



Handling the Client Request: Form Data

Core Servlets & JSP book: www.coreservlets.com
More Servlets & JSP book: www.moreservlets.com
Servlet and JSP Training Courses: courses.coreservlets.com

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

1

Agenda

- Why form data is important
- Processing form data in traditional CGI
- Processing form data in servlets
- Reading individual request parameters
- Reading all request parameters
- Real-life servlets: handling malformed data
- Filtering HTML-specific characters

2

The Role of Form Data

- **Example URL at online travel agent**
 - `http://host/path?user=Marty+Hall&origin=bwi&dest=lax`
 - Names come from HTML author; values usually come from end user
- **Parsing form (query) data in traditional CGI**
 - Read the data one way (QUERY_STRING) for GET requests, another way (standard input) for POST requests
 - Chop pairs at ampersands, then separate parameter names (left of the equal signs) from parameter values (right of the equal signs)
 - URL decode values (e.g., "%7E" becomes "~")
 - Need special cases for omitted values (param1=val1¶m2=¶m3=val3) and repeated parameters (param1=val1¶m2=val2¶m1=val3)

3

Form Data

www.coreservlets.com

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>
```

- **See CSAJSP Chapter 16 for details on forms**

4

Form Data

www.coreservlets.com

Aside: Installing HTML Files

- **Tomcat**
 - *install_dir*\webapps\ROOT\Form.html or
 - *install_dir*\webapps\ROOT*SomeDir*\Form.html
- **JRun**
 - *install_dir*\servers\default\default-app\Form.html or
 - *install_dir*\servers\default\default-app*SomeDir*\Form.html
- **URL**
 - http://localhost/Form.html or
 - http://localhost/*SomeDir*/Form.html
- **Custom Web applications**
 - Use a different directory with the same structure as the default Web app
 - Use directory name in URL (http://host/*dirName*/...)
 - See Chapter 4 of *More Servlets & JSP* for details

5

Form Data

www.coreservlets.com

HTML Form: Initial Result

A Sample Form Using GET - Netscape

File Edit View Go Communicator Help

Bookmarks Location: http://localhost/GetForm.html

A Sample Form Using GET

First name:

Last name:

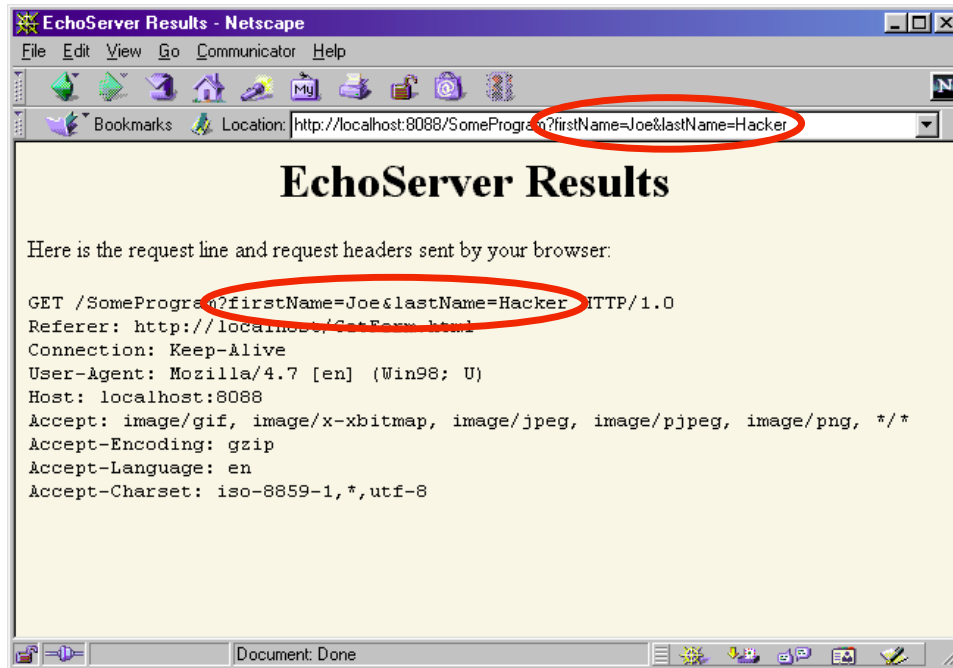
Document: Done

6

Form Data

www.coreservlets.com

HTML Form: Submission Result (Data Sent to EchoServer)



7

Form Data

www.coreservlets.com

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>

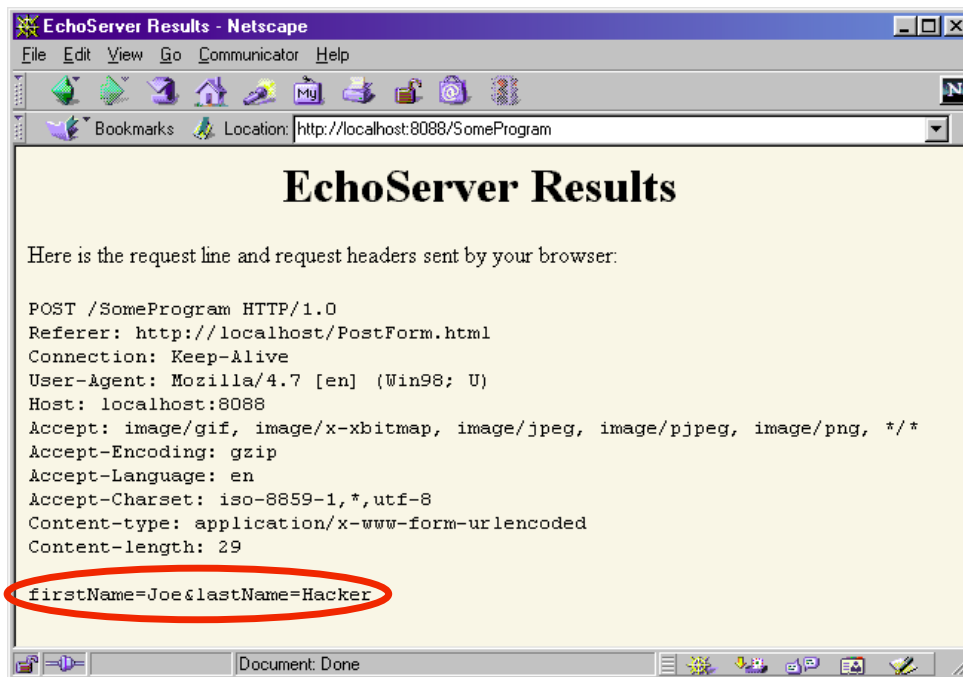
</BODY></HTML>
```

8

Form Data

www.coreservlets.com

Sending POST Data



9

Form Data

www.coreservlets.com

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

10

Form Data

www.coreservlets.com

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");
```

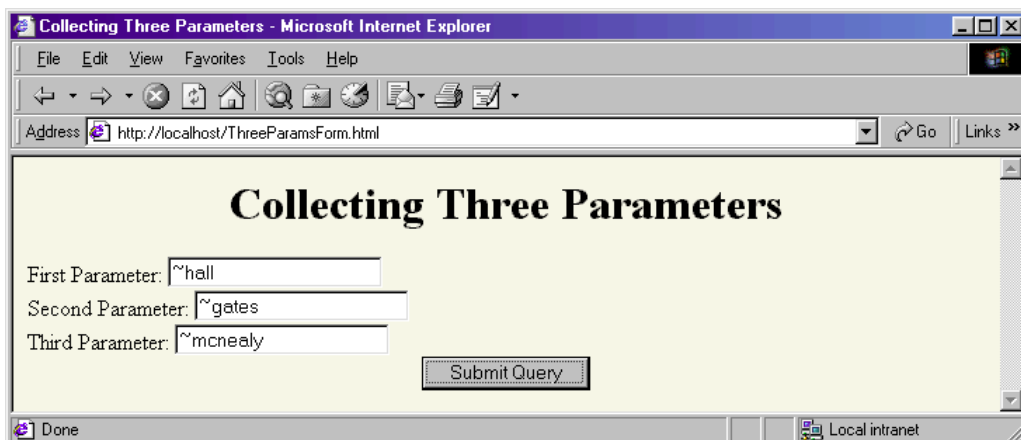
11

Form Data

www.coreservlets.com

An HTML Form With Three Parameters

```
<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>
```



12

Form Data

www.coreservlets.com

Reading the Three Parameters

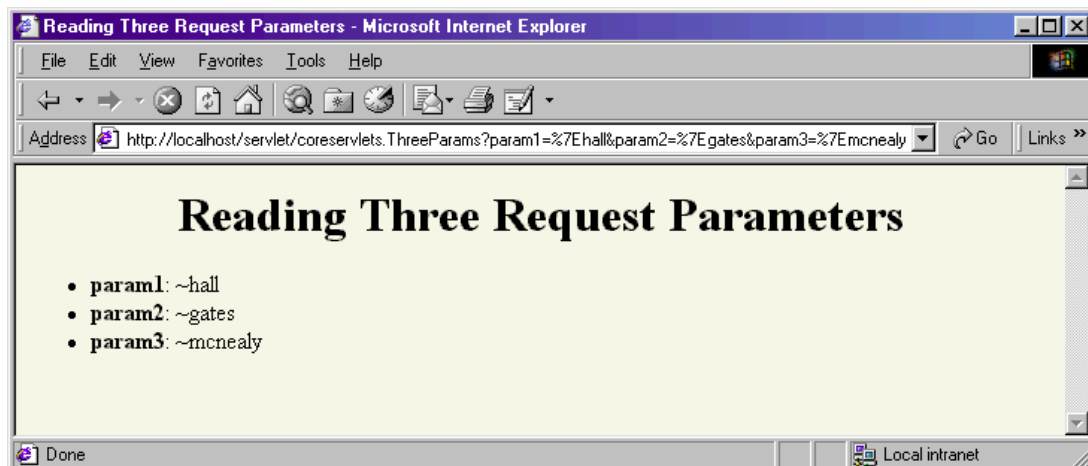
```
public class ThreeParams extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String title = "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>"); }}
```

13

Form Data

www.coreservlets.com

Reading Three Parameters: Result



14

Form Data

www.coreservlets.com

Reading All Parameters

```
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);  
}  
}
```

17

Form Data

www.coreservlets.com

Result of ShowParameters Servlet

The image shows two side-by-side screenshots from a Netscape browser window. The left window, titled "A Sample FORM using POST", displays a web form with the following fields: Item Number (127A), Quantity (12), Price Each (\$4.95), First Name (Marty), Last Name (Hall), Middle Initial (empty), Shipping Address (Johns Hopkins Applied Physics Lab, 11100 Johns Hopkins Rd., Laurel, MD 20723), Credit Card type (Java SmartCard selected), Credit Card Number (masked with asterisks), and Repeat Credit Card Number (masked with asterisks). A "Submit Order" button is at the bottom. The right window, titled "Reading All Request Parameters", shows a table of the request parameters. The table has two columns: "Parameter Name" and "Parameter Value(s)".

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">• 3.14159• 3.14159
firstName	Marty
itemNum	127A
cardType	Java SmartCard
quantity	12
lastName	Hall

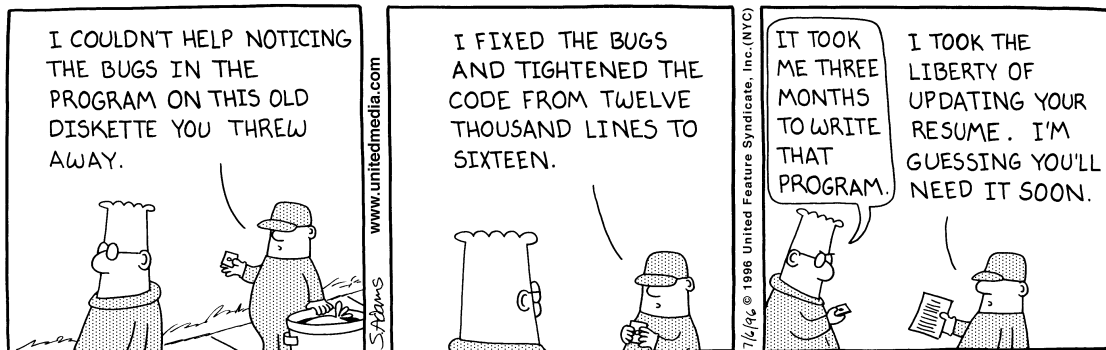
– Note that order of parameters in Enumeration does not match order they appeared in Web page

18

Form Data

www.coreservlets.com

A Resumé Posting Service



Dilbert used with permission of United Syndicates Inc.

19

Form Data

www.coreservlets.com

Posting Service: Front End

- Gathers resumé formatting and content information

Free Resume Posting - Microsoft Internet Explorer

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:

Heading text size:

Body font:

Body text size:

Foreground color:

Background color:

Next, give some general information about yourself:

Name:

Current or most recent title:

Email address:

Programming Languages:

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

Expert in data structures and computational methods.

<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>

Can prove that P is not equal to NP. Doesn't want to work for companies that don't know what this means.

<P>

Not related to the American politician.

Preview Submit

www.coreservlets.com

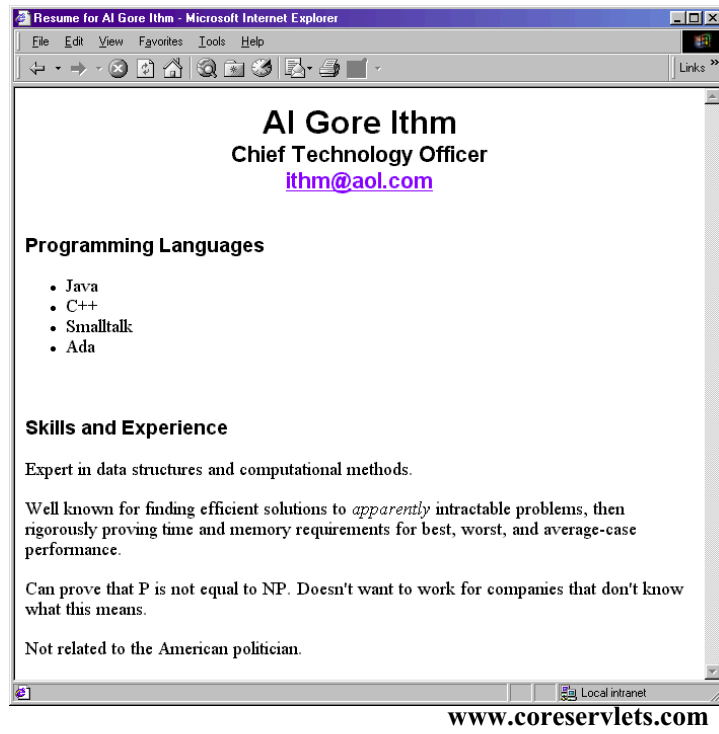
20

Form Data

www.coreservlets.com

Posting Service: Back End

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



21

Form Data

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("")) {
    ...
}
```

22

Form Data

www.coreservlets.com

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";

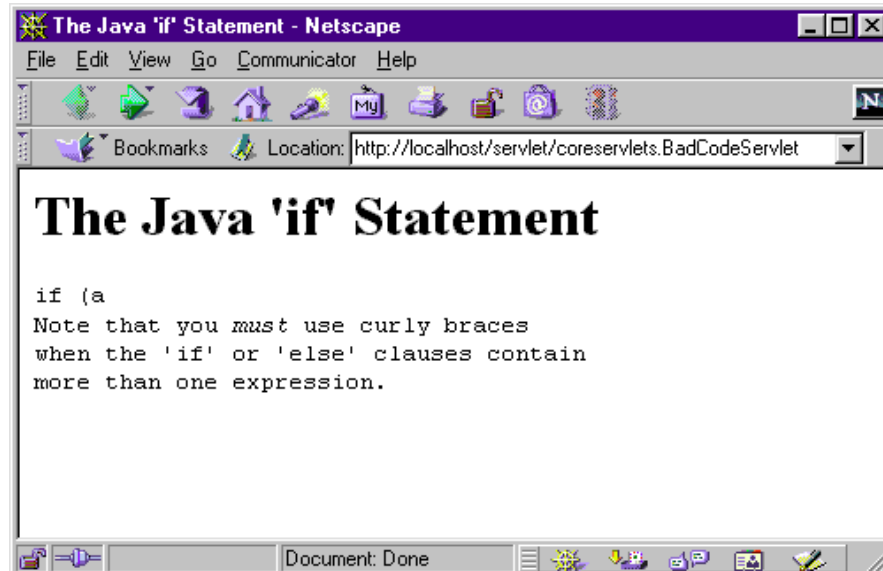
    out.println(ServletUtilities.headWithTitle(title) +
                "<BODY>\n" +
                "<H1>" + title + "</H1>\n" +
                "<PRE>\n" +
                getCodeFragment() +
                "</PRE>\n" +
                "Note that you <I>must</I> use curly braces\n" +
                "when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

27

Form Data

www.coreservlets.com

Servlet That Fails to Filter (Result)



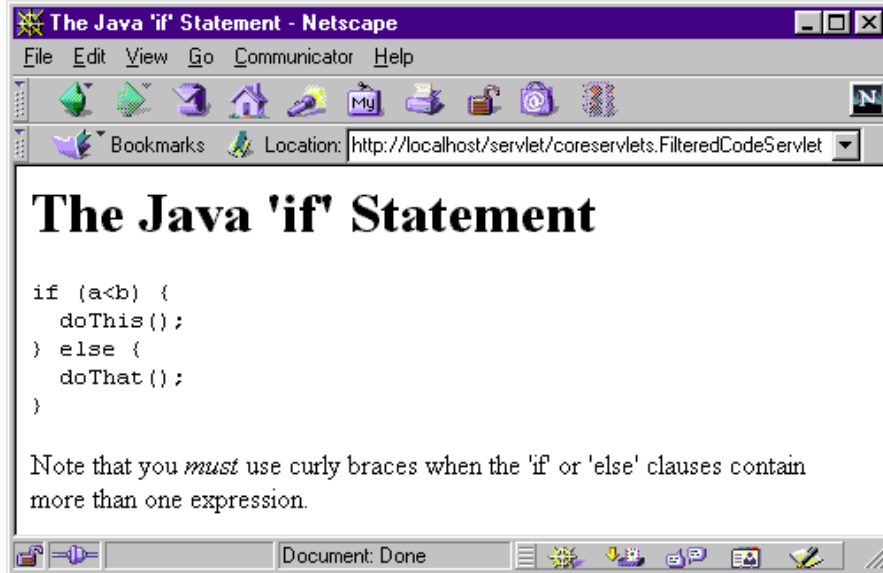
28

Form Data

www.coreservlets.com

Servlet That Properly Filters

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



29

Form Data

www.coreservlets.com

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

30

Form Data

www.coreservlets.com