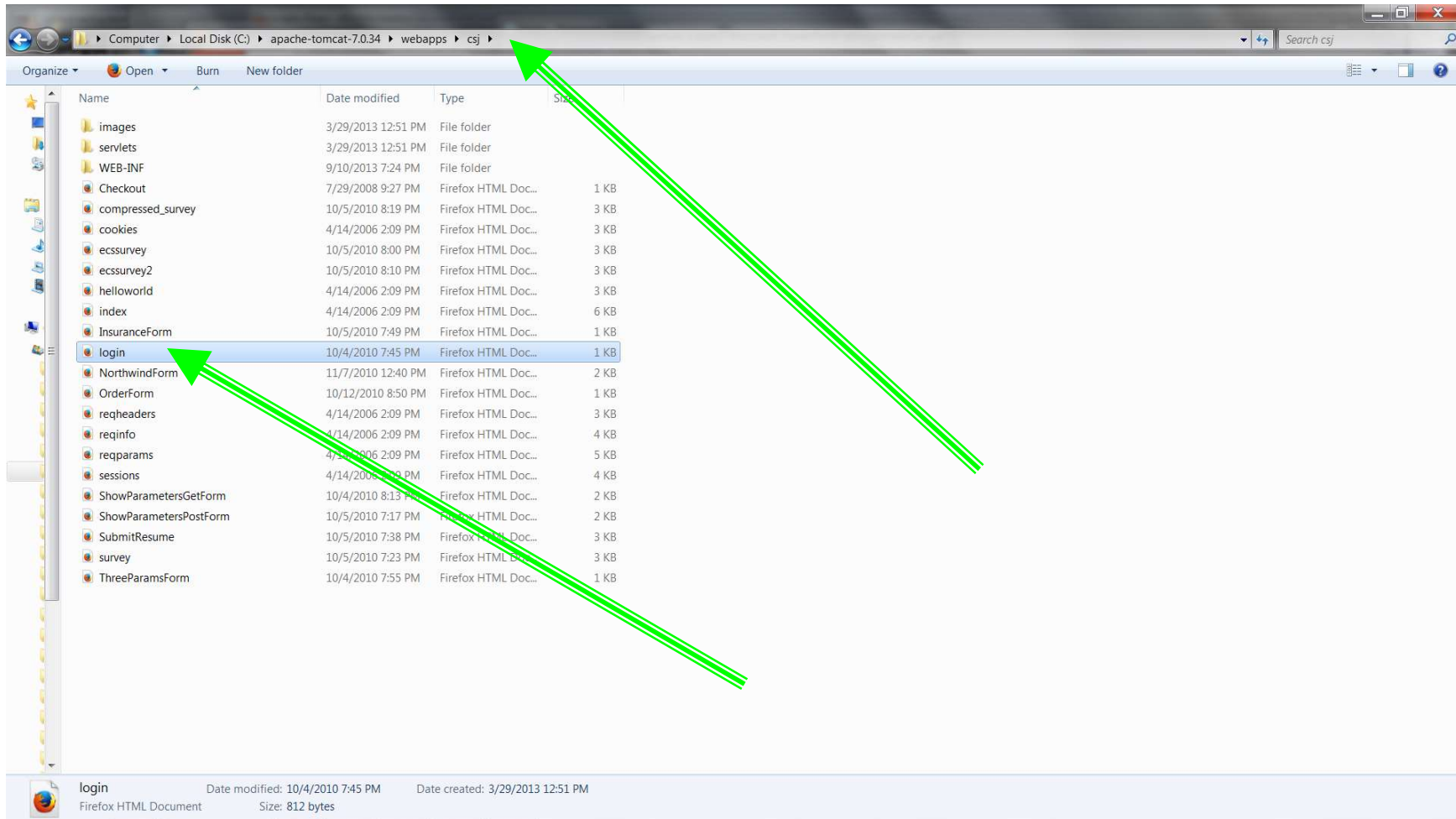
Password

The status bar at the bottom of the browser window shows "Done" on the left and "Local intranet" and "100%" zoom level on the right.

A Login HTML form



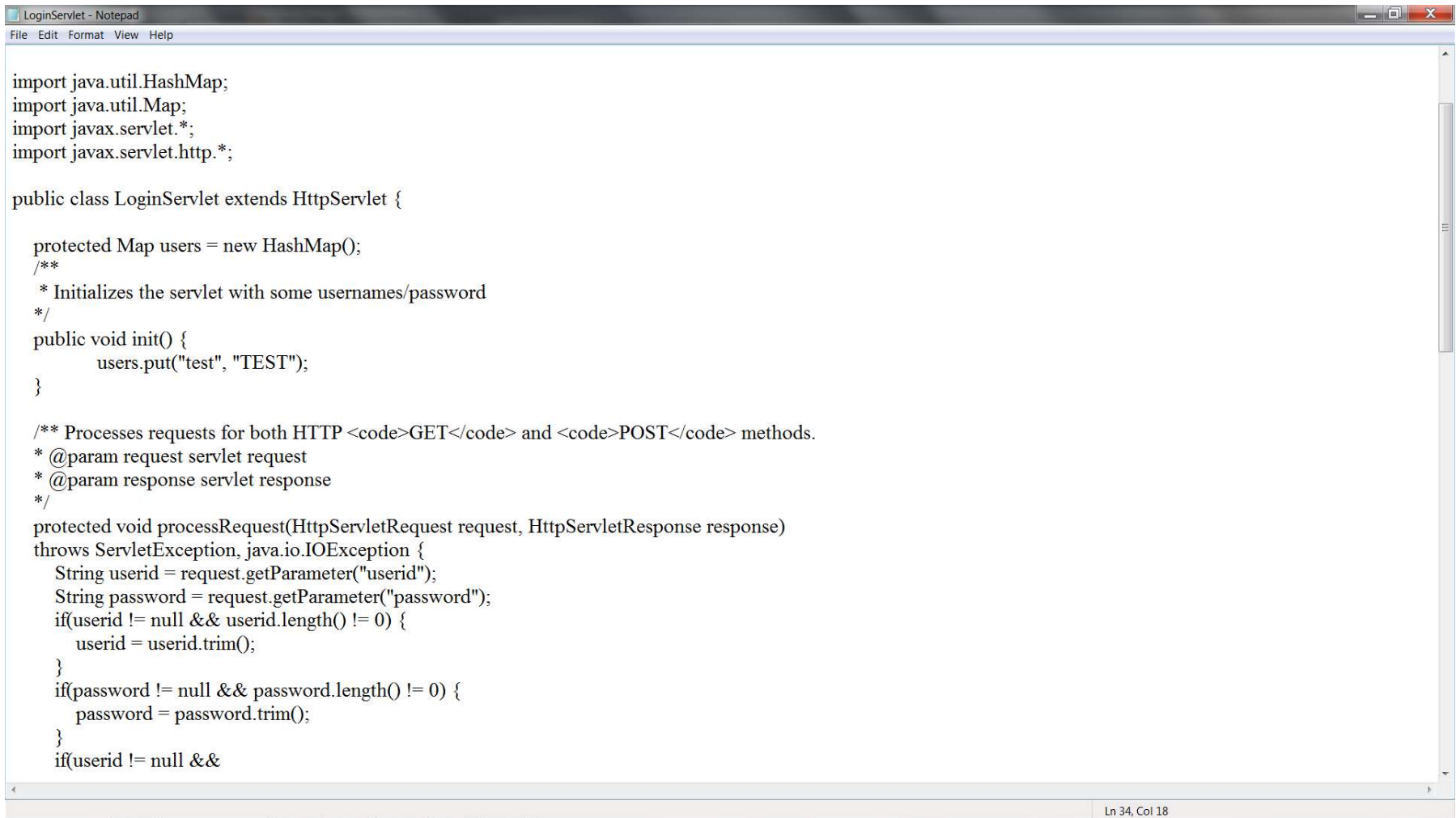
A Login HTML form



login.html

```
<html>
  <head><title>Login Page</title></head>
  <body>
    <form method="post" action="/csj/LoginServlet">
      <h4>Enter your User ID and Password and click Login</h4>
      <table cellpadding='2' cellspacing='1'>
        <tr><td>User ID</td>
          <td><input type="TEXT" size="15"
name="userid"></input></td></tr>
        <tr><td>Password</td>
          <td><input type="PASSWORD" size="15"
name="password"/></td></tr>
        <tr><td colspan='2'>
          <center><input type="SUBMIT" value="Login" /></center>
        </td></tr>
      </table>
    </form>
  </body>
</html>
```

LoginServlet.java



```
import java.util.HashMap;
import java.util.Map;
import javax.servlet.*;
import javax.servlet.http.*;

public class LoginServlet extends HttpServlet {

    protected Map users = new HashMap();
    /**
     * Initializes the servlet with some usernames/password
     */
    public void init() {
        users.put("test", "TEST");
    }

    /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
     * @param request servlet request
     * @param response servlet response
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, java.io.IOException {
        String userid = request.getParameter("userid");
        String password = request.getParameter("password");
        if(userid != null && userid.length() != 0) {
            userid = userid.trim();
        }
        if(password != null && password.length() != 0) {
            password = password.trim();
        }
        if(userid != null &&
```

Ln 34, Col 18

Request Parameters

- A servlet can access the data that is in the form via Request Parameters

`String getParameter (String name)`

- Returns the value of a Request Parameter as a string, or null if the parameter doesn't exist
- Works the same for GET or POST
- The name is case sensitive
- Does the hard work of parsing the QUERY_STRING environment variable for you.

HTML Forms: Radio Buttons

Company Type ☐ Private ☒ Public ☐ Government

- Only one option can be selected at a time

Company Type

```
<input type="radio" name="co_type"
value="private" />Private &nbsp;
<input type="radio" name="co_type"
value="public" checked="true" />Public &nbsp;
<input type="radio" name="co_type"
value="government" />Government
```


HTML Forms: Text Input

First Name

- Allows arbitrary data to be entered
- The amount can be limited, but doesn't prevent the client from submitting more than the `size` in the request

First Name

```
<input type="text" name="first" size="20"/>
```

HTML Forms: Combo Box

State



- Allows single or multiple selection via the multiple option.
- Without multiple, it displays a Combo Box
- With multiple, it displays a Select Box

```
<select name="state">  
  <option value="IL">Illinois</option>  
  <option value="OH">Ohio</option>  
  <option value="IN">Indiana</option>  
  <option value="WI">Wisconsin</option>  
  <option value="MI">Michigan</option>  
</select>
```

HTML Forms: CheckBoxes

Languages Used ☐ Java ☐ C++ ☐ C ☐ Visual Basic ☐ C# ☐ Perl

- Like the Radio buttons, but allows more than one to be selected

```
<input type="checkbox" name="java">Java     
<input type="checkbox" name="cpp">C++     
<input type="checkbox" name="c">C     
<input type="checkbox" name="vb">Visual Basic     
<input type="checkbox" name="csharp">C#     
<input type="checkbox" name="perl">Perl   
```

HTML Forms: TextArea

- Allows for more text to be entered than text inputs, over multiple rows.

Comments

```
<textarea name="comments" rows="2" cols="40">  
</textarea>
```

HTML Forms: Password Fields

- Just like a text input, but will display a '*' instead of the character typed.
- WARNING! – the value of the field is not encrypted. The data is just not displayed to the user. To encrypt your data, use SSL (Secure Sockets Layer).

More ServletRequest methods

`Enumeration` **getParameterNames()**

- Returns an Enumeration of the parameter names in this request

`String[]` **getParameterValues(String name)**

- Returns a String array of values for a given Parameter name. This method is needed if you have parameters with multiple results, like a Select input.
- Returns null if the Parameter doesn't exist

Retrieving all form data

```
Enumeration parameters = request.getParameterNames();
if(parameters.hasMoreElements()) {
    out.println("<tr><th>Parameter Name</th><th>Parameter" +
        "Value</th></tr>");
}
while(parameters.hasMoreElements()) {
    String parameter = (String)parameters.nextElement();
    // get the parameter values
    String[] values = request.getParameterValues(parameter);
    if(values != null) {
        for(int i = 0; i < values.length; i++) {
            out.println("<tr><td><b><fontcolor=\"blue\">" +
                parameter + "</font></b></td>" +
                "<td>" + values[i] + "</td></tr>");
        }
    }
}
```

A Login Servlet

```
import java.util.HashMap;
import java.util.Map;
import javax.servlet.*;
import javax.servlet.http.*;
public class LoginServlet extends HttpServlet {
    protected Map users = new HashMap();
    /**
     * Initializes the servlet with some usernames/passwords
     */
    public void init() {

        users.put("test", "TEST");
    }
}
```


A Login Servlet (cont.)

- Since we want to handle both GET and POST requests, just use the same boilerplate code for `doGet()` and `doPost()`, calling the same `processRequest()` method.

```
protected void doGet(  
    HttpServletRequest request,  
    HttpServletResponse response)  
    throws ServletException, java.io.IOException {  
    processRequest(request, response);  
}
```

A Login Servlet (cont.)

```
protected void processRequest(  
    HttpServletRequest request,  
    HttpServletResponse response)  
    throws ServletException, java.io.IOException {  
    String userid = request.getParameter("userid");  
    String password = request.getParameter("password");  
    if(userid != null && userid.length() != 0) {  
        userid = userid.trim();  
    }  
    if(password != null && password.length() != 0) {  
        password = password.trim();  
    }  
}
```

A Login Servlet (cont.)

- `processRequest()` continued...

```
if(userid != null && password != null) {  
    String realpassword = (String)users.get(userid);  
    if(realpassword != null &&  
        realpassword.equals(password)) {  
        showPage(response, "Login Success!");  
    } else {  
        showPage(response, "Login Failure! Username or"  
            + "password is incorrect");  
    }  
} else {  
    showPage(response, "Login Failure! You must supply a"  
        + "username and password");  
}  
}
```

A Login Servlet (cont.)

- Showing the result

```
protected void showPage(HttpServletResponse response, String
    message)
    throws ServletException, java.io.IOException {
    response.setContentType("text/html");
    java.io.PrintWriter out = response.getWriter();
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Login Servlet Result</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h2>" + message + "</h2>"); out.println("</body>");
    out.println("</html>");
    out.close();
}
```

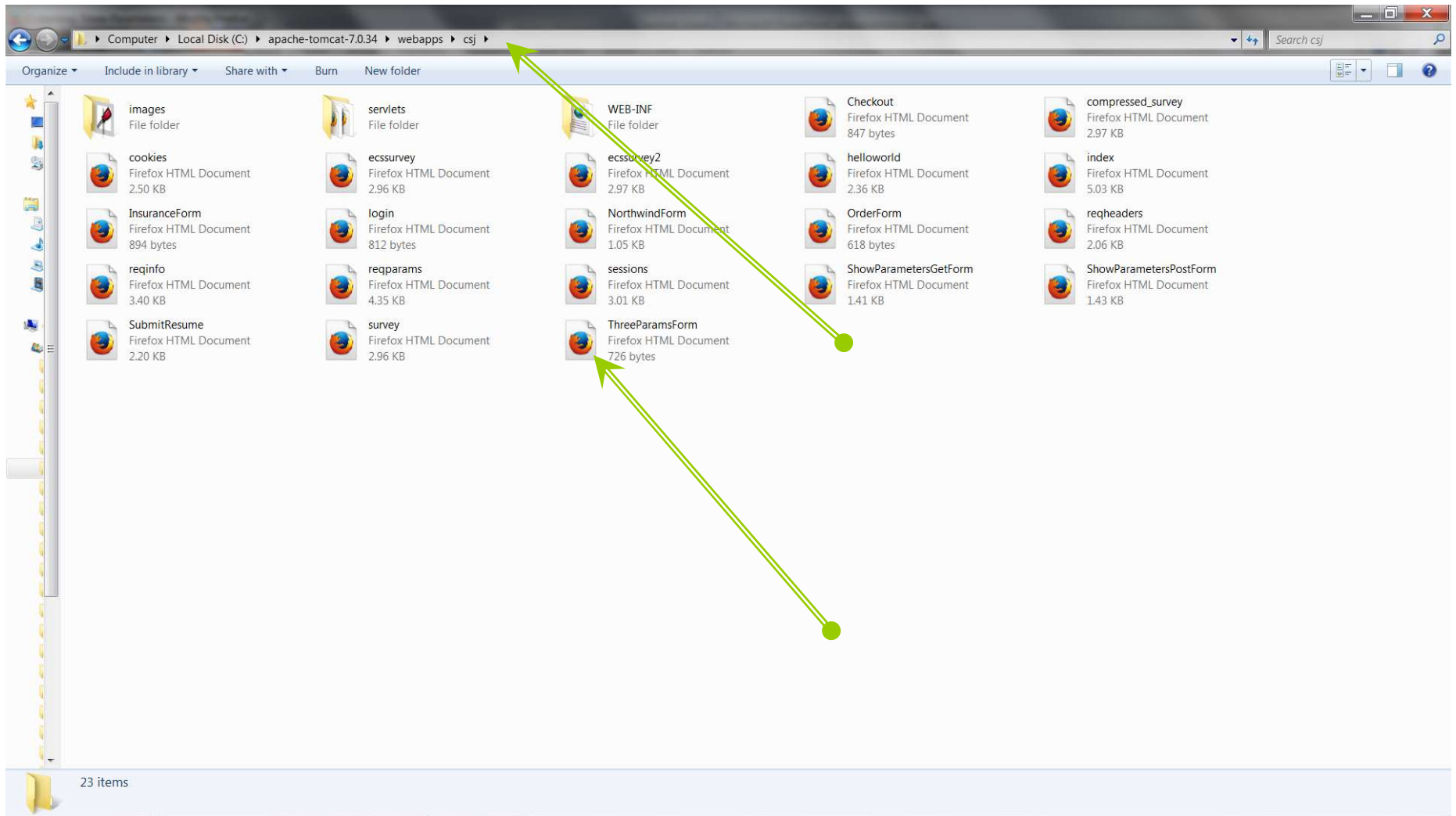
Deploy the Login Servlet

- Compile the code
- Move the LoginServlet.class file to
`<app_home>/WEB-INF/classes/`
- Move the login.html file to `<app_home>`
- Point your browser to:
 - `http://localhost/csj/login.html`
- Try to log in!

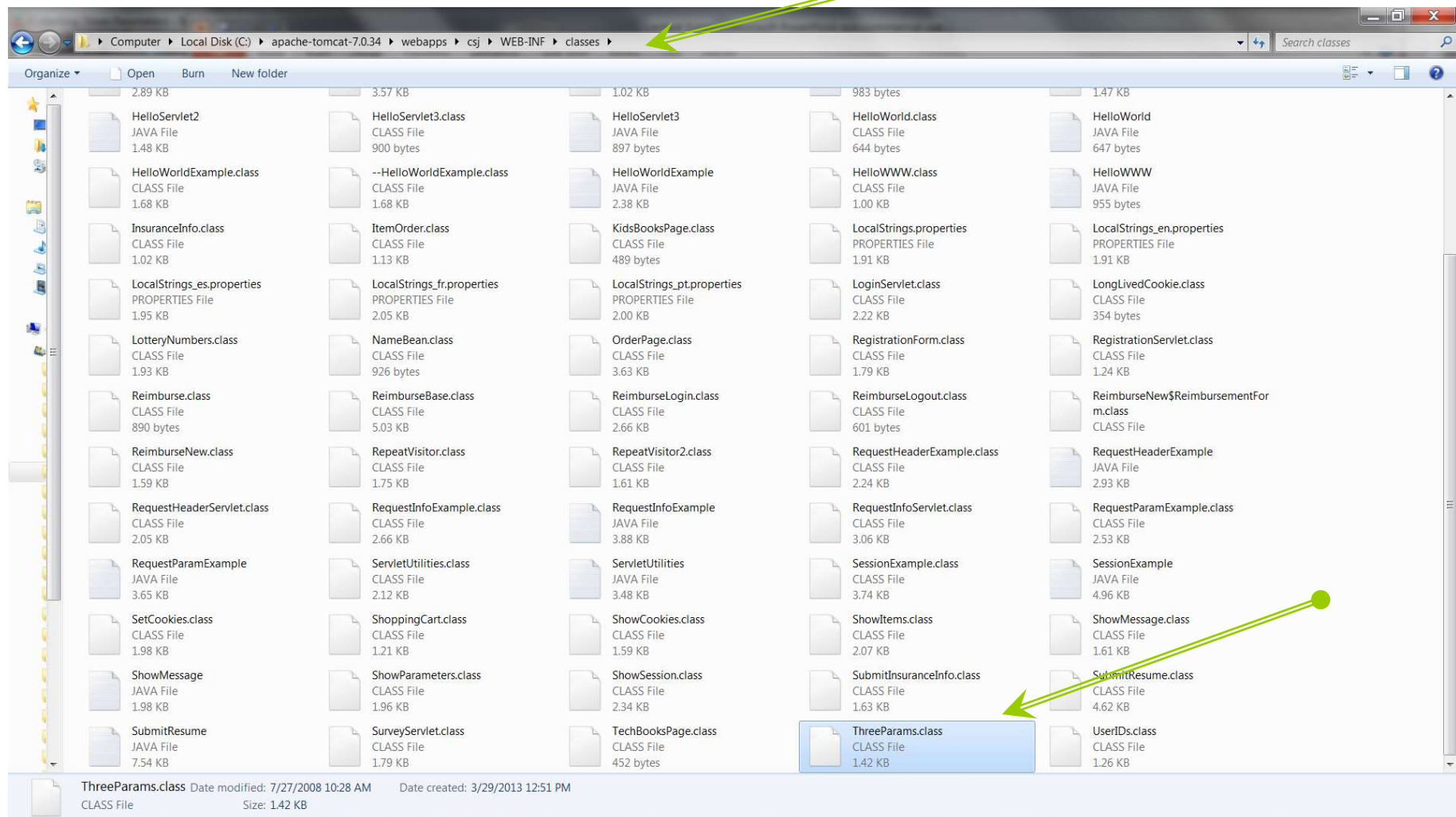
ThreeParams Servlet

- How to read 3 parameters passed from html page?
 - ThreeParams.java
 - ThreeParamsForm.html
 - <http://localhost/cs3/ThreeParamsForm.htm>

ThreeParams Servlet



ThreeParams Servlet



ThreeParams Servlet

Collecting Three Parameters - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/cs/ThreeParamsForm.htm

Google Search

Most Visited Getting Started Latest Headlines

Search Share Sidewiki Bookmarks Translate AutoFill Sign in

Collecting Three Parameters

Collecting Three Parameters

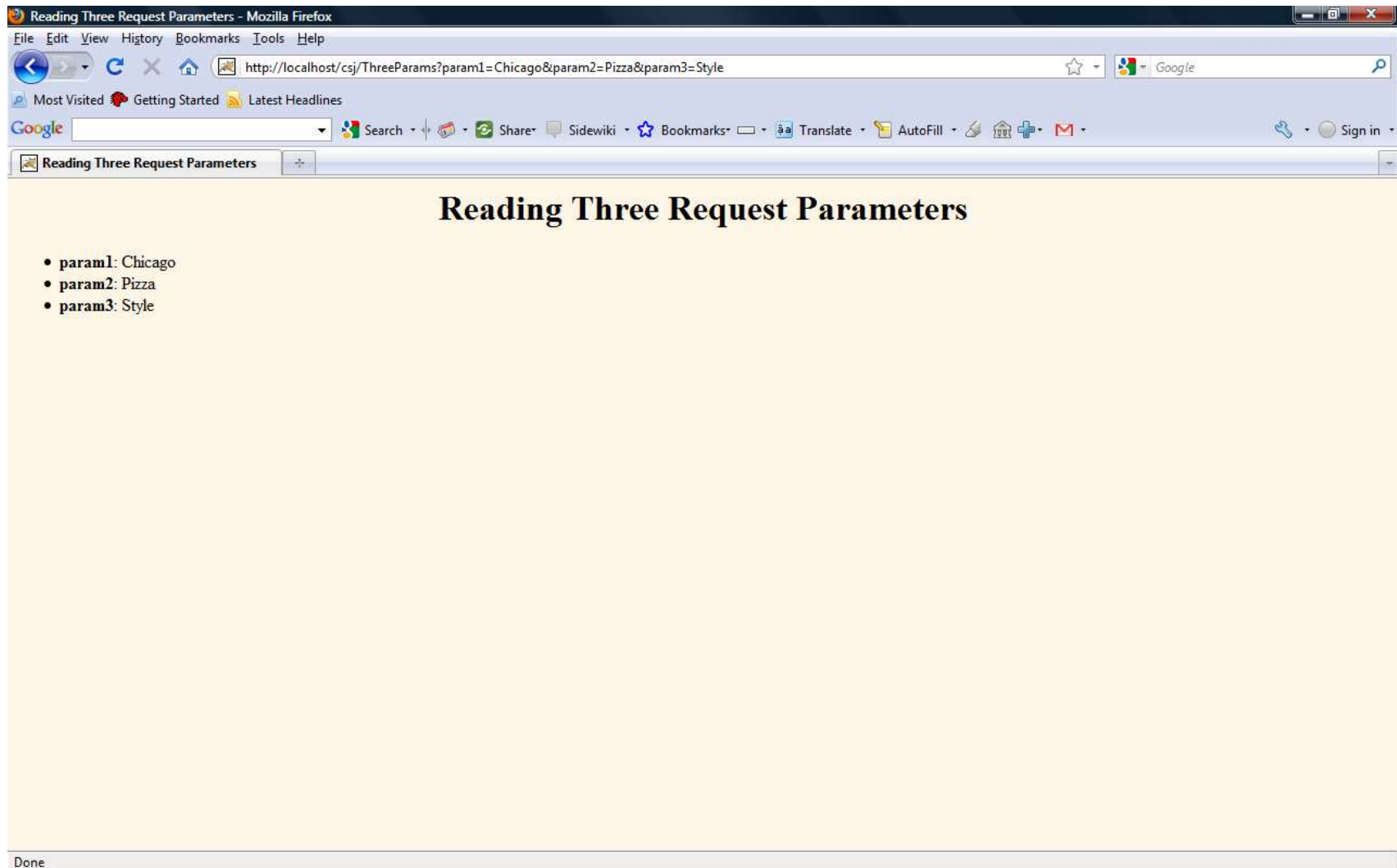
First Parameter:

Second Parameter:

Third Parameter:

Done

ThreeParams Servlet



<http://localhost/csj/ShowParametersGetForm.htm>

- How to read N parameters passed from html page?

- <http://localhost/csj/ShowParametersGetForm.htm>
- [ShowParameters.java](#)

<http://localhost/csj/ShowParametersGetForm.htm>

- We will look at two examples
 - Using Get
 - Using Post

- When using Get in the form ...
 - Watch how the parameters are passed next to the URL

<http://localhost/csj/ShowParametersGetForm.htm>

A Sample FORM using GET - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/csj/ShowParametersGetForm.htm

Most Visited Getting Started Latest Headlines

Google Search Share Sidewiki Bookmarks Translate AutoFill Sign in

A Sample FORM using GET

Item Number: 123
Description: car
Price Each: \$22000

First Name: Tom
Last Name: Rinka
Middle Initial: J

Shipping Address: Adams St.
Chicago, IL

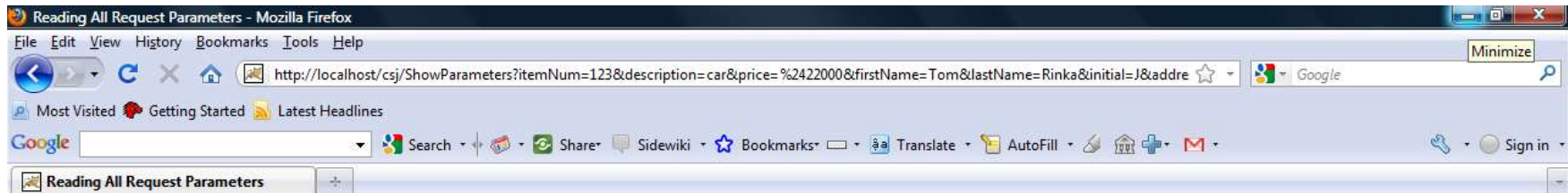
Credit Card:
☒ Visa
☐ MasterCard
☐ American Express
☐ Discover
☐ Java SmartCard

Credit Card Number:
Repeat Credit Card Number:

Submit Order

Done

http://localhost/csj/ShowParametersGetForm.htm



Reading All Request Parameters

Parameter Name	Parameter Value(s)
cardNum	<ul style="list-style-type: none">• 0987654321• 0987654321
cardType	Visa
price	\$22000
initial	J
itemNum	123
address	Adams St. Chicago, IL
firstName	Tom
description	car
lastName	Rinka

<http://localhost/csj/ShowParametersPostForm.htm>

- When using Post in the form ...
 - Watch how the parameters are passed separate from the URL
 - Note that doPost called doGet to avoid repeating the same code

http://localhost/csj/ShowParametersPostForm.htm

A Sample FORM using POST - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/csj/ShowParametersPostForm.htm

Most Visited Getting Started Latest Headlines

Google Search Share Sidewiki Bookmarks Translate AutoFill Sign in

A Sample FORM using POST

Item Number: 123

Description: car

Price Each: \$22000

First Name: Tom

Last Name: Rinka

Middle Initial: J

Chicago, IL

Shipping Address:

Credit Card:

☐ Visa

☒ MasterCard

☐ American Express

☐ Discover

☐ Java SmartCard

Credit Card Number:

Repeat Credit Card Number:

Submit Order

Done

http://localhost/csj/ShowParametersPostForm.htm

Reading All Request Parameters - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/csj/ShowParameters

Most Visited Getting Started Latest Headlines

Google Search Share Sidewiki Bookmarks Translate AutoFill Sign in

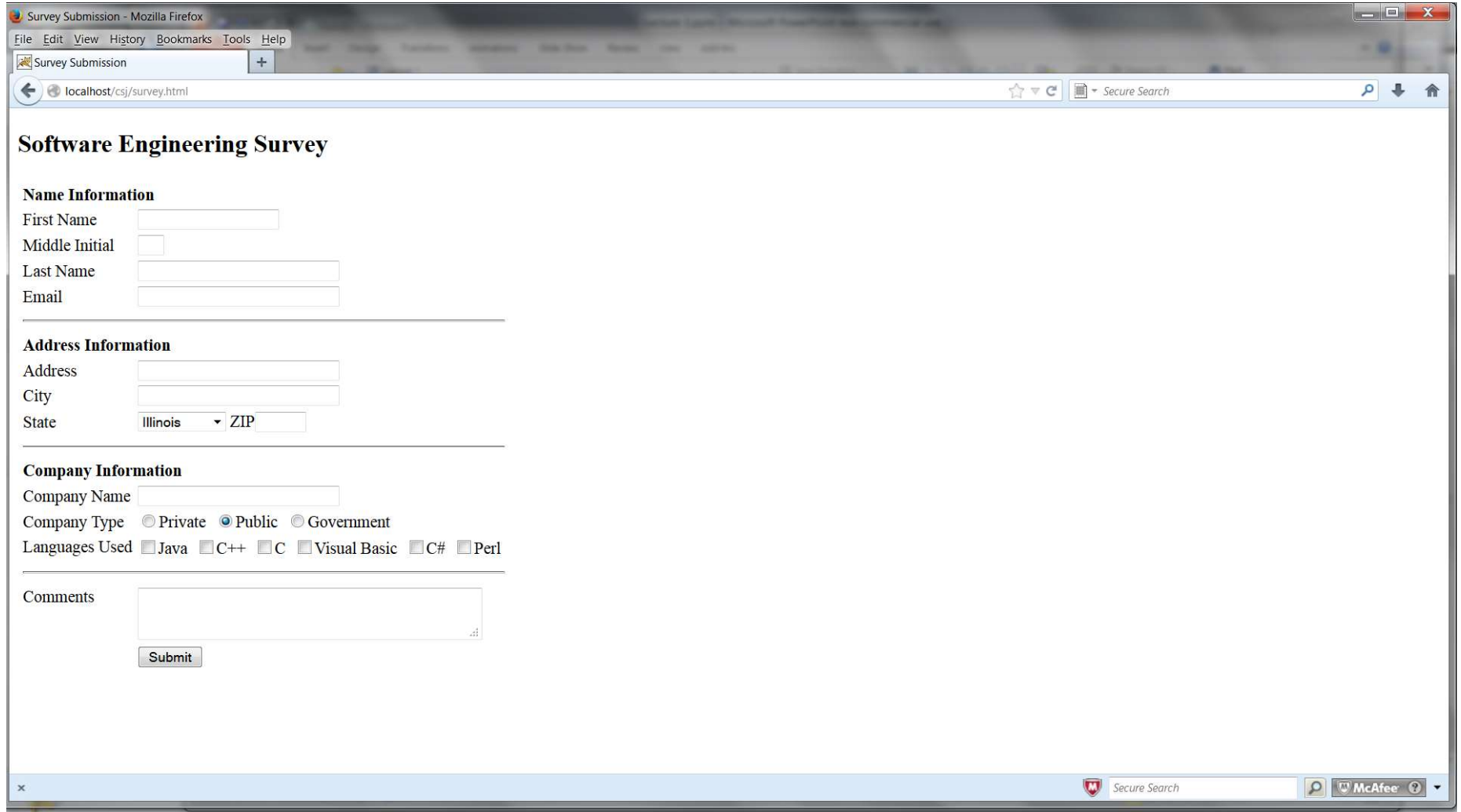
Reading All Request Parameters

Parameter Name	Parameter Value(s)
cardNum	<ul style="list-style-type: none">09876543210987654321
cardType	MasterCard
price	\$22000
initial	J
itemNum	123
address	Chicago, IL
description	car
firstName	Tom
lastName	Rinka

Done

A more advanced Form, A Survey

<http://localhost/csj/survey.html>



The screenshot shows a Mozilla Firefox browser window with the title 'Survey Submission - Mozilla Firefox'. The address bar displays 'localhost/csj/survey.html'. The page content is a survey form titled 'Software Engineering Survey'. The form is divided into four sections: 'Name Information', 'Address Information', 'Company Information', and 'Comments'. The 'Name Information' section has fields for 'First Name', 'Middle Initial', 'Last Name', and 'Email'. The 'Address Information' section has fields for 'Address', 'City', 'State' (a dropdown menu currently showing 'Illinois'), and 'ZIP'. The 'Company Information' section has a 'Company Name' field, 'Company Type' radio buttons (Private, Public, Government) with 'Public' selected, and 'Languages Used' checkboxes (Java, C++, C, Visual Basic, C#, Perl). The 'Comments' section has a large text area and a 'Submit' button. The browser's status bar at the bottom shows 'Secure Search' and 'McAfee'.

Survey Submission - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Survey Submission

localhost/csj/survey.html

Secure Search

Software Engineering Survey

Name Information

First Name

Middle Initial

Last Name

Email

Address Information

Address

City

State ZIP

Company Information

Company Name

Company Type ☐ Private ☒ Public ☐ Government

Languages Used ☐ Java ☐ C++ ☐ C ☐ Visual Basic ☐ C# ☐ Perl

Comments

Submit

Secure Search McAfee

After submitting the form...

Survey Submission - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/csj/survey.html

Most Visited Getting Started Latest Headlines

Google Search Share Sidewiki Bookmarks Translate AutoFill Sign in

Survey Submission

Software Engineering Survey

Name Information

First Name

Middle Initial

Last Name

Email

Address Information

Address

City

State ZIP

Company Information

Company Name

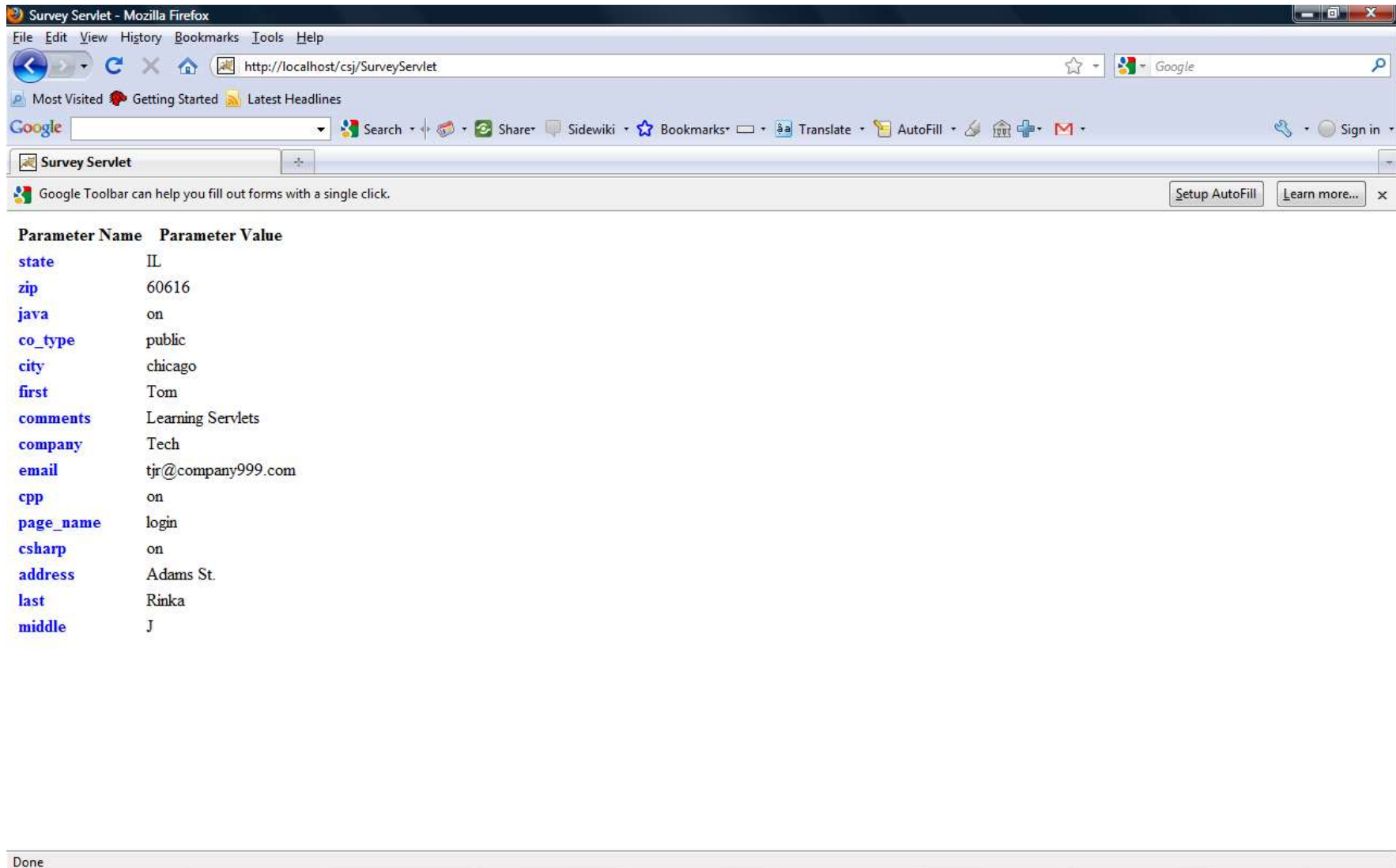
Company Type ☐ Private ☒ Public ☐ Government

Languages Used ☒ Java ☒ C++ ☐ C ☐ Visual Basic ☒ C# ☐ Perl

Comments

Done

After submitting the form...



The screenshot shows a Mozilla Firefox browser window with the title "Survey Servlet - Mozilla Firefox". The address bar displays "http://localhost:csj/SurveyServlet". The browser's toolbar includes navigation buttons (back, forward, home, stop), a search bar with the Google logo, and various utility buttons like "Search", "Share", "Sidewiki", "Bookmarks", "Translate", "AutoFill", and "Sign in". Below the toolbar, a "Survey Servlet" tab is active. The main content area displays a table of parameter names and values. At the bottom of the browser window, a status bar shows the word "Done".

Parameter Name	Parameter Value
state	IL
zip	60616
java	on
co_type	public
city	chicago
first	Tom
comments	Learning Servlets
company	Tech
email	tjr@company999.com
cpp	on
page_name	login
csharp	on
address	Adams St.
last	Rinka
middle	J

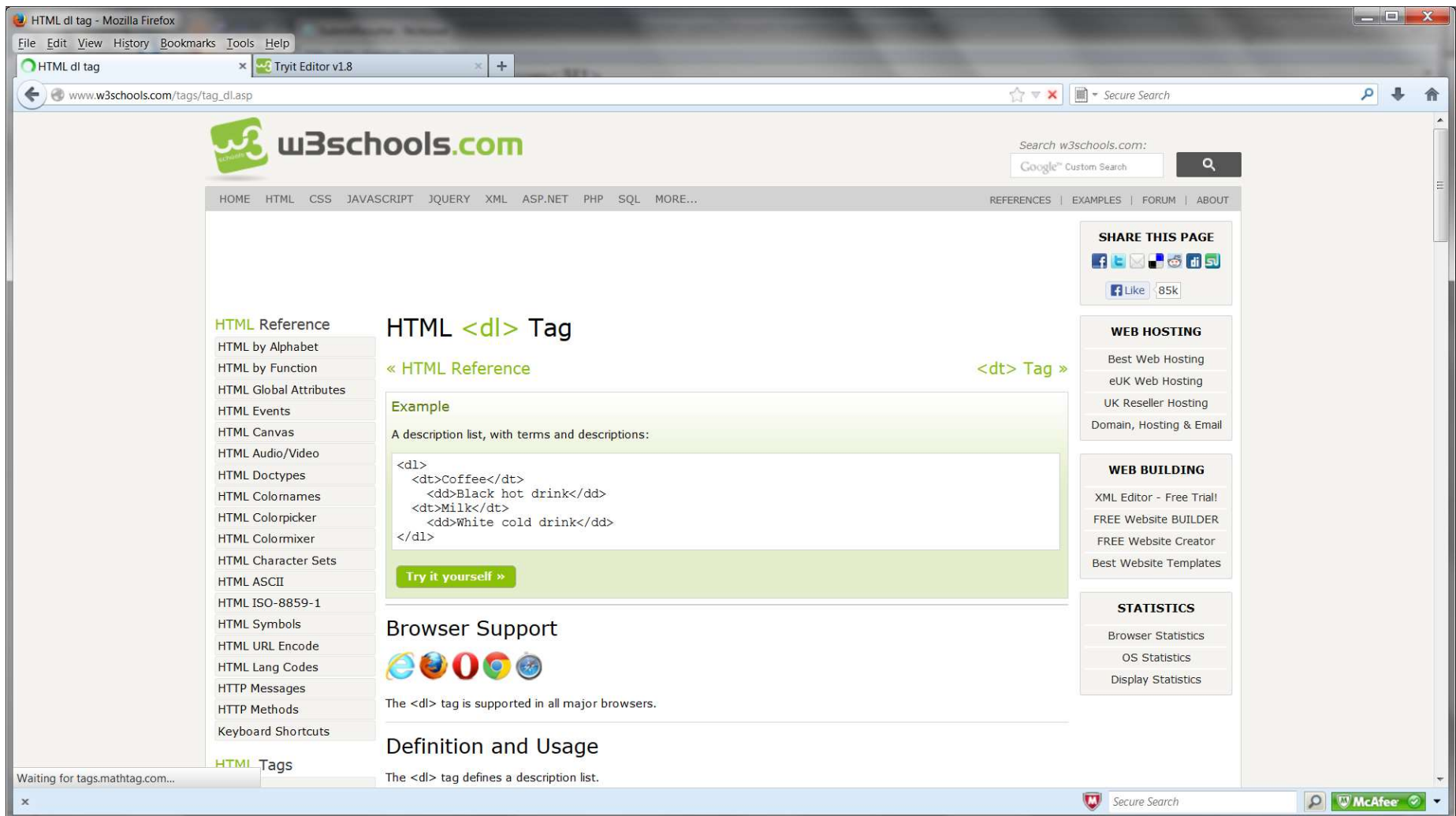
Using Default Values When Parameters are missing or Malformed

- What the servlet should do when dealing with bad input?
 1. Use Default Values
 2. Redisplay the form for the user to type again

Example: Resume Submission

- <http://localhost/csjs/SubmitResume.htm>
- SubmitResume.java Servlet

In case you never used DL, DD, DT HTML tags ...
visit this website to get demo



HTML dl tag - Mozilla Firefox

File Edit View History Bookmarks Tools Help

HTML dl tag x Tryit Editor v1.8

www.w3schools.com/tags/tag_dl.asp

Search w3schools.com: Google™ Custom Search

HOME HTML CSS JAVASCRIPT JQUERY XML ASP.NET PHP SQL MORE... REFERENCES | EXAMPLES | FORUM | ABOUT

HTML Reference

- HTML by Alphabet
- HTML by Function
- HTML Global Attributes
- HTML Events
- HTML Canvas
- HTML Audio/Video
- HTML Doctypes
- HTML Colormnames
- HTML Colorpicker
- HTML Colormixer
- HTML Character Sets
- HTML ASCII
- HTML ISO-8859-1
- HTML Symbols
- HTML URL Encode
- HTML Lang Codes
- HTTP Messages
- HTTP Methods
- Keyboard Shortcuts

HTML <dl> Tag

« HTML Reference <dt> Tag »

Example

A description list, with terms and descriptions:

```
<dl>
  <dt>Coffee</dt>
  <dd>Black hot drink</dd>
  <dt>Milk</dt>
  <dd>White cold drink</dd>
</dl>
```

Try it yourself »

Browser Support

The <dl> tag is supported in all major browsers.

Definition and Usage

The <dl> tag defines a description list.

SHARE THIS PAGE

Like 85k

WEB HOSTING

- Best Web Hosting
- eUK Web Hosting
- UK Reseller Hosting
- Domain, Hosting & Email

WEB BUILDING

- XML Editor - Free Trial!
- FREE Website BUILDER
- FREE Website Creator
- Best Website Templates

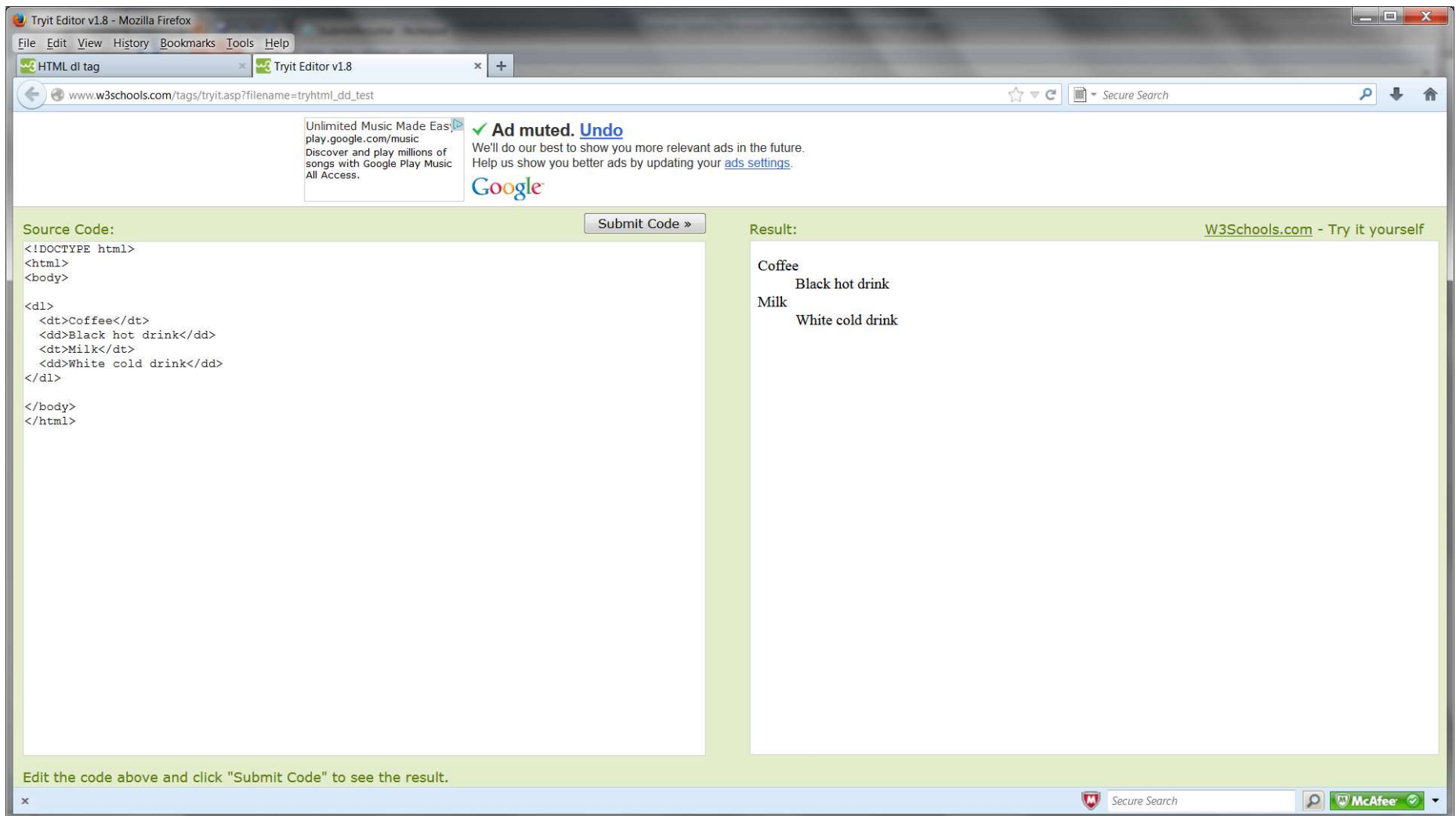
STATISTICS

- Browser Statistics
- OS Statistics
- Display Statistics

Waiting for tags.mathtag.com...

Secure Search McAfee

In case you never used DL, DD, DT HTML tags ...
visit this website to get demo



The screenshot shows a web browser window with the address bar displaying `www.w3schools.com/tags/tryit.asp?filename=tryhtml_dd_test`. The page title is "Tryit Editor v1.8". The main content area is divided into two panels: "Source Code" and "Result".

Source Code:

```
<!DOCTYPE html>
<html>
<body>

<dl>
  <dt>Coffee</dt>
  <dd>Black hot drink</dd>
  <dt>Milk</dt>
  <dd>White cold drink</dd>
</dl>

</body>
</html>
```

Result:

Coffee
Black hot drink
Milk
White cold drink

At the bottom of the editor, a green bar contains the text: "Edit the code above and click 'Submit Code' to see the result."

http://localhost/csj/SubmitResume.htm

Free Resume Posting - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/csj/SubmitResume.htm

Most Visited Getting Started Latest Headlines

Google Search Share Sidewiki Bookmarks Translate AutoFill Sign in

Free Resume Posting

not-computer-jobs.com

To use our *free* resume-posting service, simply fill out the brief summary of your skills below. Use "Preview" to check the results, then press "Submit" once it is ready. Your mini-resume will appear online within 24 hours.

First, give some general information about the look of your resume:

Heading font: default

Heading text size: 32

Body font: default

Body text size: 18

Foreground color: BLACK

Background color: WHITE

Next, give some general information about yourself:

Name: Joe Rao

Current or most recent title: SE

Email address: jrao@seproject.com

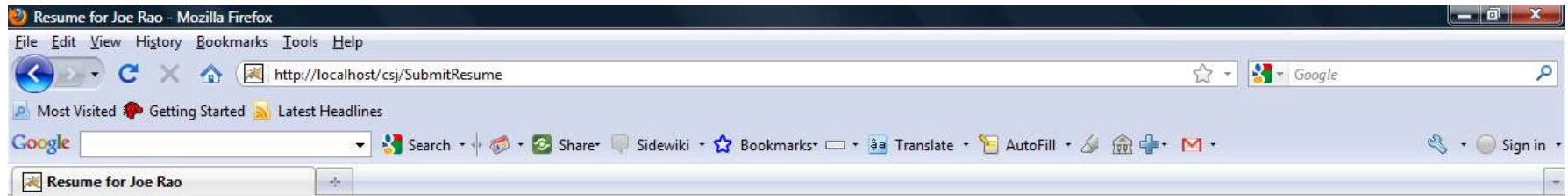
Programming Languages: Java, C#

Finally, enter a brief summary of your skills and experience: (use <P> to separate paragraphs. Other HTML markup is also permitted.)

Building web based QQ application

Done

Once you Click Preview



Joe Rao
SE
jrao@seproject.com

Programming Languages

- Java
- C#

Skills and Experience

Building web based OO application

Once you Click Submit



Submission Confirmed.

Your resume should appear online within 24 hours. If it doesn't, try submitting again with a different email address.

Done

Using BeanUtils to access Request Parameters

- How to automatically populate a bean based on the request parameters?
 - Apache/tomcat has common packages that could be downloaded and automatically populate a bean according to incoming request parameters
- You need to download the following jars from <http://commons.apache.org/>
 - [BeanUtils](#)
 - [Collections](#)
 - [Logging](#)

Using BeanUtils to access Request Parameters

The screenshot shows the Apache Commons website in a Mozilla Firefox browser. The browser's address bar displays 'commons.apache.org'. The website's header features the Apache Commons logo and the URL 'http://commons.apache.org/'. Below the header, a navigation bar includes links for 'Components', 'Sandbox', 'Dormant', 'ApacheCon', and 'Apache'. The main content area is titled 'Welcome to the Apache Commons' and contains the following text:

The Commons is an Apache project focused on all aspects of reusable Java components.

The Apache Commons project is composed of three parts:

- [The Commons Proper](#) - A repository of reusable Java components.
- [The Commons Sandbox](#) - A workspace for Java component development.
- [The Commons Dormant](#) - A repository of components that are currently inactive.

You may also read our [charter](#), which spells out the goals of the project in greater detail.

The Commons Proper

The Commons Proper is dedicated to one principal goal: creating and maintaining reusable Java components. The Commons Proper is a place for collaboration and sharing, where developers from throughout the Apache community can work together on projects to be shared by the Apache projects and Apache users.

Commons developers will make an effort to ensure that their components have minimal dependencies on other libraries, so that these components can be deployed easily. In addition, Commons components will keep their interfaces as stable as possible, so that Apache users (including other Apache projects) can implement these components without having to worry about changes in the future.

This [article](#) gives an overview of (some of) the components which can be found here.

We welcome participation from all that are interested, at all skill levels. Coding, documentation and testing are all critical parts of the software development process. If you are interested in participating in any of these aspects, please join us!

Below the text, there is a table listing various components:

Components	
Attributes	Runtime API to metadata attributes such as doclet tags.
BCEL	Byte Code Engineering Library - analyze, create, and manipulate Java class files
BeanUtils	Easy-to-use wrappers around the Java reflection and introspection APIs.
Betwixt	Services for mapping JavaBeans to XML documents, and vice versa.
BSF	Bean Scripting Framework - interface to scripting languages, including JSR-223
Chain	Chain of Responsibility pattern implementation.
CLI	Command Line arguments parser.
Codas	General encoding/decoding algorithms (for example, hexadecimal, base64, URL).

The left sidebar of the website contains a 'Commons' section with links to 'Home', 'License', 'Mailing Lists', 'PMC', 'Components', 'Sandbox', 'Dormant', 'Releases', and 'Source Repositories'. Below this is a 'General Information' section with links to 'Volunteering', 'Contributing Patches', 'Building Components', 'Site Publication', 'Releasing Components', and 'Wiki'. At the bottom of the sidebar is an 'ASF' section with links to 'How the ASF works', 'Get Involved', 'Developer Resources', 'Sponsorship', and 'Thanks'. The footer of the page includes the 'ApacheCon North America' logo and the text '25th - 28th February 2013 Portland, Oregon'.

Using BeanUtils to access Request Parameters

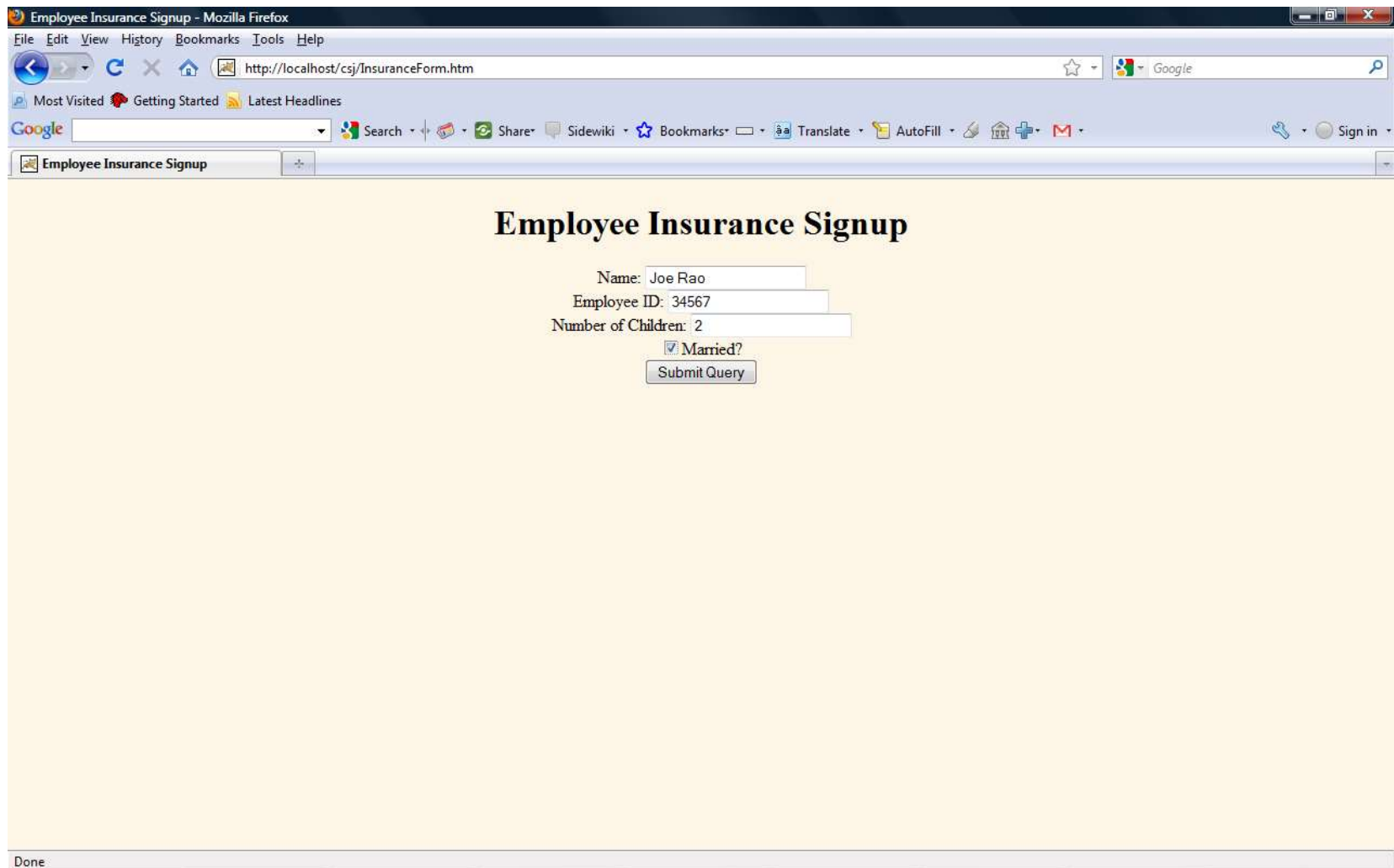
- Once you unzip these files you need to copy the following jar files to:
 - C:\apache-tomcat-7.0.34\lib
 - commons-beanutils-1.8.3
 - commons-beanutils-bean-collections-1.8.3
 - commons-beanutils-core-1.8.3
 - commons-collections-3.2.1
 - commons-logging-1.1.1

Example: BeanUtils to access Request Parameters

- The following example has the following files:
 - InsuranceForm.htm
 - BeanUtilities.java
 - Has a method `BeanUtilities.populateBean` that lets you fill in the required information in a single method call.
 - `BeanUtilities.populateBean` calls `BeanUtils.populate` from `tomcat/apache Common` package
 - InsuranceInfo.java (This is your simple Bean that will be populated `BeanUtils.populate`)
 - SubmitInsuranceInfo.java

Using BeanUtils to access Request Parameters

<http://localhost/csj/InsuranceForm.htm>

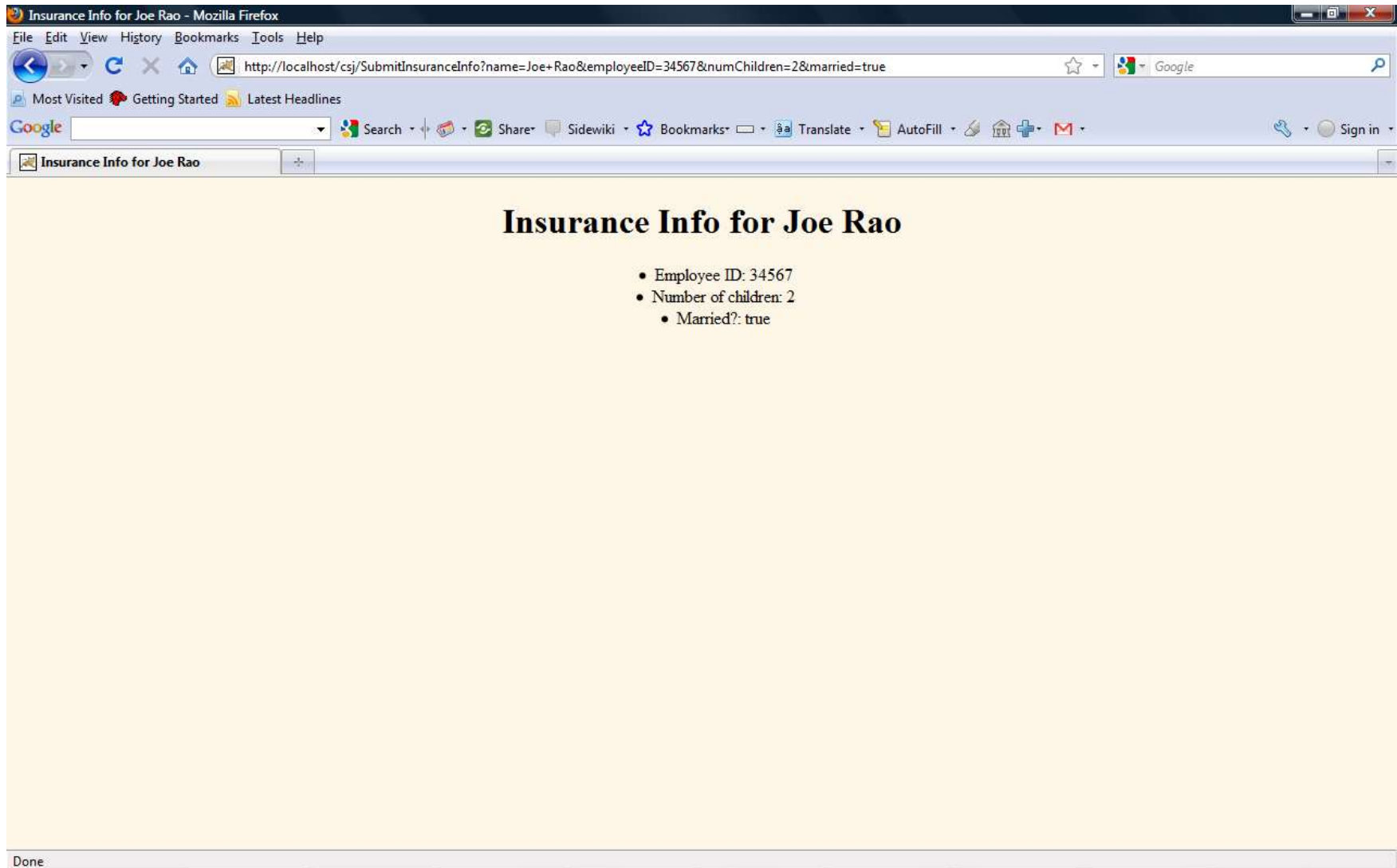


The screenshot shows a Mozilla Firefox browser window with the title 'Employee Insurance Signup - Mozilla Firefox'. The address bar displays 'http://localhost/csj/InsuranceForm.htm'. The browser's toolbar includes various icons for navigation and search. The main content area has a light yellow background and features the title 'Employee Insurance Signup' in a bold, black, serif font. Below the title is a form with the following fields and controls:

- Name: Joe Rao
- Employee ID: 34567
- Number of Children: 2
- ☒ Married?
- Submit Query

The status bar at the bottom of the browser window shows the word 'Done'.

Using BeanUtils to access Request Parameters



HTTP Request Object

GET / HTTP/1.1

Accept: image/gif, image/x-bitmap, image/jpeg, */*

Accept-Language: en-us

Accept-Encoding: gzip, deflate

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0;
Q312461)

Host: localhost

Connection: Keep-Alive

HTTP Request Information

- The class `ServletRequest` contains methods that provide information about the current request
 - `int getLength()`
 - The length, in bytes, of the request body. Returns -1 if length isn't known.
 - `String getProtocol()`
 - The name and version of the protocol the client uses. In the form: *protocol / majorVersion . minorVersion*
e. g., `HTTP/ 1.1` .

HTTP Request Information (cont.)

- `String getRemoteAddr()`
 - The IP address of the client.
- `String getRemoteHost()`
 - The fully qualified host name of the client.
- `boolean isSecure()`
 - Whether this request was made using a secure channel, such as `https`.

HTTP Request Information (cont.)

- In class `HttpServletRequest`
 - `String getAuthType()`
 - The name of the authentication scheme used, e. g., `BASIC` or `SSL` or `null`
 - `String getContextPath()`
 - The portion of the request URI that indicates the context of the request.
 - `String getMethod()`
 - The name of the HTTP request method e. g., `GET`, `POST`, or `PUT`.

HTTP Request Information (cont.)

- `String getPathInfo()`
 - Any extra path information associated with the URL the client sent.
- `String getPathTranslated()`
 - Any extra path information after the servlet name but before the query string, and translates it to a real path.
- `String getQueryString()`
 - The query string that is appended to the request URL after the path.

HTTP Request Information (cont.)

- `String getRemoteUser()`
 - The login of the user making this request, if the user has been authenticated, or `null`.
- `String getRequestURL()`
 - The part of this request's URL from the protocol name up to the query string in the first line of the HTTP request.
- `String getServletPath()`
 - The part of this request's URL that calls the servlet.

Request Information Servlet

```
protected void processRequest(HttpServletRequest request, HttpServletResponse
    response)
    throws ServletException, java.io.IOException {
    String [][] requestInfo = {
        {"Content Length", String.valueOf(request.getContentLength()) },
        {"Content Type", request.getContentType() },
        {"Method" , request.getMethod() },
        {"Authorization Type", request.getAuthType() },
        {"Remote User", request.getRemoteUser() },
        {"Remote Address", request.getRemoteAddr() },
        {"Scheme", request.getScheme() },
        {"Is Secure", String.valueOf(request.isSecure()) },
        {"Protocol", request.getProtocol() },
        {"Context Path", request.getContextPath() },
        {"Path Info", request.getPathInfo() },
        {"Path Translated", request.getPathTranslated() },
        {"Query String", request.getQueryString() },
        {"Servlet Path", request.getServletPath() },
        {"Request URI", request.getRequestURI() }
    };
}
```


Request Headers

- Most of the above Request information is actually stored in a Request Header.
- Each header is one line of the HTTP request, in a **key:value** format.

Common Request Headers

- User-Agent

- Identifies the browser type and version, e. g., Mozilla/ 4.72 [en] (X11; U; Linux 2.2.14- 5.0 i686)

- Host

- Indicates the host given in the request URL, e. g.,
- Required in HTTP 1.1

- Accept

- Indicates MIME types browser can handle, e. g., image/gif, image/jpeg, image/png, */*

- Accept-Encoding

- Indicates encodings browser can handle, e. g., gzip or compress

Common Request Headers (cont.)

- `Connection`
 - `keep-alive` : browser can handle persistent connection.
- `Authorization`
 - User identification for password- protected pages.
- `Cookie`
 - Cookies previously sent to the client by the same server.
- `If-Modified-Since`
 - Send the page only if it has been changed after specified date.
- `Referer`
 - URL of the referring Web page.

Request Headers API

- `String getHeader(String name)`
 - Returns the value of the specified request header as a `String`.
- `Enumeration getHeaderNames()`
 - Returns an enumeration of all the header names this request contains.
- `Enumeration getHeaders(String name)`
 - Returns all the values of the specified request header as an `Enumeration` of `String` objects.

Retrieving Request Headers

```
Enumeration headers = request.getHeaderNames();  
    while(headers.hasMoreElements()) {  
        String headerName = (String)headers.nextElement();  
        String value = request.getHeader(headerName);  
        ...  
    }
```