

Servlet: usare i parametri provenienti dai forms

Programmazione per il web

AA 2013-2014 Gino Perna

Parametri (request : da client a server)

- Ogni pagina web che abbia dei campi (visibili o nascosti) reinvia, attraverso un link in GET od una action in POST i parametri al server
- Questi parametri possono essere processati dal server (ovvero una servlet per noi)

Esempio Parametri

http://www.vodafone.it/190/trilogy/jsp/home.do?tabName=Privati&ty_skip_md=tri

4 forms

Form

ID	NAME	METHOD	ACTION
mainsearch	searchForm	post	/190/trilogy/jsp/search.do?tk=HPPrivati,t

Elements

INDEX	ID	NAME	TYPE	VALUE	LABEL	SIZE	MAXIMUM LENGTH	STATE
0	searchInputField	criteria	text	Ricerca			32	

Form

ID	NAME	METHOD	ACTION
account	loginForm	post	https://www.vodafone.it/190/trilogy/jsp/login.do?tk=9608,L0

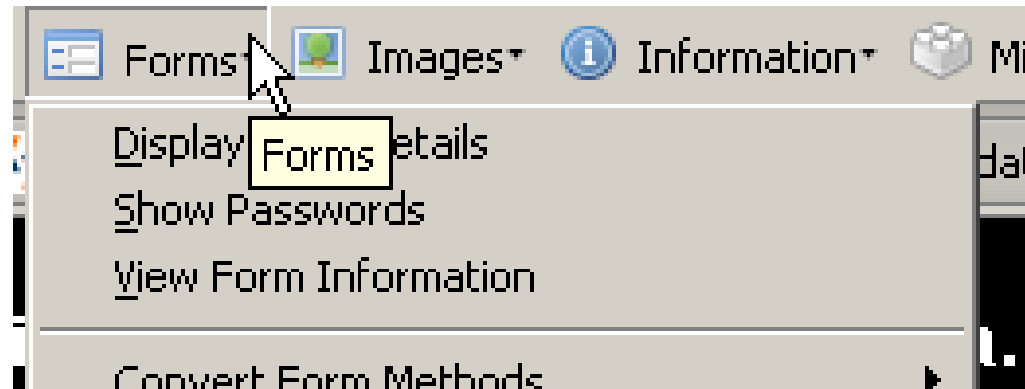
Elements

INDEX	ID	NAME	TYPE	VALUE	LABEL	SIZE	MAXIMUM LENGTH	STATE
0		username	text	Username				
1	pwd	password	password					
2	bt	passwordTxt	text	Password				

The screenshot shows the Vodafone Italy website. At the top, there is a search bar with the text 'Ricerca' and a 'Vai' button. Below the search bar is a red horizontal bar. To the right of the red bar is the Vodafone logo and the text 'vodafone lab'. Below the logo is a 'Login' section with a purple header. It contains a 'Username' field, a 'Password' field, and a 'login' button. Below the login button are three links: 'Li hai dimenticati?', 'Registrati', and 'Accedi alla tua Mail', each with a right-pointing arrow.

Parametri analisi

- Per analizzare i parametri di un form utilizzare le funzioni dell'estensione web-developer
 - ❑ forms>view forms info
 - ❑ forms>display forms details



Passaggio parametri (request)

- Ogni accesso alla servlet puo' avere un qualsiasi numero di parametri associati.
- Questi sono normalmente coppie nome/valore
- I valori sono ottenuti assieme alla query string (GET) o come dati encoded (POST).

Passaggio parametri (request)

- Servlets handles form data parsing automatically using the following methods depending on the situation:
 - ❑ **getParameter():** You call `request.getParameter()` method to get the value of a form parameter.
 - ❑ **getParameterValues():** Call this method if the parameter appears more than once and returns multiple values, for example checkbox.
 - ❑ **getParameterNames():** Call this method if you want a complete list of all parameters in the current request.

Metodo Get

- The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the ? character as follows:
`http://www.test.com/hello?key1=value1&key2=value2`
- The GET method is the default method to pass information from browser to web server and it produces a long string that appears in your browser's Location:box. Never use the GET method if you have password or other sensitive information to pass to the server. The GET method has size limitation: only 1024 characters can be in a request string.
- This information is passed using QUERY_STRING header and will be accessible through QUERY_STRING environment variable and Servlet handles this type of requests using doGet() method.

Metodo Post

- A generally more reliable method of passing information to a backend program is the POST method. This packages the information in exactly the same way as GET methods, but instead of sending it as a text string after a ? in the URL it sends it as a separate message. This message comes to the backend program in the form of the standard input which you can parse and use for your processing. Servlet handles this type of requests using doPost() method.

Esempio lettura in get

```
// Import required java libraries
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

// Extend HttpServlet class
public class HelloForm extends HttpServlet {

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException
    {
        // Set response content type
        response.setContentType("text/html");
    }
}
```

Esempio lettura in get

```
PrintWriter out = response.getWriter();
String title = "Using GET Method to Read Form Data";
String docType =
    "<!doctype html public \"-//w3c//dtd html 4.0 \" +
    \"transitional//en\">\n";
out.println(docType + "<html>\n" +
    "<head><title>" + title + "</title></head>\n" +
    "<body bgcolor=\"#f0f0f0\">\n" +
    "<h1 align=\"center\">" + title + "</h1>\n" +
    "<ul>\n" +
    "  <li><b>First Name</b>: "
    + request.getParameter("first_name") + "\n" +
    "  <li><b>Last Name</b>: "
    + request.getParameter("last_name") + "\n" +
    "</ul>\n" +
    "</body></html>");
```

```
}}
```

Esempio lettura in get

- Per usare la servlet precedente digitare nel browser:

`http://<nomemiasapplicazione>/HelloForm?
first_name=ZARA&last_name=ALI`

Using GET Method to Read Form Data

- First Name: ZARA
- Last Name: ALI

Esempio lettura in get da form

- Sempre utilizzando la servlet precedente, scrivere il seguente file index.html

```
<html>
<body>
<form action="HelloForm" method="GET">
First Name: <input type="text" name="first_name">
<br />
Last Name: <input type="text" name="last_name" />
<input type="submit" value="Submit" />
</form>
</body>
</html>
```

Esempio lettura in get da form

- Si noti che:
 - Action corrisponde al nome della servlet
 - I due parametri (campi di input) hanno come name il nome dei campi
- Se richiamato nel browser visualizza

First Name:

Last Name:

e se riempio i campi e premo submit dovrei
ottenere l'uscita dell'esempio precedente

Esempio lettura in post da form

- Sempre utilizzando la servlet precedente, scrivere il seguente file index.html

```
<html>
<body>
<form action="HelloForm" method="POST">
First Name: <input type="text" name="first_name">
<br />
Last Name: <input type="text" name="last_name" />
<input type="submit" value="Submit" />
</form>
</body>
</html>
```

Here is the actual output of the above form, Try to enter First and Last Name and then click submit button to see the result on your local machine where tomcat is running.

First Name:

Last Name:

Esempio lettura in post da form

- Come si notera' il risultato e' lo stesso
- In get pero' posso chiamare la servlet con un url (esempio da un bookmark)
- In post devo passare per forza dal form

Utilizzo e lettura dei checkbox

- Checkboxes are used when more than one option is required to be selected.
- Here is example HTML code, CheckBox.htm, for a form with two checkboxes

```
<html>
<body>
<form action="CheckBox" method="POST" target="_blank">
<input type="checkbox" name="maths" checked="checked" /> Maths
<input type="checkbox" name="physics" /> Physics
<input type="checkbox" name="chemistry" checked="checked" />
Chemistry
<input type="submit" value="Select Subject" />
</form>
</body>
</html>
```

The result of this code is the following form

☒ Maths ☐ Physics ☒ Chemistry

Lettura dei checkbox

```
// Import required java libraries
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

// Extend HttpServlet class
public class CheckBox extends HttpServlet {
    // Method to handle GET method request.
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException
    {
        // Set response content type
        response.setContentType("text/html");
```

Lettura dei checkbox

```
PrintWriter out = response.getWriter();
String title = "Reading Checkbox Data";
String docType =
    "<!doctype html public \"-//w3c//dtd html 4.0 " +
    "transitional//en\">\n";
out.println(docType + "<html>\n" +
    "<head><title>" + title +
"</title></head>\n" +
    "<body bgcolor=\"#f0f0f0\">\n" +
    "<h1 align=\"center\">" + title + "</h1>\n" +
    "<ul>\n" +
    "  <li><b>Maths Flag : </b>: "
    + request.getParameter("maths") + "\n" +
    "  <li><b>Physics Flag: </b>: "
```

Lettura dei checkbox

```
        + request.getParameter("physics") + "\n" +  
        "    <li><b>Chemistry Flag: </b>: "  
        + request.getParameter("chemistry") + "\n" +  
        "</ul>\n" +  
        "</body></html>");  
    }  
  
    // Method to handle POST method request.  
    public void doPost(HttpServletRequest request,  
                        HttpServletResponse response)  
        throws ServletException, IOException {  
        doGet(request, response);  
    }  
}
```

Lettura dei checkbox

Reading Checkbox Data

- **Maths Flag** : : on
- **Physics Flag**: : null
- **Chemistry Flag**: : on

Lettura di tutti i parametri

- Following is the generic example which uses **getParameterNames()** method of `HttpServletRequest` to read all the available form parameters. This method returns an `Enumeration` that contains the parameter names in an unspecified order.
- Once we have an `Enumeration`, we can loop down the `Enumeration` in the standard manner, using `hasMoreElements()` method to determine when to stop and using `nextElement()` method to get each parameter name.

Lettura di tutti i parametri

```
// Import required java libraries
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;

// Extend HttpServlet class
public class ReadParams extends HttpServlet {
    // Method to handle GET method request.
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException
    {
        // Set response content type
        response.setContentType("text/html");
```

Lettura di tutti i parametri

```
PrintWriter out = response.getWriter();
String title = "Reading All Form Parameters";
String docType =
    "<!doctype html public \"-//w3c//dtd html 4.0 " +
    "transitional//en\ ">\n";
out.println(docType +    "<html>\n" +
    "<head><title>" + title + "</title></head>\n" +
    "<body bgcolor=\n\"#f0f0f0\ ">\n" +
    "<h1 align=\n\"center\ ">" + title + "</h1>\n" +
    "<table width=\n\"100%\n" border=\n\"1\ "
align=\n\"center\ ">\n" +
    "<tr bgcolor=\n\"#949494\ ">\n" +
    "<th>Param Name</th><th>Param Value(s)</th>\n"+
    "</tr>\n");
```

Lettura di tutti i parametri

```
Enumeration paramNames = request.getParameterNames();

while(paramNames.hasMoreElements()) {
    String paramName = (String)paramNames.nextElement();
    out.print("<tr><td>" + paramName + "</td>\n<td>");
    String[] paramValues =
        request.getParameterValues(paramName);
    // Read single valued data
    if (paramValues.length == 1) {
        String paramValue = paramValues[0];
        if (paramValue.length() == 0)
            out.println("<i>No Value</i>");
        else
            out.println(paramValue);
    }
}
```


Lettura di tutti i parametri

```
    } else {  
        // Read multiple valued data  
        out.println("<ul>");  
        for(int i=0; i < paramValues.length; i++) {  
            out.println("<li>" + paramValues[i]);  
        }  
        out.println("</ul>");  
    }  
}  
out.println("</tr>\n</table>\n</body></html>");  
}
```

```
}
```

Lettura di tutti i parametri esempio

```
<html>
<body>
<form action="ReadParams" method="POST" target="_blank">
<input type="checkbox" name="maths" checked="checked" /> Maths
<input type="checkbox" name="physics" /> Physics
<input type="checkbox" name="chemistry" checked="checked" /> Chem
<input type="submit" value="Select Subject" />
</form>
</body>
</html>
```

Now calling servlet using above form would generate following result:

Reading All Form Parameters

Param Name	Param Value(s)
maths	on
chemistry	on

You can try above servlet to read any other form's data which is having other objects like text box, radio button or drop down box etc.

Leggere da un file in una servlet

```
response.setContentType("text/html");

String filename = "/WEB-INF/message.properties";
ServletContext context = getServletContext();
InputStream inp = context.getResourceAsStream(filename);
if (inp != null) {
    InputStreamReader isr = new InputStreamReader(inp);
    BufferedReader reader = new BufferedReader(isr);
    PrintWriter pw = response.getWriter();
    pw.println("<html><head><title>Read Text File</title></head>
<body bgcolor='cyan'></body></html>");
    String text = "";
    while ((text = reader.readLine()) != null) {
        pw.println("<h2><i><b>" + text + "</b></i></b><br>");
    }
}
```
