

4.4 Example: Reading All Parameters

The previous example extracts parameter values from the form data according to prespecified parameter names. It also assumes that each parameter has exactly one value. Here's an example that looks up *all* the parameter names that are sent and puts their values in a table. It highlights parameters that have missing values as well as ones that have multiple values. Although this approach is rarely used in production servlets (if you don't know the names of the form parameters, you probably don't know what to do with them), it is quite useful for debugging.

First, the servlet looks up all the parameter names with the `getParameterNames` method of `HttpServletRequest`. This method returns an `Enumeration` that contains the parameter names in an unspecified order. Next, the servlet loops down the `Enumeration` in the standard manner, using `hasMoreElements` to determine when to stop and using `nextElement` to get each parameter name. Since `nextElement` returns an `Object`, the servlet casts the result to a `String` and passes that to `getParameterValues`, yielding an array of strings. If that array is one entry long and contains only an empty string, then the parameter had no values and the servlet generates an italicized "No Value" entry. If the array is more than one entry long, then the parameter had multiple values and the values are displayed in a bulleted list. Otherwise, the single value is placed directly into the table.

The source code for the servlet is shown in [Listing 4.3](#); [Listing 4.4](#) shows the HTML code for a front end you can use to try out the servlet. [Figures 4-3](#) and [4-4](#) show the result of the HTML front end and the servlet, respectively.

Listing 4.3 ShowParameters.java

```
package coreservlets;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

/** Shows all the parameters sent to the servlet via either
 * GET or POST. Specially marks parameters that have
 * no values or multiple values.
 */
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)");
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>");
}

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

```

Figure 4-3. HTML form that collects data for the `ShowParameters` servlet.

A Sample FORM using POST - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://localhost/form-data>ShowParametersPostForm.html Go

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: [empty]

Shipping Address: One Microsoft Way
Redmond, WA 98052

Credit Card:

Visa

MasterCard

American Express

Discover

Java SmartCard

Credit Card Number: [XXXXXXXXXXXX]

Repeat Credit Card Number: [XXXXXXXXXXXX]

Submit Order

Done Local intranet

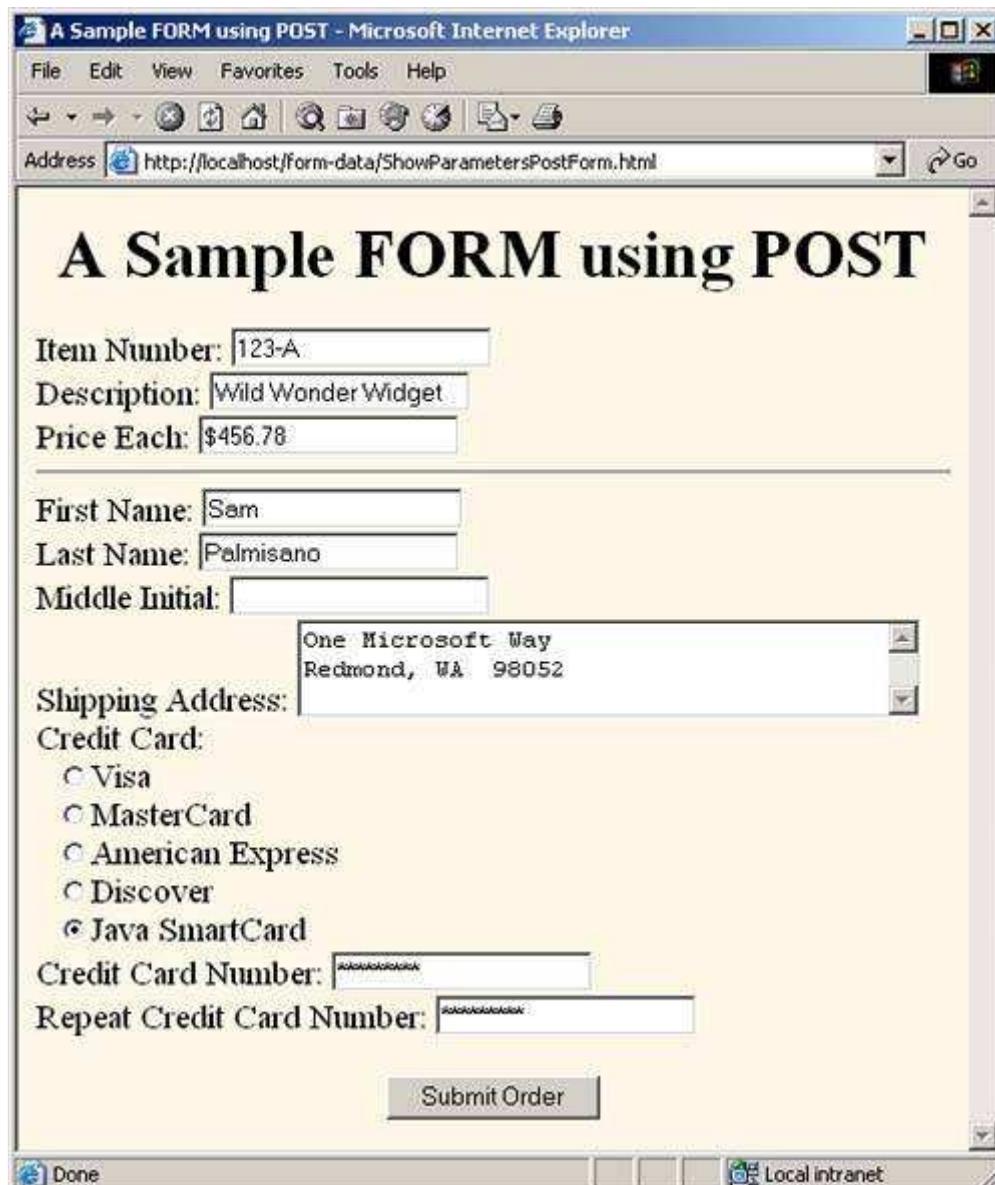
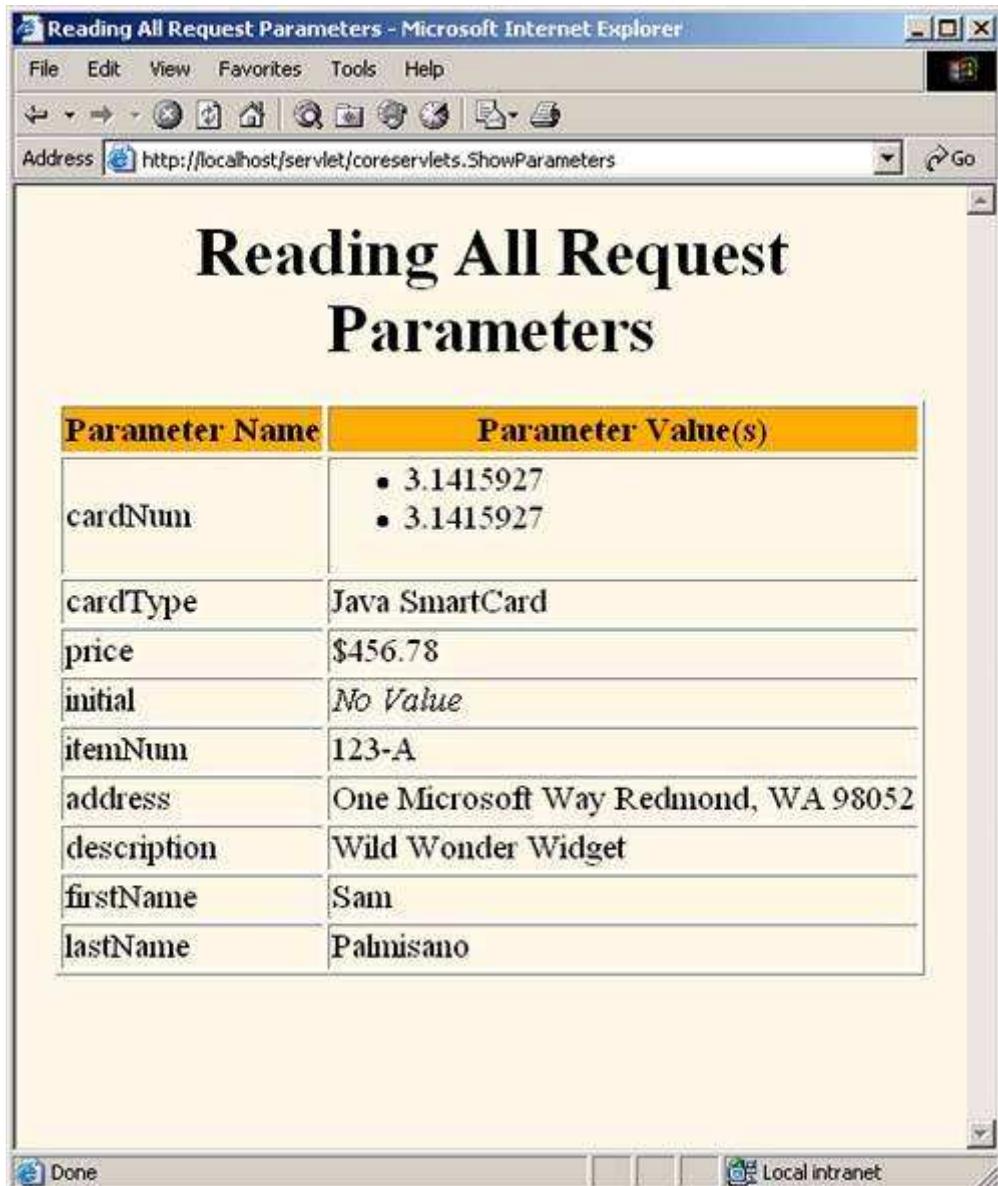


Figure 4-4. Result of the `ShowParameters` servlet.



Notice that the servlet uses a `doPost` method that simply calls `doGet`. That's because we want it to be able to handle *both* `GET` and `POST` requests. This approach is a good standard practice if you want HTML interfaces to have some flexibility in how they send data to the servlet. See the discussion of the `service` method in [Section 3.6](#) (The Servlet Life Cycle) for a discussion of why having `doPost` call `doGet` (or vice versa) is preferable to overriding `service` directly. The HTML form from [Listing 4.4](#) uses `POST`, as should *all* forms that have password fields (for details, see [Chapter 19](#), "Creating and Processing HTML Forms"). However, the `ShowParameters` servlet is not specific to that particular front end, so the source code archive site at <http://www.coreservlets.com/> includes a similar HTML form that uses `GET` for you to experiment with.

Listing 4.4 ShowParametersPostForm.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>A Sample FORM using POST</TITLE></HEAD>
<BODY BGCOLOR="#FDF5E6">
<H1 ALIGN="CENTER">A Sample FORM using POST</H1>
<FORM ACTION="/servlet/coreservlets.ShowParameters"
      METHOD="POST">
  Item Number: <INPUT TYPE="TEXT" NAME="itemNum"><BR>
  Description: <INPUT TYPE="TEXT" NAME="description"><BR>
  Price Each: <INPUT TYPE="TEXT" NAME="price" VALUE="$"><BR>
```

```
<HR>
First Name: <INPUT TYPE="TEXT" NAME="firstName"><BR>
Last Name: <INPUT TYPE="TEXT" NAME="lastName"><BR>
Middle Initial: <INPUT TYPE="TEXT" NAME="initial"><BR>
Shipping Address:
<TEXTAREA NAME="address" ROWS=3 COLS=40></TEXTAREA><BR>
Credit Card:<BR>
  &nbsp;&nbsp;<INPUT TYPE="RADIO" NAME="cardType"
    VALUE="Visa">Visa<BR>
  &nbsp;&nbsp;<INPUT TYPE="RADIO" NAME="cardType"
    VALUE="MasterCard">MasterCard<BR>
  &nbsp;&nbsp;<INPUT TYPE="RADIO" NAME="cardType"
    VALUE="Amex">American Express<BR>
  &nbsp;&nbsp;<INPUT TYPE="RADIO" NAME="cardType"
    VALUE="Discover">Discover<BR>
  &nbsp;&nbsp;<INPUT TYPE="RADIO" NAME="cardType"
    VALUE="Java SmartCard">Java SmartCard<BR>
Credit Card Number:
<INPUT TYPE="PASSWORD" NAME="cardNum"><BR>
Repeat Credit Card Number:
<INPUT TYPE="PASSWORD" NAME="cardNum"><BR><BR>
<CENTER><INPUT TYPE="SUBMIT" VALUE="Submit Order"></CENTER>
</FORM>
</BODY></HTML>
```

[\[Team LiB \]](#)

◀ PREVIOUS NEXT ▶