

# JSP - FORM PROCESSING

[http://www.tutorialspoint.com/jsp/jsp\\_form\\_processing.htm](http://www.tutorialspoint.com/jsp/jsp_form_processing.htm)

Copyright © tutorialspoint.com

You must have come across many situations when you need to pass some information from your browser to web server and ultimately to your backend program. The browser uses two methods to pass this information to web server. These methods are GET Method and POST Method.

## GET method:

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 which can be handled using `getQueryString` and `getParameter` methods of request object.

## POST method:

A generally more reliable method of passing information to a backend program is the POST method.

This method 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.

JSP handles this type of requests using `getParameter` method to read simple parameters and `getInputStream` method to read binary data stream coming from the client.

## Reading Form Data using JSP

JSP 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.
- **getInputStream:** Call this method to read binary data stream coming from the client.

## GET Method Example Using URL:

Here is a simple URL which will pass two values to HelloForm program using GET method.

**`http://localhost:8080/main.jsp?first_name=ZARA&last_name=ALI`**

Below is **main.jsp** JSP program to handle input given by web browser. We are going to use **getParameter** method which makes it very easy to access passed information:

```
<html>
<head>
<title>Using GET Method to Read Form Data</title>
</head>
<body>
<center>
<h1>Using GET Method to Read Form Data</h1>
<ul>
<li><p><b>First Name:</b>
    <%= request.getParameter("first_name")%>
</p></li>
<li><p><b>Last Name:</b>
    <%= request.getParameter("last_name")%>
</p></li>
</ul>
</body>
</html>
```

Now type *http://localhost:8080/main.jsp?first\_name=ZARA&last\_name=ALI* in your browser's Location:box. This would generate following result:

## USING GET METHOD TO READ FORM DATA

---

- **First Name:** ZARA
- **Last Name:** ALI

### GET Method Example Using Form:

Here is a simple example which passes two values using HTML FORM and submit button. We are going to use same JSP main.jsp to handle this input.

```
<html>
<body>
<form action="main.jsp" 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>
```

Keep this HTML in a file *Hello.htm* and put it in <Tomcat-installation-directory>/webapps/ROOT directory. When you would access *http://localhost:8080/Hello.htm*, here is the actual output of the above form.

First Name:

Last Name:

Try to enter First Name and Last Name and then click submit button to see the result on your local machine where tomcat is running. Based on the input provided, it will generate similar result as mentioned in the above example.

## POST Method Example Using Form:

Let us do little modification in the above JSP to handle GET as well as POST methods. Below is **main.jsp** JSP program to handle input given by web browser using GET or POST methods.

Infact there is no change in above JSP because only way of passing parameters is changed and no binary data is being passed to the JSP program. File handling related concepts would be explained in separate chapter where we need to read binary data stream.

```
<html>
<head>
<title>Using GET and POST Method to Read Form Data</title>
</head>
<body>
<center>
<h1>Using GET Method to Read Form Data</h1>
<ul>
<li><p><b>First Name:</b>
    <%= request.getParameter("first_name")%>
</p></li>
<li><p><b>Last Name:</b>
    <%= request.getParameter("last_name")%>
</p></li>
</ul>
</body>
</html>
```

Following is the content of Hello.htm file:

```
<html>
<body>
<form action="main.jsp" 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>
```

Now let us keep main.jsp and hello.htm in <Tomcat-installation-directory>/webapps/ROOT directory. When you would access <http://localhost:8080/Hello.htm>, below is the actual output of the above form.

First Name:

Last Name:

Try to enter First and Last Name and then click submit button to see the result on your local machine where tomcat is running.

Based on the input provided, it would generate similar result as mentioned in the above examples.

## Passing Checkbox Data to JSP Program

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="main.jsp" 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

Below is main.jsp JSP program to handle input given by web browser for checkbox button.

```
<html>  
<head>  
<title>Reading Checkbox Data</title>  
</head>  
<body>  
<center>  
<h1>Reading Checkbox Data</h1>  
<ul>  
<li><p><b>Maths Flag:</b>  
    <%= request.getParameter("maths")%>  
</p></li>  
<li><p><b>Physics Flag:</b>  
    <%= request.getParameter("physics")%>  
</p></li>  
<li><p><b>Chemistry Flag:</b>  
    <%= request.getParameter("chemistry")%>  
</p></li>  
</ul>  
</body>  
</html>
```

For the above example, it would display following result:

## READING CHECKBOX DATA

---

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

## Reading All Form Parameters:

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.

```
<%@ page import="java.io.*,java.util.*" %>
<html>
<head>
<title>HTTP Header Request Example</title>
</head>
<body>
<center>
<h2>HTTP Header Request Example</h2>
<table width="100%" border="1" align="center">
<tr bgcolor="#949494">
<th>Param Name</th><th>Param Value(s)</th>
</tr>
<%
    Enumeration paramNames = request.getParameterNames();

    while(paramNames.hasMoreElements()) {
        String paramName = (String)paramNames.nextElement();
        out.print("<tr><td>" + paramName + "</td>\n");
        String paramValue = request.getHeader(paramName);
        out.println("<td> " + paramValue + "</td></tr>\n");
    }
%>
</table>
</center>
</body>
</html>
```

Following is the content of Hello.htm:

```
<html>
<body>
<form action="main.jsp" 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 try calling JSP using above Hello.htm, this would generate a result something like as below based on the provided input:

## Reading All Form Parameters

Param Name	Param Value
maths	on
chemistry	on

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