



客户请求的处理： 表单数据

JSP, Servlet, & Struts Training Courses: <http://courses.coreservlets.com>
Available in US, China, Taiwan, HK, and Worldwide

JSP and Servlet Books from Sun Press: <http://www.coreservlets.com>
*Available in English, Chinese (simplified and traditional script),
and 12 other languages*

2

议程

- 表单数据的功用
- HTML表单的创建和提交
- 单个请求参数的读取
- 整个系列请求参数的读取
- 数据缺失或者异常时的处理
- 不完整表单提交的应对
- 请求参数中特殊字符的过滤

3

表单数据的功用

- 在线旅行社可能会用到的URL
 - `http://host/path?user=Marty+Hall&origin=bwi&dest=lax`
 - 名称由HTML制作者指定，而值由最终用户提供。
- 传统CGI中对表单（查询）数据的解析
 - 采用不同的方式读取GET请求（`QUERY_STRING`）和POST请求（标准输入）的数据。
 - 在&符号处将名/值对拆分开来，然后将参数名（等号左边）和参数值（等号右边）分开。
 - 对值进行URL解码（比如，“%7E”变为“~”）
- 在servlet中这一切得到极大简化
 - 所有情况下都使用`request.getParameter`。
 - 直接给出经过URL解码后的结果

4

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

表单数据的创建：HTML窗体

```
<!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>
```

- 表单的细节参见CSAJSP/2 第19章

5

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

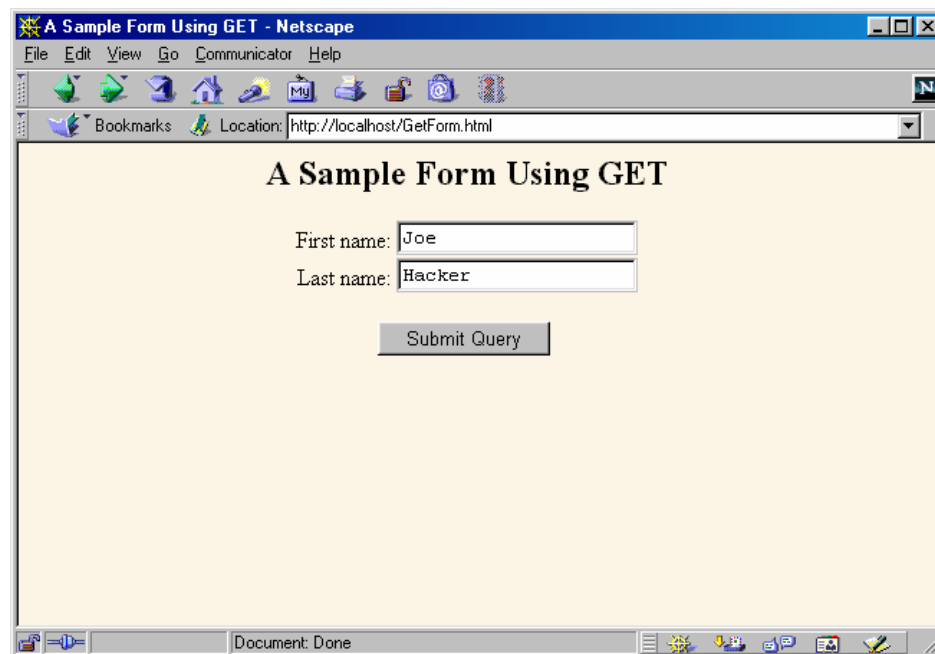
插入语: HTML文件的安装

- **HTML文件不放置在WEB-INF/classes目录中**
 - 它们需要放置在含有WEB-INF目录的目录中。
- **Tomcat**
 - `install_dir\webapps\ROOT\Form.html` 或
 - `install_dir\webapps\ROOT\SomeDir\Form.html`
- **URL**
 - `http://localhost/Form.html` 或
 - `http://localhost/SomeDir/Form.html`
- **自定义Web应用**
 - 使用不同于默认Web应用的目录，但使用相同的结构
 - 在URL中使用目录名 (`http://host/dirName/...`)
 - 详细信息参见 *Core Servlets & JSP* (第2版)的2.11节 和 *More Servlets & JSP*的第4章

6

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

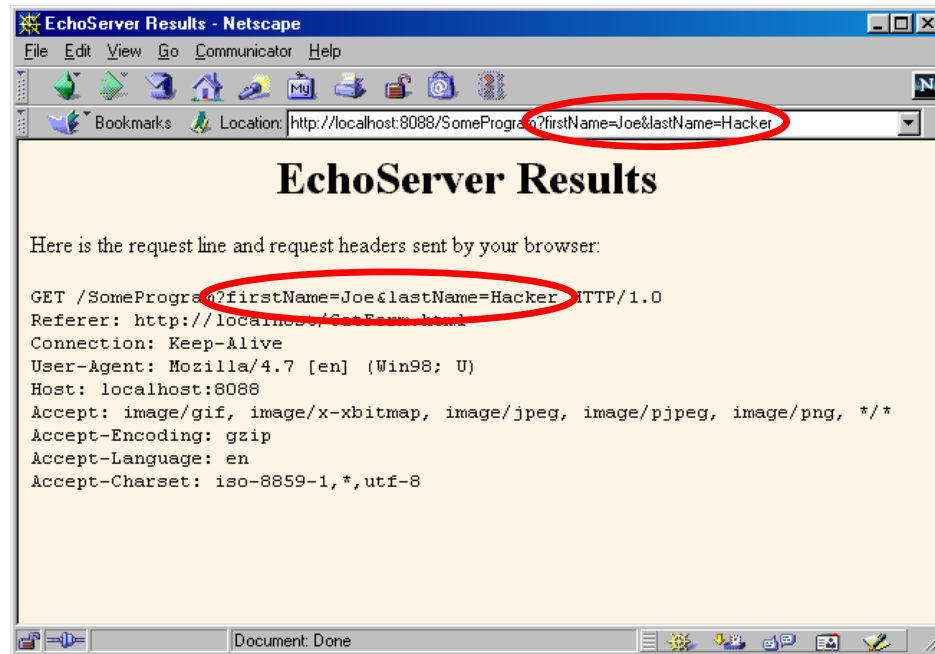
HTML表单：初始结果



7

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

HTML表单：提交的结果 (发送到EchoServer的数据)



8

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

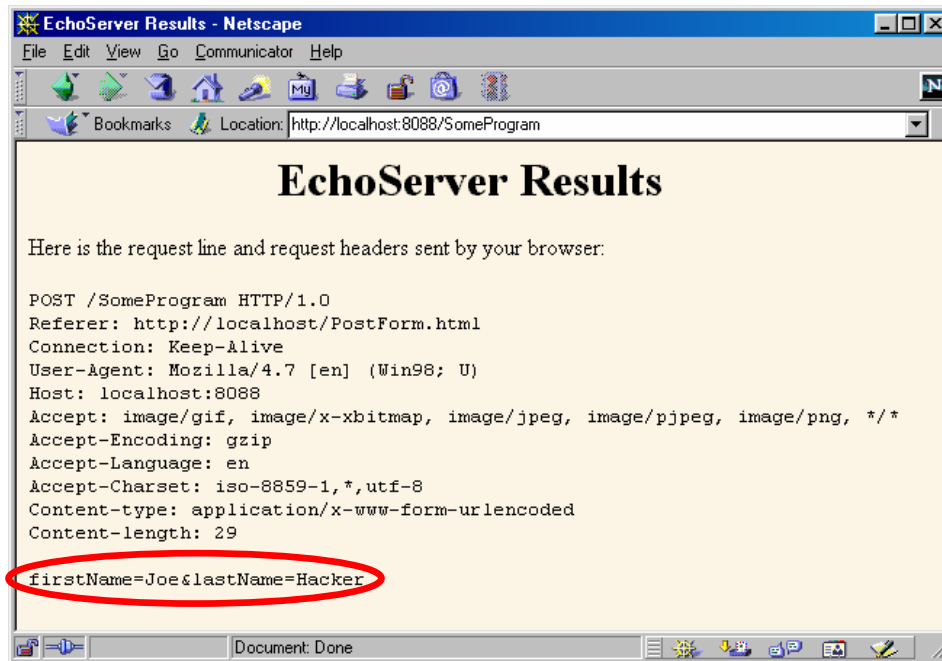
POST数据的发送

```
<!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>
```

9

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

POST数据的发送



10

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

Servlet中表单数据的读取

- **request.getParameter("name")**
 - 返回查询字符串中name首次出现时所对应的值，已经完成URL解码
 - 无论GET和POST请求都以相同的方式工作
 - 如果在查询数据中没有这个参数，则返回null
- **request.getParameterValues("name")**
 - 返回查询字符串中name所对应的所有值构成的数组，已完成URL解码
 - 如果参数没有重复出现，则返回仅有一个元素的数组
 - 如果在查询中没有这个参数则返回null
- **request.getParameterNames()或request.getParameterMap()**
 - 返回请求参数构成的Enumeration或Map
 - 常常仅用于调试目的

11

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

多语言输入的处理

- 使用server的默认字符集

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

- 从英文 (Latin-1)转换成日文

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

- 接收英文或日文

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

12

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

原始表单数据的读取以及对上载文件的分析

- 原始数据

- request.getReader
- request.getInputStream
 - 这样做之后，就不能再通过getParameter来获取数据

- 上载文件的分析

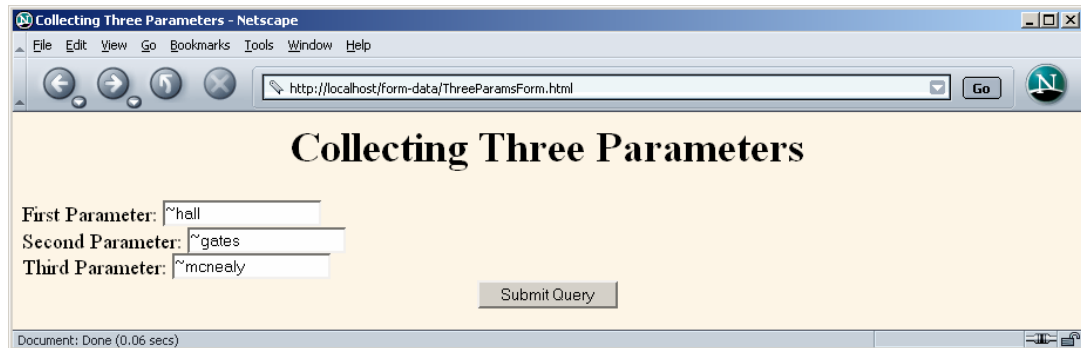
- HTML拥有提交整个文件的方法
 - `<INPUT TYPE="FILE"...>`
 - 参见《servlet和JSP核心编程》第2版的19.7节。
- servlet/JSP API没有内建的方式可以对这类文件进行分析
- Apache/Jakarta的“Commons”库中提供流行的第三方库提供这类支持
 - <http://jakarta.apache.org/commons/fileupload/>

13

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

含有三个参数的HTML表单

```
<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>
```



14

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

读取三个参数

```
public class ThreeParams extends HttpServlet {  
  public void doGet(HttpServletRequest request,  
                    HttpServletResponse response)  
    throws ServletException, IOException {  
    ...  
    out.println(docType +  
                "<HTML>\n" +  
                "<HEAD><TITLE>"+title + "</TITLE></HEAD>\n" +  
                "<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>");  
  }  
}
```

15

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

读取三个参数处理后的结果



16

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

读取所有参数

```
public class ShowParameters extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String docType =
            "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 " +
            "Transitional//EN">\n";
        String title = "Reading All Request Parameters";
        out.println(docType +
            "<HTML>\n" +
            "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
            "<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)");
```

17

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

读取所有参数 (续)

```
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);
    } 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>");
}
```

18 JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

读取所有参数 (续)

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

19 JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

读取所有参数 (简单表单)

A Sample FORM using POST

Item Number: 123-A
Description: Wild Wonder Widget
Price Each: \$456.78

First Name: Sam
Last Name: Palmisano
Middle Initial:

Shipping Address: One Microsoft Way
Redmond, WA 98052

Credit Card:
 Visa
 MasterCard
 American Express
 Discover
 Java SmartCard

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

Submit Order

20

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

读取所有参数 (结果)

Reading All Request Parameters

Parameter Name	Parameter Value(s)
cardNum	<ul style="list-style-type: none">• 3.1415927• 3.1415927
cardType	Java SmartCard
price	\$456.78
initial	No Value
itemNum	123-A
address	One Microsoft Way Redmond, WA 98052
description	Wild Wonder Widget
firstName	Sam
lastName	Palmisano

21

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

数据缺失或异常的检查

- **缺失**

- 表单中缺失某个字段
 - `getParameter`返回null
- 表单提交时字段为空
 - `getParameter`返回空字符串 (或者由空格组成的字符串)
- 在检查字符串是否为空之前必须检查它是否为null

```
String param = request.getParameter("someName");
if ((param == null) || (param.trim().equals(""))) {
    doSomethingForMissingValues(...);
} else {
    doSomethingWithParameter(param);
}
```

- **异常**

- 值为非空字符串，但格式错误

22

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

缺失或异常数据的处理

- **使用默认值**

- 用应用程序特定的标准值来替代缺失的值
- 参见随后的例子

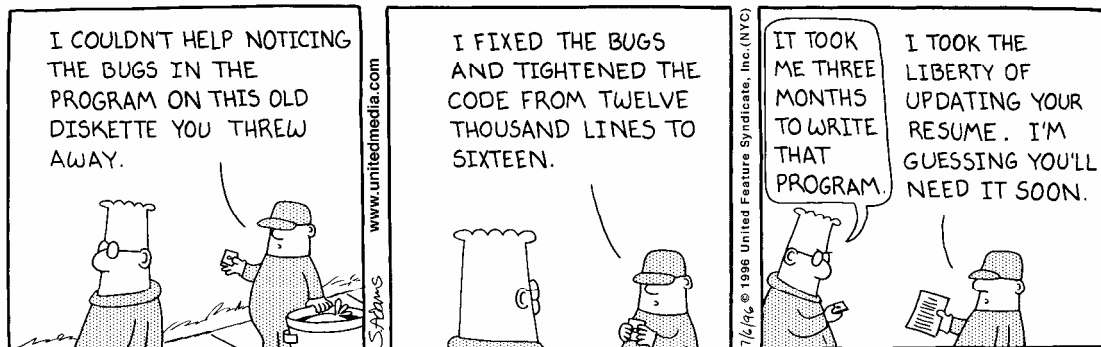
- **再次显示表单**

- 再次显示表单，将缺失的值标示出来
- 之前已经输入的值应该保留
- 实现这个功能有四种选择：
 - 由同一个servlet呈现表单、处理数据、并呈现结果。
 - 由一个servlet呈现表单；由另一个servlet处理数据并呈现结果。
 - 由JSP页面“手动地”呈现表单；由servlet或JSP页面处理数据并呈现结果。
 - 由JSP页面呈现表单，自动用从数据对象获得的值填充各个字段。由servlet或JSP页面处理数据并呈现结果。
- 具体的例子请参考本书的内容

23

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

应用默认值的例子：简历张贴网站



24

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

简历张贴网站：输入表单

Free Resume Posting - Netscape

http://localhost:form-data/SubmitResume.html

hot-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: Arial Black
Heading text size: 36
Body font: default
Body text size: 22
Foreground color: BLACK
Background color: #CCCC00

Next, give some general information about yourself:

Name: Al Gore Item
Current or most recent title: Chief Technologist
Email address: jhm@ocm.org
Programming Languages: Java, C++, Lisp, Ada

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 intractable problems, then rigorously proving time and space complexity for best-, worst-, and average-case performance.
<P>
Can prove that P is not equal to NP. Does not want to work for a company that does not know what this means.
<P>
Not related to the American politician.

Preview Submit

Document: Done (0.261 sec)

25

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

简历张贴网站：servlet代码

```
headingFont =
    replaceIfMissingOrDefault(headingFont, "");
int headingSize =
    getSize(request.getParameter("headingSize"),
            32);
String bodyFont =
    request.getParameter("bodyFont");
bodyFont =
    replaceIfMissingOrDefault(bodyFont, "");
int bodySize =
    getSize(request.getParameter("bodySize"), 18);
String fgColor = request.getParameter("fgColor");
fgColor =
    replaceIfMissing(fgColor, "BLACK");
String bgColor = request.getParameter("bgColor");
```

26

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

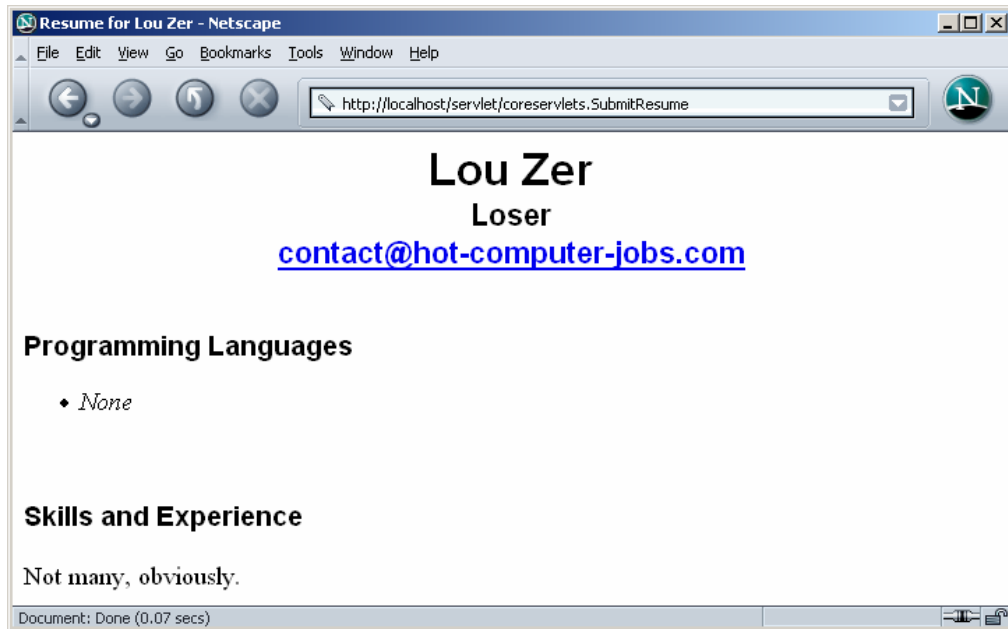
简历张贴网站：servlet代码(续)

```
private String replaceIfMissing(String orig,
                                String replacement) {
    if ((orig == null) || (orig.trim().equals(""))) {
        return(replacement);
    } else {
        return(orig);
    }
}
```

27

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

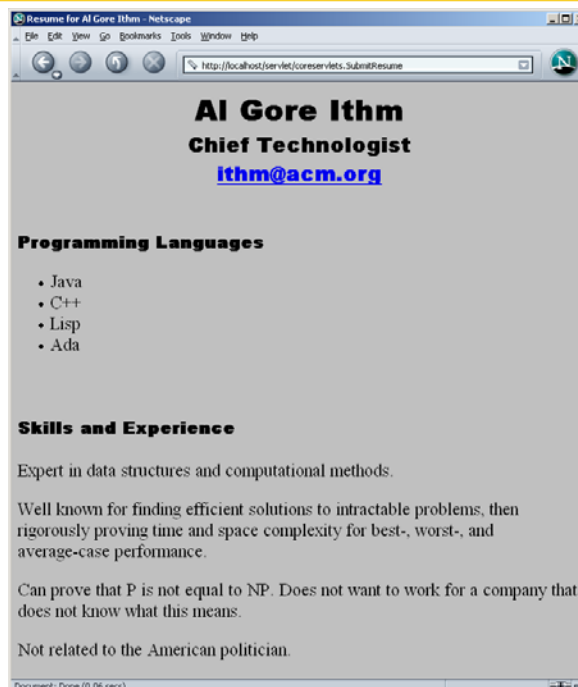
简历张贴网站： 不完整数据所获得的结果



28

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

简历张贴网站：完整数据的结果



29

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

过滤字符串中的HTML特殊字符

- 并非任何字符串都可以安全地插入到servlet的输出中
 - <和>在任何地方都会引起问题
 - &和“在HTML属性中会引发问题
- 有时手动的转换是不可能的
 - 字符串来源于另外的程序，或者已经按照某种标准格式进行组织的其他来源。
 - 字符串来源于HTML表单数据
- 如果不能有效地过滤掉来自于表单数据的特殊字符，则易受到跨网站脚本攻击 (*cross-site scripting attack*)
 - <http://www.cert.org/advisories/CA-2000-02.html>
 - <http://www.microsoft.com/technet/security/topics/ExSumCS.asp>

30

JSP/servlet/Struts/JSE training: <http://www.coreservlets.com>

过滤字符串中的HTML特殊字符 (代码)

```
public class ServletUtilities {
    public static String filter(String input) {
        if (!hasSpecialChars(input)) {
            return(input);
        }
        StringBuffer filtered =
            new StringBuffer(input.length());
        char c;
        for(int i=0; i<input.length(); i++) {
            c = input.charAt(i);
            switch(c) {
                case '<': filtered.append("&lt;"); break;
                case '>': filtered.append("&gt;"); break;
                case '"': filtered.append("&quot;"); break;
                case '&': filtered.append("&amp;"); break;
                default: filtered.append(c);
            }
        }
        return(filtered.toString());
    }
}
```

31

JSP/servlet/Struts/JSE training: <http://www.coreservlets.com>

显示代码示例的servlet: 没有过滤措施

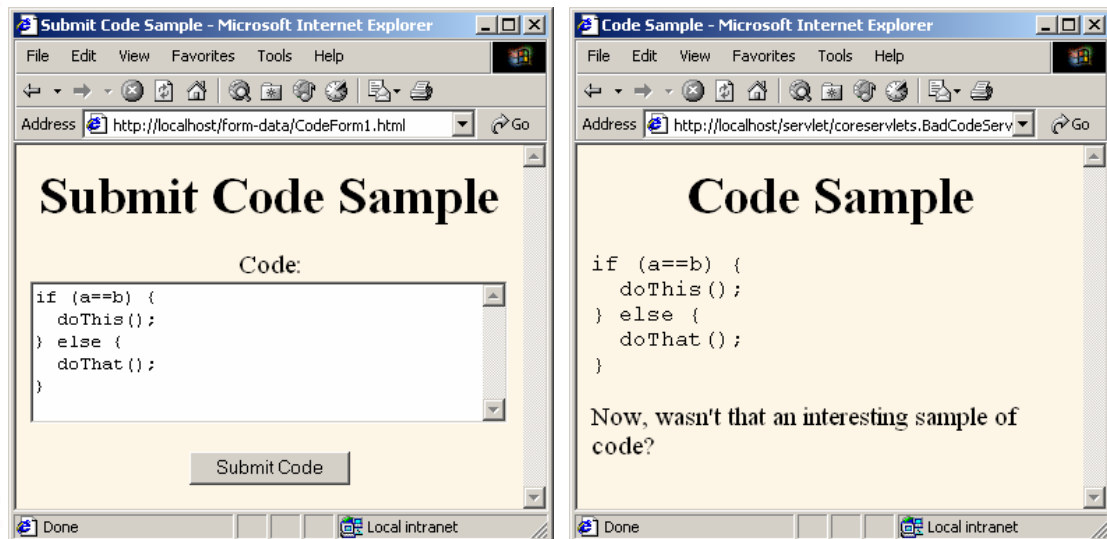
```
public class BadCodeServlet extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        ...
        out.println(docType +
            "<HTML>\n" +
            "<HEAD><TITLE>"+title+"</TITLE></HEAD>\n" +
            "<BODY BGCOLOR=\"#FDF5E6\">\n" +
            "<H1 ALIGN=\"CENTER\">" + title + "</H1>\n"+
            "<PRE>\n" +
            getCode(request) +
            "</PRE>\n" +
            "Now, wasn't that an interesting sample\n" +
            "of code?\n" +
            "</BODY></HTML>");
    }

    protected String getCode(HttpServletRequest request) {
        return(request.getParameter("code"));
    }
}
```

32

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

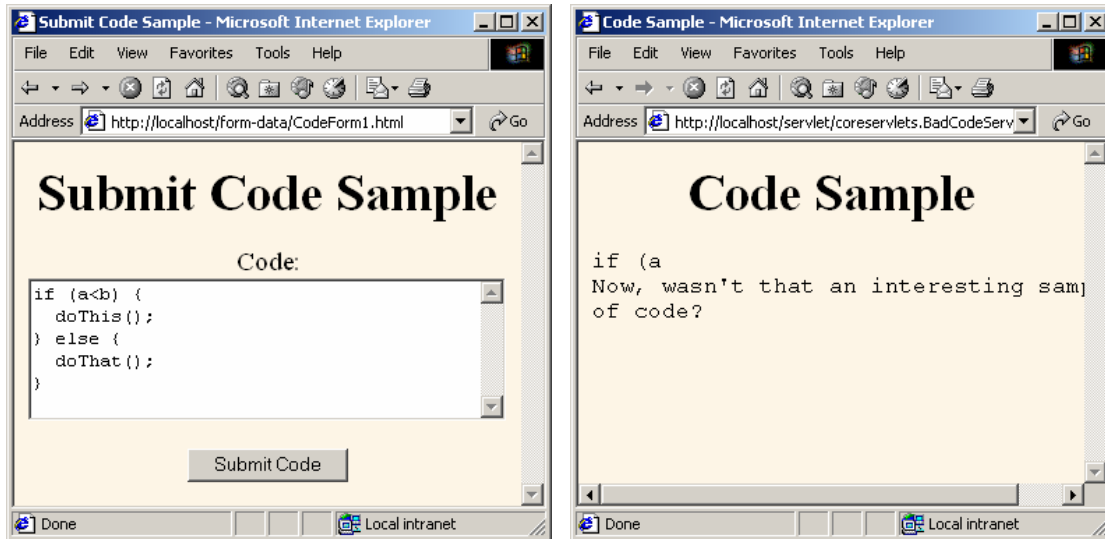
显示代码示例的servlet: 没有特殊字符



33

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

显示代码示例的servlet: 含有特殊字符



34

JSP/servlet/Struts/JSE training: <http://www.coreservlets.com>

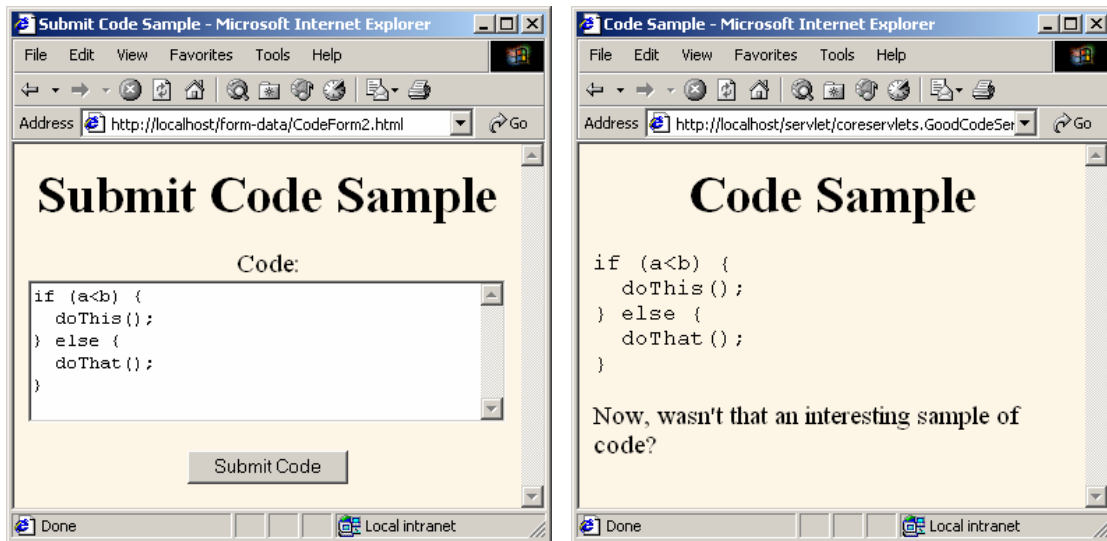
显示代码示例的servlet: 进行过滤

```
public class GoodCodeServlet extends BadCodeServlet {
    protected String getCode(HttpServletRequest request) {
        return
            (ServletUtilities.filter(super.getCode(request)));
    }
}
```

35

JSP/servlet/Struts/JSE training: <http://www.coreservlets.com>

显示代码示例的servlet (经过修正) : 含有特殊字符



36

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

高级议题 (具体的例子及细节请参考本书)

- 使用请求参数的值自动填充数据对象
 - 使用Javabeans (方法按照getBlah和setBlah进行命名的Java对象) 来存储输入的数据
 - 名为blah的请求参数会自动传递给相应的 setBlah方法。类型转换自动完成。有错误发生时使用默认值。
- 当参数缺失或异常时重新显示输入表单
 - 同一servlet呈现表单、处理数据、并呈现结果。
 - 该servlet首先检查输入的请求数据：如果没有找到任何数据，则呈现一个空表单。如果这个servlet找到部分请求数据，则将这部分数据提取出来，填入到表单中，并将其他缺失的字段标示出来。如果找到全部所需要的数据，则对请求进行处理并显示结果。

37

JSP/servlet/Struts/JSP training: <http://www.coreservlets.com>

小结

- **HTML表单发送的查询数据的形式是：经过URL编码的名/值对**
- **servlet调用request.getParameter(“name”)读取数据**
 - 按照在表单中输入的形式返回相应的值，而非在网络上的传送形式（经过URL编码后的形式）。
- **一定要检查数据是否缺失或异常**
 - 缺失：null或空字符串
 - 特殊情况：查询数据含有HTML专用字符
 - 如果需要将查询数据放到作为结果生成的HTML页面中，则需要过滤并处理这些字符。

38

JSP/servlet/Struts/JSE training: <http://www.coreservlets.com>

© 2004 Marty Hall



问题？

JSP, Servlet, & Struts Training Courses: <http://courses.coreservlets.com>
Available in US, China, Taiwan, HK, and Worldwide

JSP and Servlet Books from Sun Press: <http://www.coreservlets.com>
*Available in English, Chinese (simplified and traditional script),
and 12 other languages*

39