



# Handling the Client Request: Form Data

**Core Servlets & JSP book: [www.coreservlets.com](http://www.coreservlets.com)**

**More Servlets & JSP book: [www.moreservlets.com](http://www.moreservlets.com)**

**Servlet and JSP Training Courses: [courses.coreservlets.com](http://courses.coreservlets.com)**

**Slides © Marty Hall, <http://www.coreservlets.com>, book © Sun Microsystems Press**

# Agenda

- **Processing form data in servlets**
- **Reading individual request parameters**
- **Reading all request parameters**
- **Real-life servlets: handling malformed data**
- **Filtering HTML-specific characters**

## Aside: Installing forms in eclipse (forms folder in L07 code)

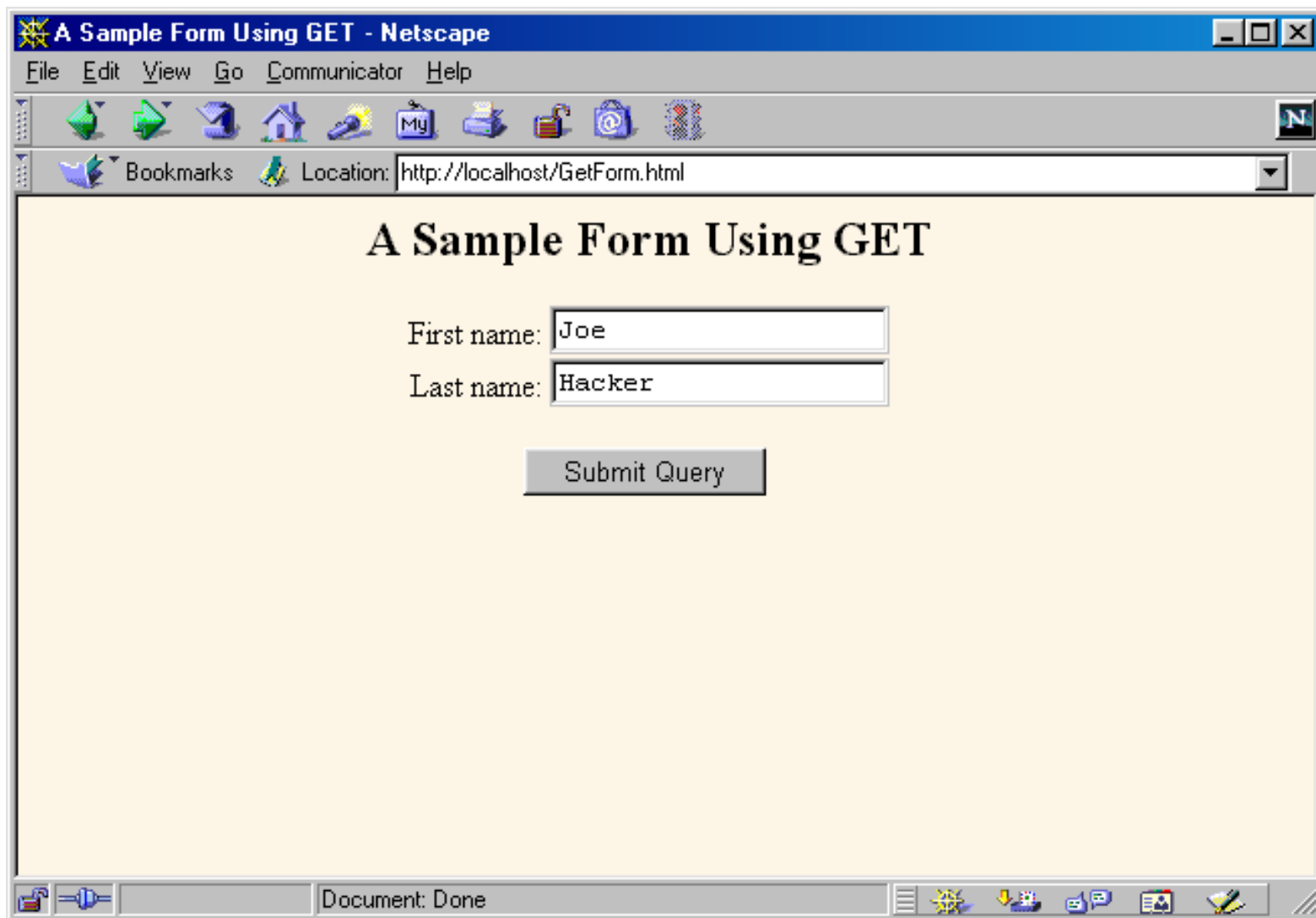
- Create a new dynamic project named “forms”
- Copy HTML files in Web Content folder
- Create the package “coreservlets” in src
- Copy the java files in src->coreservlets
- Run on a server

# Creating Form Data: HTML Forms

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>A Sample Form Using GET</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<H2 ALIGN="CENTER">A Sample Form Using GET</H2>

<FORM ACTION="http://localhost:8088/SomeProgram">
  <CENTER>
    First name:
    <INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
    Last name:
    <INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
    <INPUT TYPE="SUBMIT"> <!-- Press this to submit form -->
  </CENTER>
</FORM>
</BODY></HTML>
```

# HTML Form: Initial Result



The screenshot shows a Netscape browser window titled "A Sample Form Using GET - Netscape". The address bar displays "http://localhost/GetForm.html". The main content area contains the following HTML form:

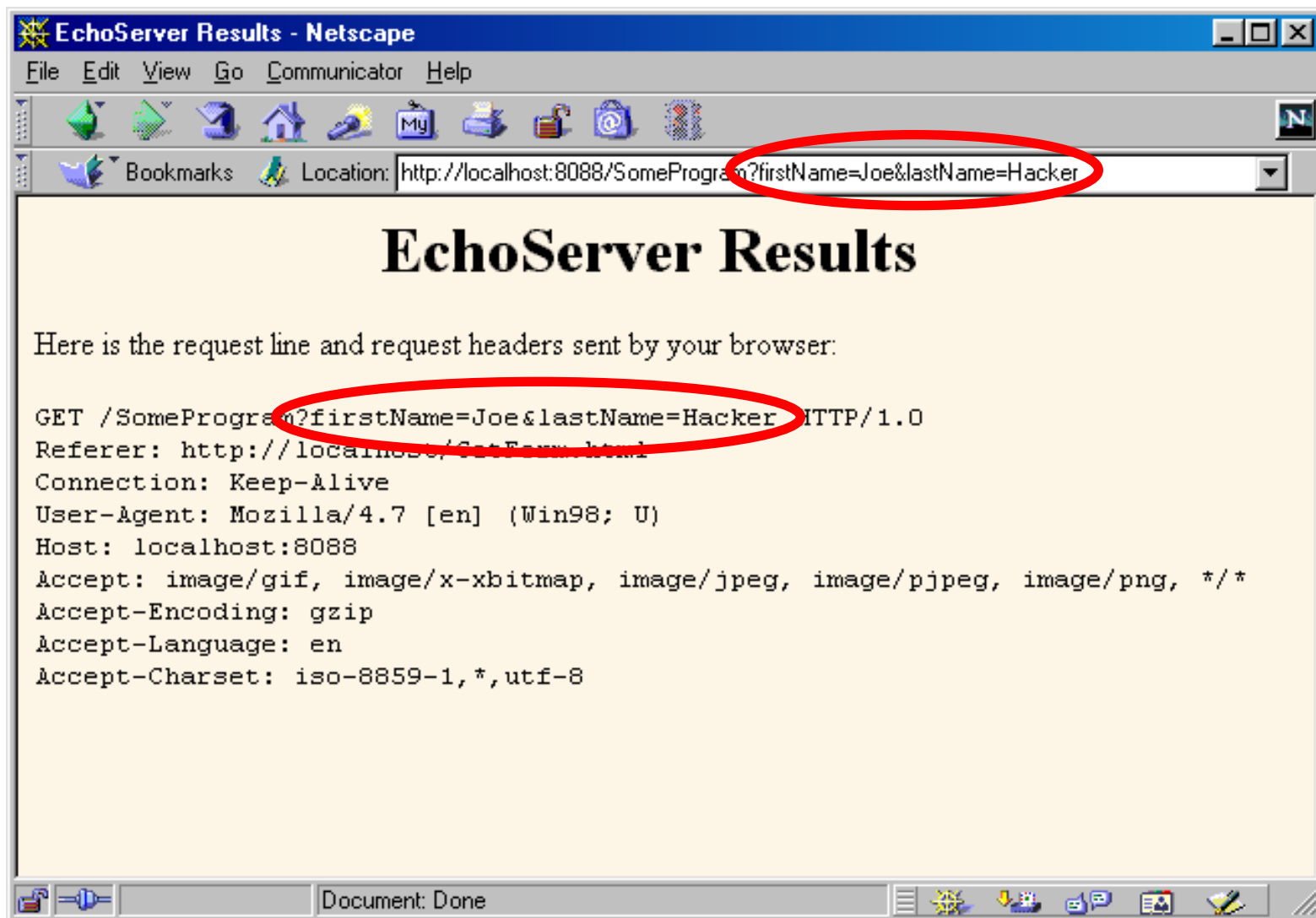
**A Sample Form Using GET**

First name:

Last name:

The status bar at the bottom indicates "Document: Done".

# HTML Form: Submission Result (Data Sent to EchoServer)





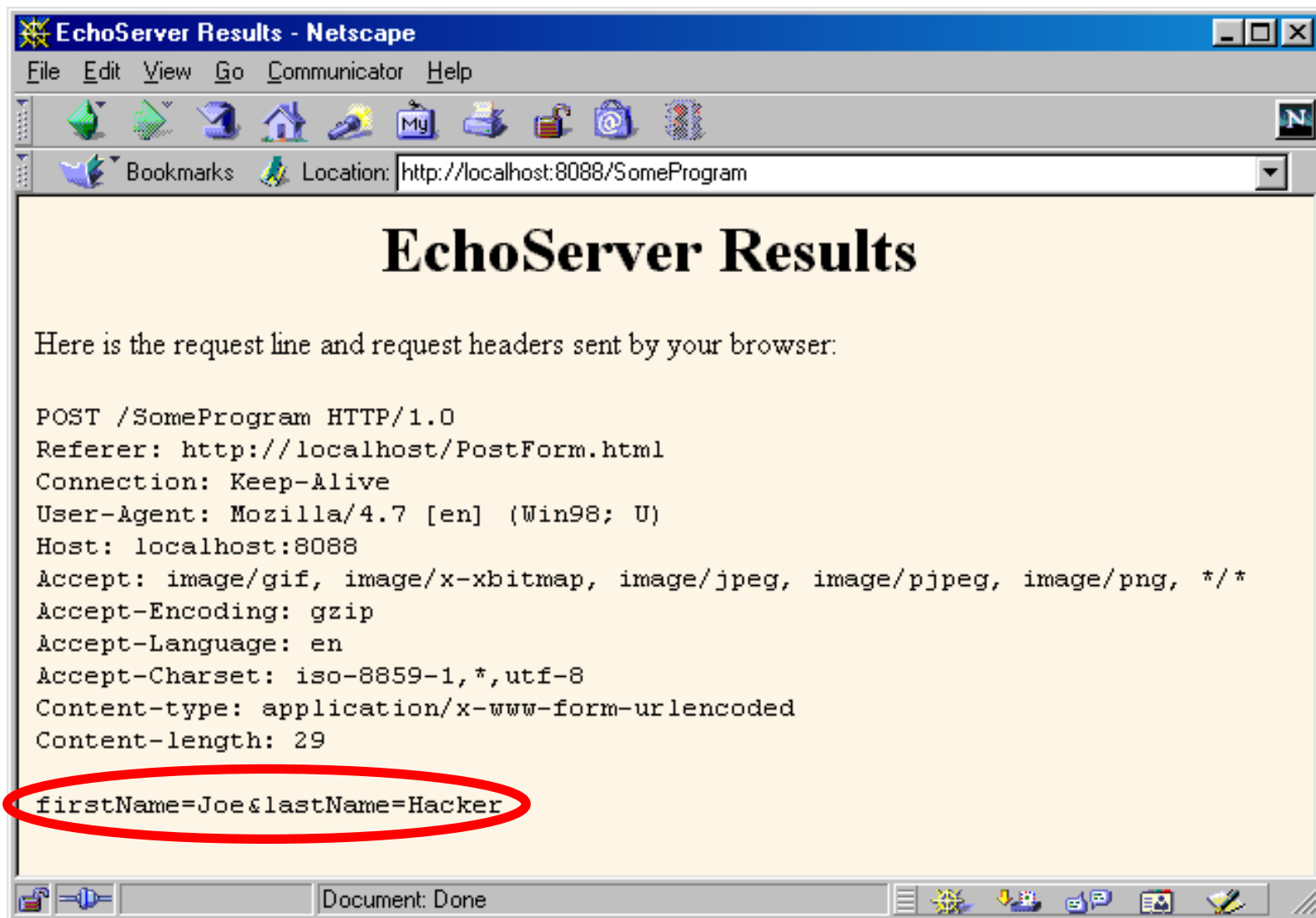
# Sending POST Data

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>A Sample Form Using POST</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<H2 ALIGN="CENTER">A Sample Form Using POST</H2>

<FORM ACTION="http://localhost:8088/SomeProgram"
      METHOD="POST">
  <CENTER>
    First name:
    <INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
    Last name:
    <INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
    <INPUT TYPE="SUBMIT">
  </CENTER>
</FORM>

</BODY></HTML>
```

# Sending POST Data





# Reading Form Data In Servlets

- **request.getParameter("name")**
  - Returns URL-decoded value of **first occurrence of name** in query string
  - Works identically for GET and POST requests
  - Returns null if no such parameter is in query
- **request.getParameterValues("name")**
  - Returns an array of the URL-decoded values of **all occurrences of name** in query string
  - Returns a one-element array if param not repeated
  - Returns null if no such parameter is in query
- **request.getParameterNames()**
  - Returns Enumeration of request params

# Handling Input in Multiple Languages

- **Use server's default character set**

```
String firstName =  
    request.getParameter("firstName");
```

- **Convert from English (Latin-1) to Japanese**

```
String firstNameWrongEncoding =  
    request.getParameter("firstName");  
String firstName =  
    new String(firstNameWrongEncoding.getBytes(),  
        "Shift_JIS");
```

- **Accept either English or Japanese**

```
request.setCharacterEncoding("JISAutoDetect");  
String firstName =  
    request.getParameter("firstName");
```

# An HTML Form With Three Parameters (three-params-form.html)

```
<FORM ACTION="/servlet/coreservlets.ThreeParams">
```

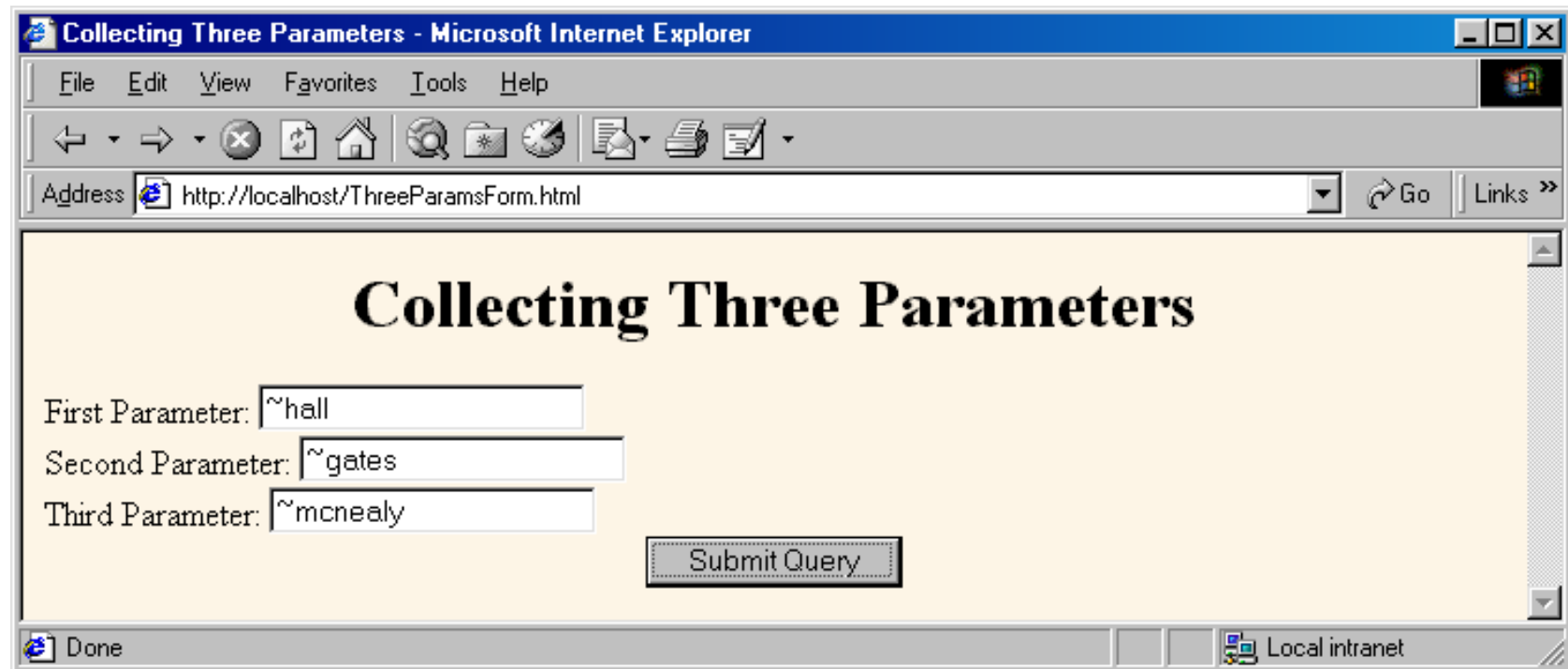
```
  First Parameter:  <INPUT TYPE="TEXT" NAME="param1"><BR>
```

```
  Second Parameter: <INPUT TYPE="TEXT" NAME="param2"><BR>
```

```
  Third Parameter:  <INPUT TYPE="TEXT" NAME="param3"><BR>
```

```
  <CENTER><INPUT TYPE="SUBMIT"></CENTER>
```

```
</FORM>
```

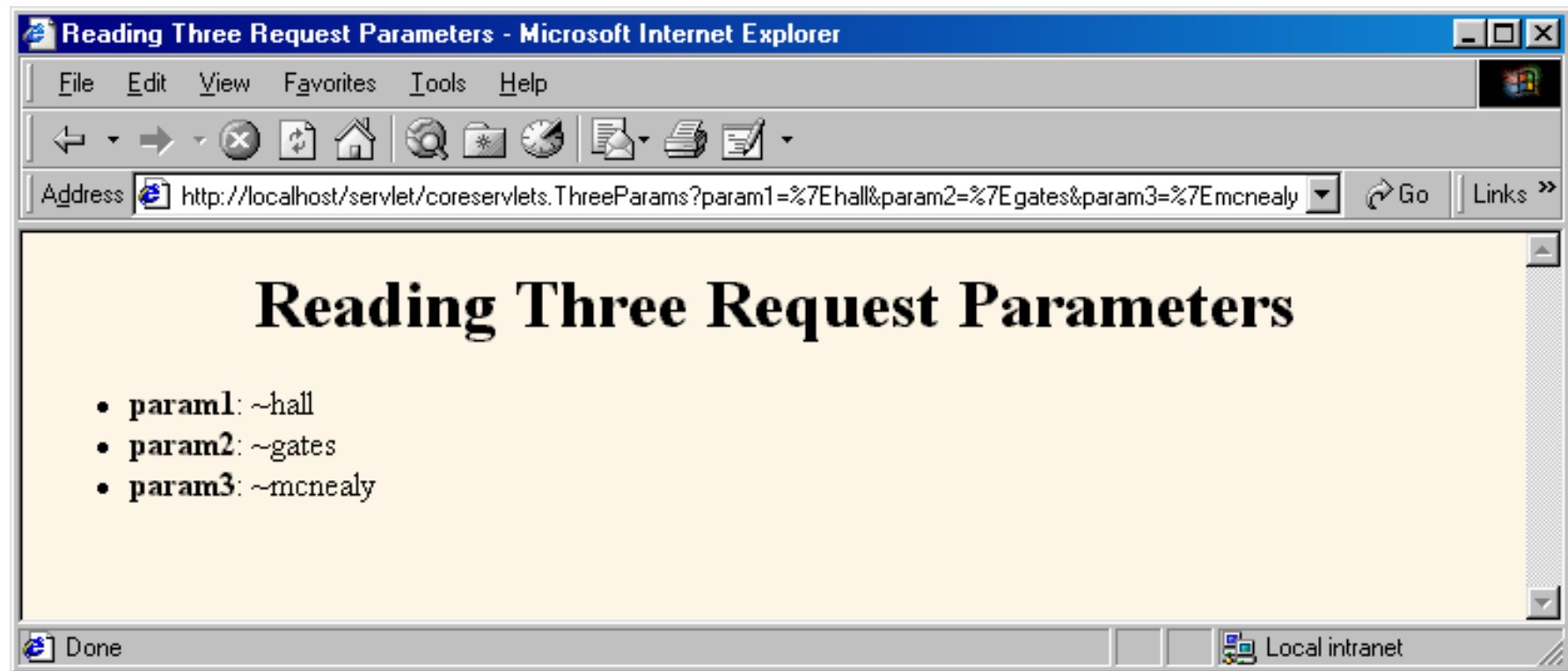


The screenshot shows a Microsoft Internet Explorer window titled "Collecting Three Parameters - Microsoft Internet Explorer". The address bar shows "http://localhost/ThreeParamsForm.html". The page content features a title "Collecting Three Parameters" in a large, bold, serif font. Below the title, there are three text input fields labeled "First Parameter:", "Second Parameter:", and "Third Parameter:". The first field contains the text "~hall", the second contains "~gates", and the third contains "~mcnealy". Below these fields is a button labeled "Submit Query". The browser's status bar at the bottom shows "Done" and "Local intranet".

# Reading the Three Parameters (ThreeParams.java)

```
public class ThreeParams extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String title = "Reading Three Request Parameters";
        out.println(ServletUtilities.headWithTitle(title) +
            "<BODY BGCOLOR=\"#FDF5E6\">\n" +
            "<H1 ALIGN=CENTER>" + title + "</H1>\n" +
            "<UL>\n" +
            "    <LI><B>param1</B>: "
            + request.getParameter("param1") + "\n" +
            "    <LI><B>param2</B>: "
            + request.getParameter("param2") + "\n" +
            "    <LI><B>param3</B>: "
            + request.getParameter("param3") + "\n" +
            "</UL>\n" +
            "</BODY></HTML>");
    }
}
```

# Reading Three Parameters: Result



# Reading All Parameters (show-parameters servlet)

```
public class ShowParameters extends HttpServlet {  
    public void doGet(HttpServletRequest request,  
                      HttpServletResponse response)  
        throws ServletException, IOException {  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();  
        String title = "Reading All Request Parameters";  
        out.println(ServletUtilities.headWithTitle(title) +  
                    "<BODY BGCOLOR=\"#FDF5E6\">\n" +  
                    "<H1 ALIGN=CENTER>" + title + "</H1>\n" +  
                    "<TABLE BORDER=1 ALIGN=CENTER>\n" +  
                    "<TR BGCOLOR=\"#FFAD00\">\n" +  
                    "<TH>Parameter Name<TH>Parameter Value(s)");  
    }  
}
```



# Reading All Parameters (Continued)

```
Enumeration paramNames = request.getParameterNames();
while(paramNames.hasMoreElements()) {
    String paramName = (String)paramNames.nextElement();
    out.print("<TR><TD>" + paramName + "\n<TD>");
    String[] paramValues =
        request.getParameterValues(paramName);
    if (paramValues.length == 1) {
        String paramValue = paramValues[0];
        if (paramValue.length() == 0)
            out.println("<I>No Value</I>");
        else
            out.println(paramValue);
    }
}
```

# Reading All Parameters (Continued)

```
        } else {
            out.println("<UL>");
            for(int i=0; i<paramValues.length; i++) {
                out.println("<LI>" + paramValues[i]);
            }
            out.println("</UL>");
        }
    }
    out.println("</TABLE>\n</BODY></HTML>");
}

public void doPost(HttpServletRequest request,
                    HttpServletResponse response)
    throws ServletException, IOException {
    doGet(request, response);
}
}
```

# Result of ShowParameters Servlet

**A Sample FORM using POST**

Item Number: 127A  
Quantity: 12  
Price Each: \$4.95

First Name: Marty  
Last Name: Hall  
Middle Initial:

Shipping Address: Johns Hopkins Applied Physics Lab  
11100 Johns Hopkins Rd.  
Laurel, MD 20723

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

Credit Card Number: \*\*\*\*\*  
Repeat Credit Card Number: \*\*\*\*\*

Submit Order

**Reading All Request Parameters**

Parameter Name	Parameter Value(s)
address	Johns Hopkins Applied Physics Lab 11100 Johns Hopkins Rd. Laurel, MD 20723
initial	No Value
price	\$4.95
cardNum	<ul style="list-style-type: none"><li>• 3.14159</li><li>• 3.14159</li></ul>
firstName	Marty
itemNum	127A
cardType	Java SmartCard
quantity	12
lastName	Hall

- Note that order of parameters in Enumeration does not match order they appeared in Web page

# Posting Service: Front End (submit-resume.html)

- Gathers resumé formatting and content information
- (submit-resume.html
- Show-resume-preview.java)

Free Resume Posting - Microsoft Internet Explorer

File Edit View Favorites Tools Help

hotcomputerjobs.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 on-line 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: Al Gore lthm

Current or most recent title: Chief Technology Officer

Email address: lthm@aol.com

Programming Languages: Java, C++, Smalltalk, Ada

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

<P>Expert in data structures and computational methods.</P><P>Well known for finding efficient solutions to <I>apparently</I> intractable problems, then rigorously proving time and memory requirements for best, worst, and average-case performance.</P><P>Can prove that P is not equal to NP. Doesn't want to work for companies that don't know what this means.</P><P>Not related to the American politician.</P>

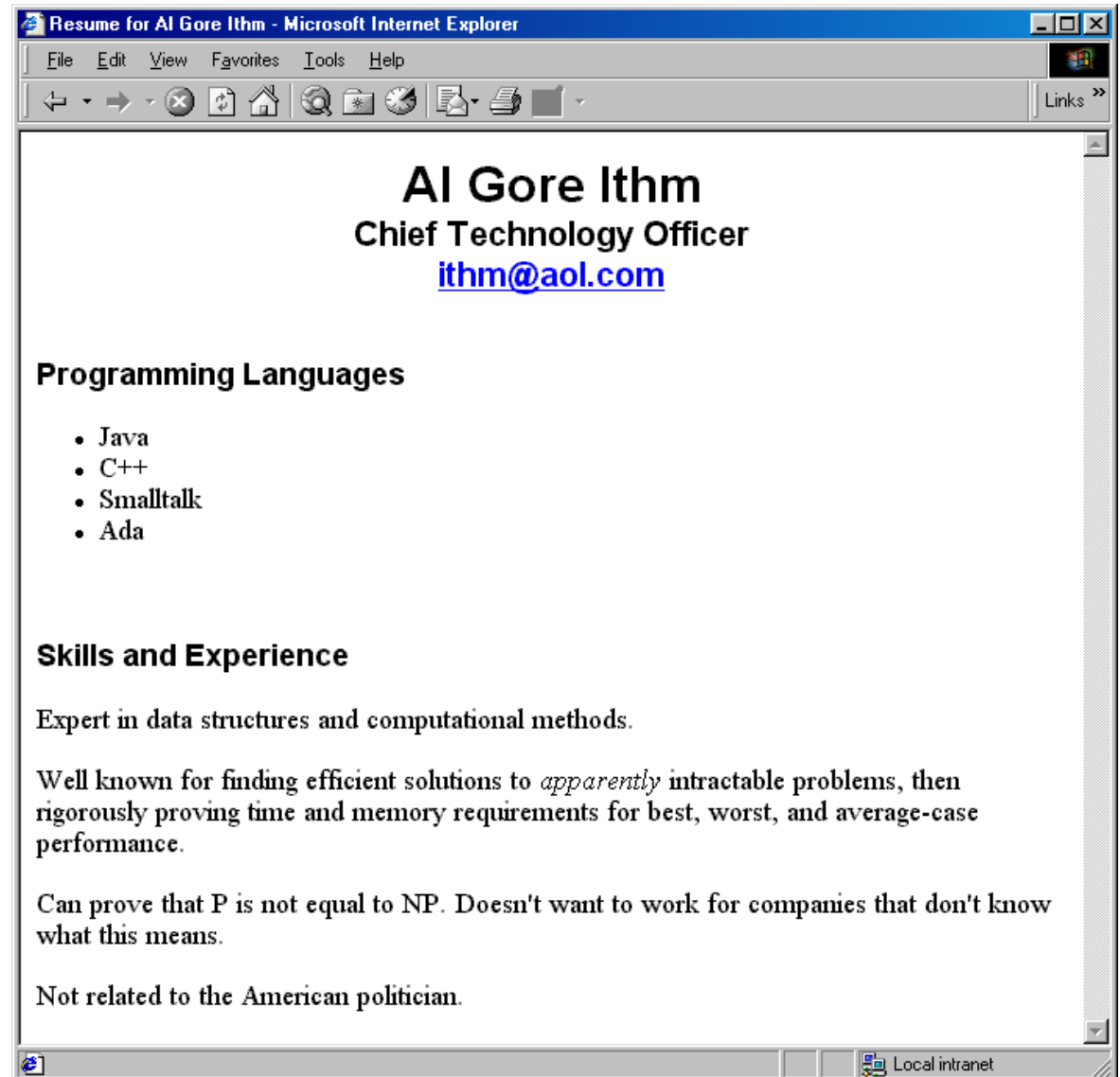
Preview Submit

See our privacy policy [here](#).

Done Local intranet

# Posting Service: Back End

- **Previews result or stores resumé in database**



# Missing data

- **What should the servlet do when the user fails to supply the necessary information?**
- **This question has two answers:**
- **use default values**
- **redisplay the form (prompting the user for missing values).**



# Point: Check for Missing Data

- **Textfield was not in HTML form at all**
  - request.getParameter returns **null**
- **Textfield was empty when form was submitted**
  - Request.getParameter returns an **empty String**
- **Example Check**

```
String value = request.getParameter("someName");  
if ((value != null) && (!value.equals("")) ) {  
    ...  
}
```

# Posting Service: Servlet Code

```
private void showPreview(HttpServletRequest request,
                        PrintWriter out) {
    String headingFont = request.getParameter("headingFont");
    headingFont = replaceIfMissingOrDefault(headingFont, "");
    ...
    String name = request.getParameter("name");
    name = replaceIfMissing(name, "Lou Zer");
    String title = request.getParameter("title");
    title = replaceIfMissing(title, "Loser");
    String languages = request.getParameter("languages");
    languages = replaceIfMissing(languages, "<I>None</I>");
    String languageList = makeList(languages);
    String skills = request.getParameter("skills");
    skills = replaceIfMissing(skills, "Not many, obviously.");
    ...
}
```

- **Point: always explicitly handle missing or malformed query data**

# Filtering Strings for HTML-Specific Characters

- **You cannot safely insert arbitrary strings into servlet output**
  - < and > can cause problems anywhere
  - & and " can cause problems inside of HTML attributes
- **You sometimes cannot manually translate**
  - The string is derived from a program excerpt or another source where it is already in some standard format
  - The string is derived from HTML form data
- **Failing to filter special characters from form data makes you vulnerable to *cross-site scripting attack***
  - <http://www.cert.org/advisories/CA-2000-02.html>
  - <http://www.microsoft.com/technet/security/crssite.asp>

# Filtering Code (ServletUtilities.java)

```
public static String filter(String input) {  
    StringBuffer filtered = new StringBuffer(input.length());  
    char c;  
    for(int i=0; i<input.length(); i++) {  
        c = input.charAt(i);  
        if (c == '<') {  
            filtered.append("&lt;");  
        } else if (c == '>') {  
            filtered.append("&gt;");  
        } else if (c == '\"') {  
            filtered.append("&quot;");  
        } else if (c == '&') {  
            filtered.append("&amp;");  
        } else {  
            filtered.append(c);  
        }  
    }  
    return(filtered.toString());  
}
```

# Servlet That Fails to Filter

```
public class BadCodeServlet extends HttpServlet {
    private String codeFragment =
        "if (a<b) {\n" +
        "    doThis();\n" +
        "} else {\n" +
        "    doThat();\n" +
        "}\n";

    public String getCodeFragment() {
        return (codeFragment);
    }
}
```

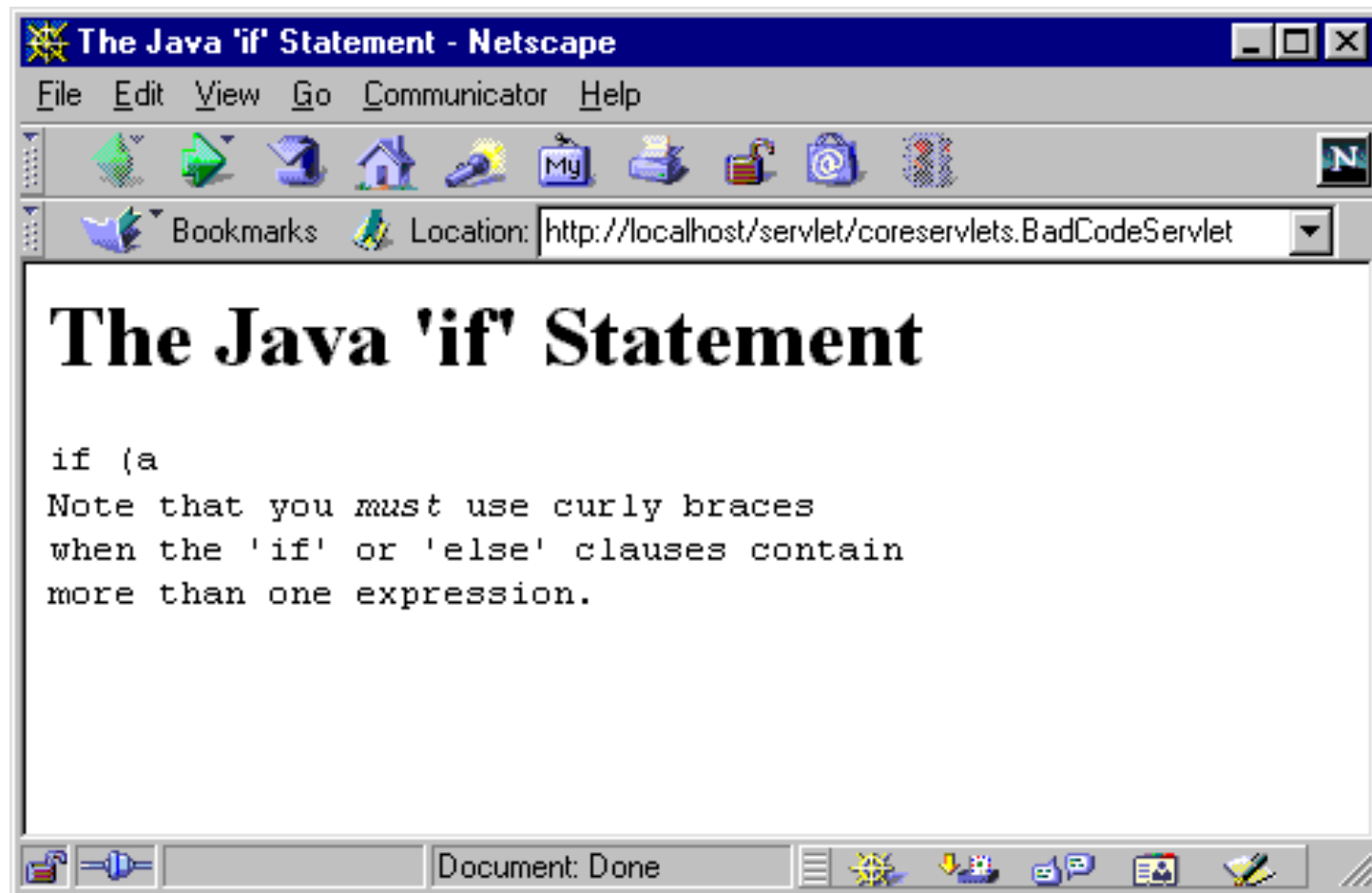
# Servlet That Fails to Filter (Continued)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String title = "The Java 'if' Statement";

    out.println(ServletUtilities.headWithTitle(title) +
                "<BODY>\n" +
                "<H1>" + title + "</H1>\n" +
                "<PRE>\n" +
                getCodeFragment() +
                "</PRE>\n" +
                "Note that you <I>must</I> use curly braces\n" +
                "when the 'if' or 'else' clauses contain\n" +
                "more than one expression.\n" +
                "</BODY></HTML>");
}
```

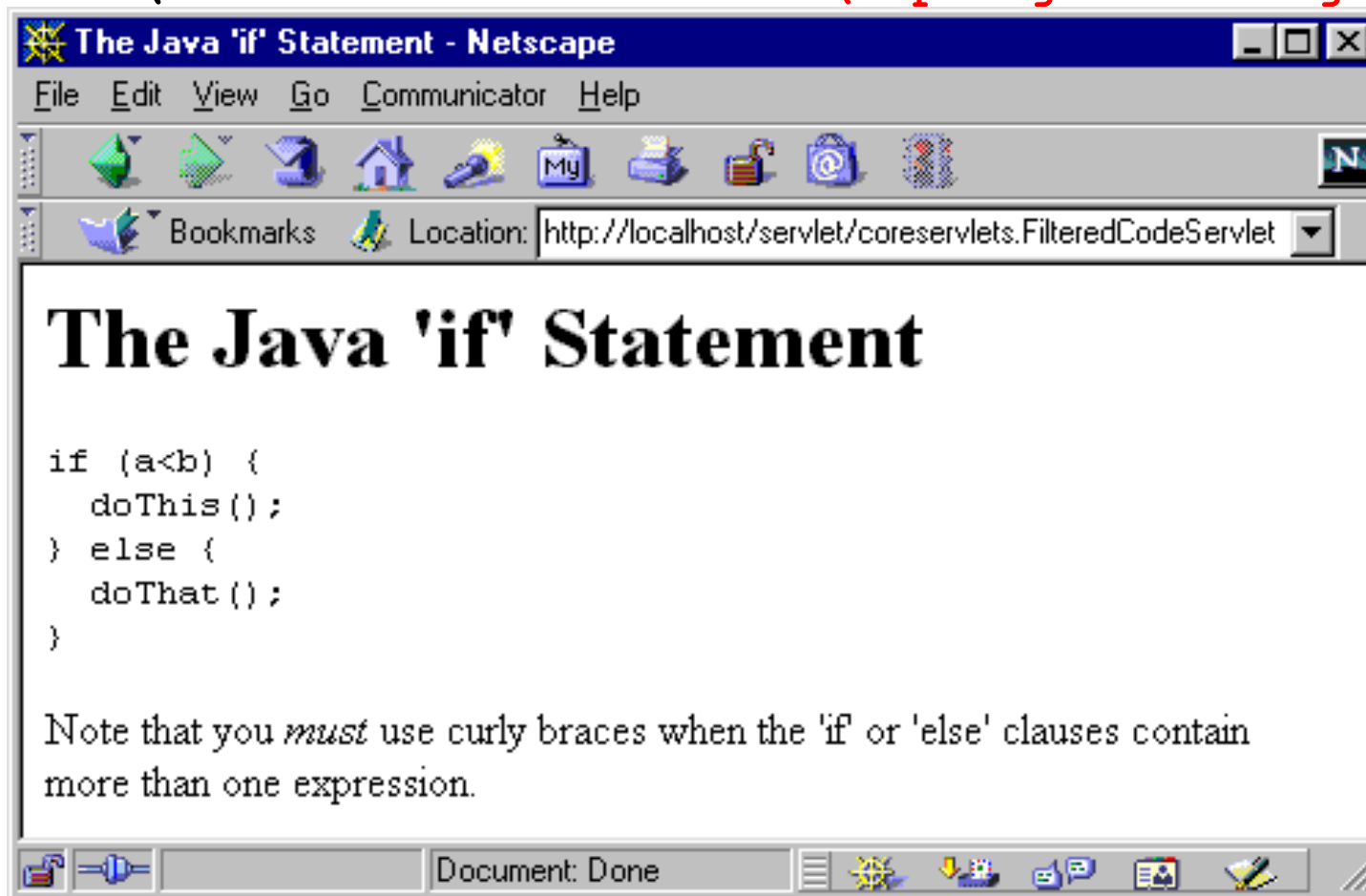


# Servlet That Fails to Filter (Result)



# Servlet That Properly Filters

```
public class FilteredCodeServlet extends BadCodeServlet {  
    public String getCodeFragment() {  
        return (ServletUtilities.filter(super.getCodeFragment()));  
    }  
}
```



# Summary

- **Query data comes from HTML forms as URL-encoded name/value pairs**
- **Servlets read data by calling `request.getParameter("name")`**
  - Results in value as entered into form, not as sent over network. I.e. *not* URL-encoded.
- **Always check for missing or malformed data**
  - Missing: null or empty string
  - Special case: query data that contains special HTML characters
    - Need to be filtered if query data will be placed into resultant HTML page



# Questions?

**Core Servlets & JSP book: [www.coreservlets.com](http://www.coreservlets.com)**

**More Servlets & JSP book: [www.moreservlets.com](http://www.moreservlets.com)**

**Servlet and JSP Training Courses: [courses.coreservlets.com](http://courses.coreservlets.com)**

**Slides © Marty Hall, <http://www.coreservlets.com>, book © Sun Microsystems Press**