

Introduction to Servlet Request

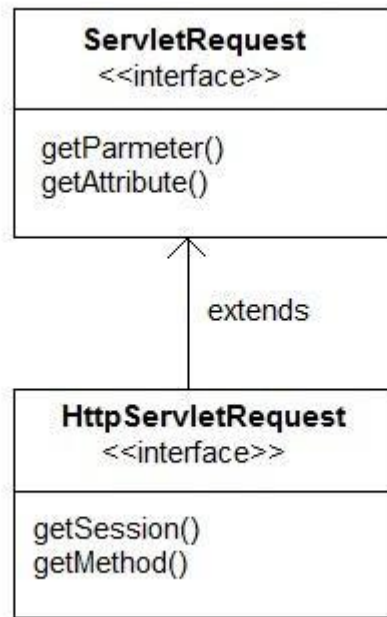
True job of a Servlet is to handle client request. Servlet API provides two important interfaces **javax.servlet.ServletRequest** and **javax.servlet.http.HttpServletRequest** to encapsulate client request. Implementation of these interfaces provide important information about client request to a servlet.

Some Important Methods of ServletRequest

Methods	Description
Object <code>getAttribute(String name)</code>	return attribute set on request object by name
Enumeration <code>getAttributeName()</code>	return an Enumeration containing the names of the attributes available in this request
int <code>getContentLength()</code>	return size of request body
int <code>getContentType()</code>	return media type of request content
ServletInputStream <code>getInputStream()</code>	returns a input stream for reading binary data
String <code>getParameter(String name)</code>	returns value of parameter by name
String <code>getLocalAddr()</code>	returns the Internet Protocol(IP) address of the interface on which the request was received
Enumeration <code>getParameterNames()</code>	returns an enumeration of all parameter names
String[] <code>getParameterValues(String name)</code>	returns an array of String objects containing all of the values the given request parameter has, or null if the parameter does not exist
ServletContext <code>getServletContext()</code>	return the servlet context of current request.
String <code>getServerName()</code>	returns the host name of the server to which the request was sent
int <code>getServerPort()</code>	returns the port number to which the request was sent
boolean <code>isSecure()</code>	returns a boolean indicating whether this request was made using a secure channel, such as HTTPS.
void <code>removeAttribute(String name)</code>	removes an attribute from this request
void <code>setAttribute(String name, Object o)</code>	stores an attribute in this request.

HttpServletRequest interface

HttpServletRequest interface adds the methods that relates to the **HTTP** protocol.



Some important methods of HttpServletRequest

Methods	Description
String getContextPath()	returns the portion of the request URI that indicates the context of the request
Cookies getCookies()	returns an array containing all of the Cookie objects the client sent with this request
String getQueryString()	returns the query string that is contained in the request URL after the path
HttpSession getSession()	returns the current HttpSession associated with this request or, if there is no current session and create is true, returns a new session
String getMethod()	Returns the name of the HTTP method with which this request was made, for example, GET, POST, or PUT.
Part getPart(String name)	gets the Part with the given name
String getPathInfo()	returns any extra path information associated with the URL the client sent when it made this request.
String getServletPath()	returns the part of this request's URL that calls the servlet

Example demonstrating Servlet Request

In this example, we will show how a parameter is passed to a Servlet in a request object from HTML page.

index.html

```
<form method="post" action="check">
Name <input type="text" name="user" >
<input type="submit" value="submit">
</form>
```

web.xml

```
<servlet>
    <servlet-name>check</servlet-name>
    <servlet-class>MyServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>check</servlet-name>
    <url-pattern>/check</url-pattern>
</servlet-mapping>
```

MyServlet.java

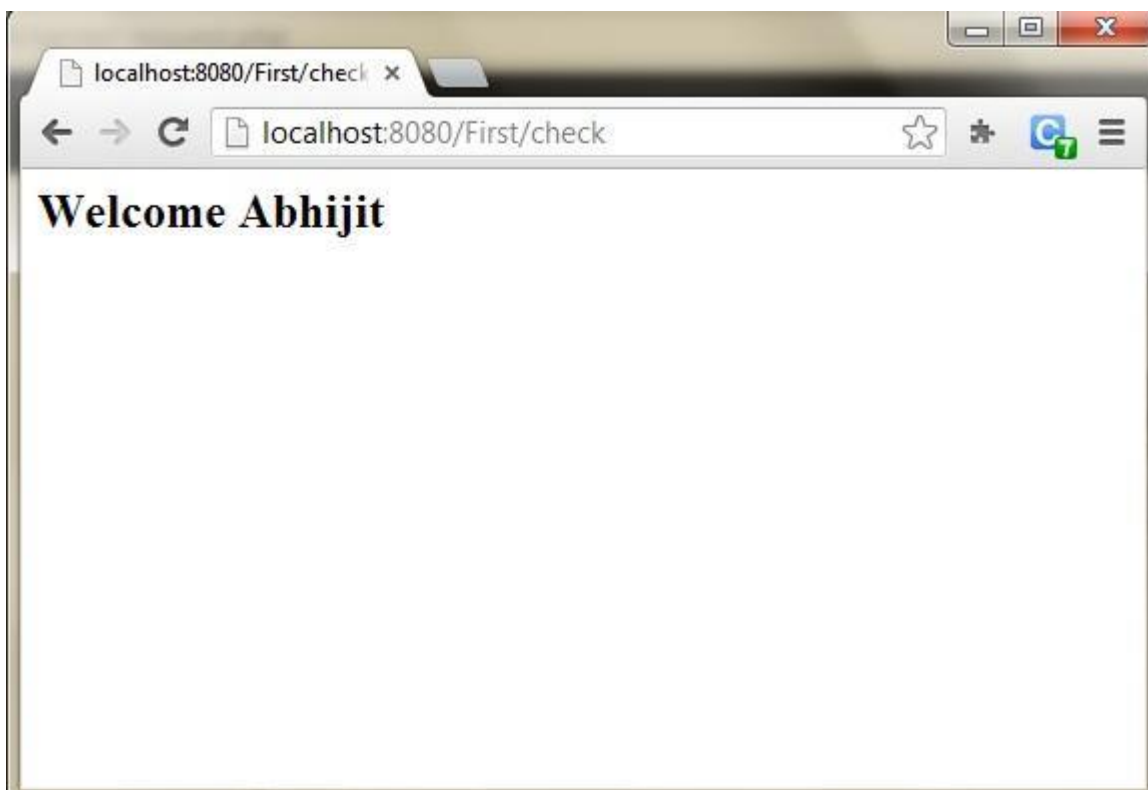
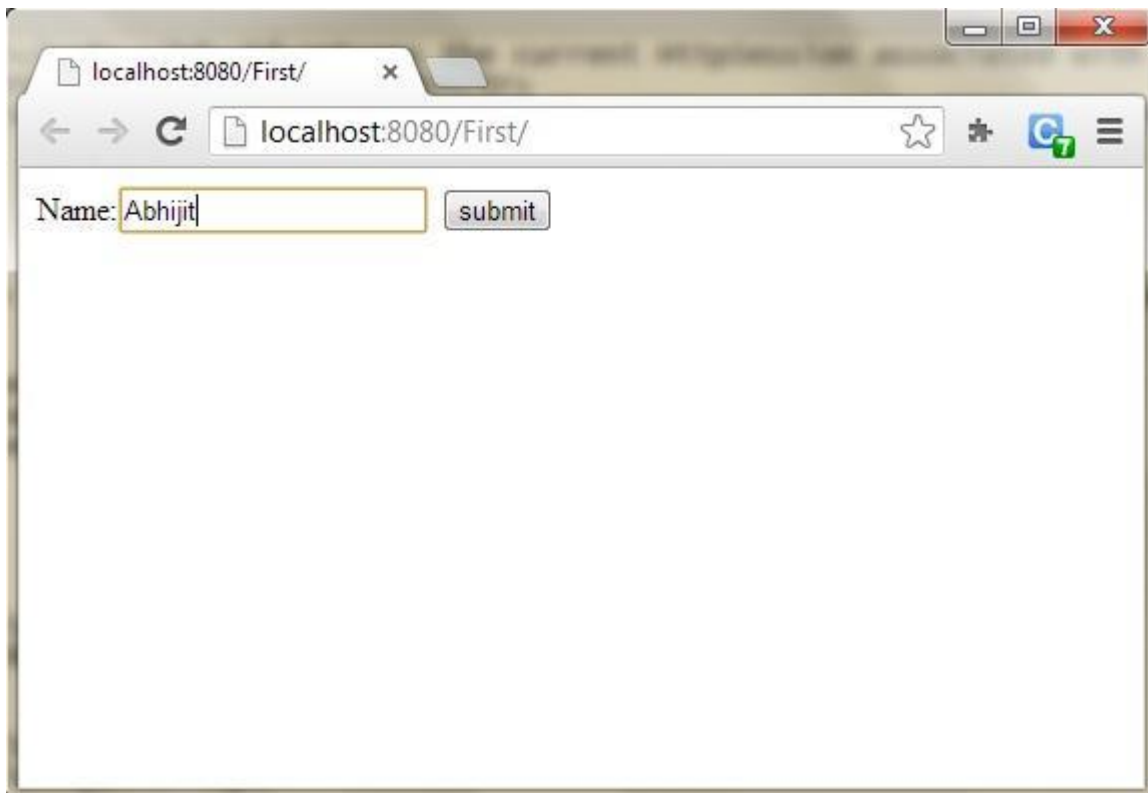
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MyServlet extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {

            String user=request.getParameter("user");
            out.println("<h2> Welcome "+user+"</h2>");
        } finally {
            out.close();
        }
    }
}
```

Output :



Introduction to Servlet Response

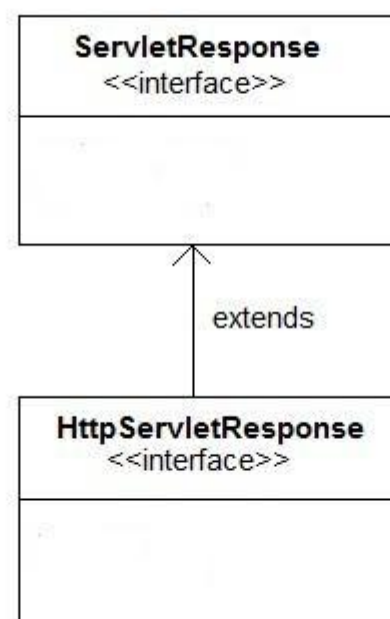
Servlet API provides two important interfaces **ServletResponse** and **HttpServletResponse** to assist in sending response to client.

Some Important Methods of ServletResponse

Methods	Description
<code>PrintWriter getWriter()</code>	returns a <code>PrintWriter</code> object that can send character text to the client.
<code>void setBufferSize(int size)</code>	Sets the preferred buffer size for the body of the response
<code>void setContentLength(int len)</code>	Sets the length of the content body in the response In HTTP servlets, this method sets the HTTP Content-Length header
<code>void setContentType(String type)</code>	sets the content type of the response being sent to the client before sending the respond.
<code>void setBufferSize(int size)</code>	sets the preferred buffer size for the body of the response.
<code>boolean isCommitted()</code>	returns a boolean indicating if the response has been committed
<code>void setLocale(Locale loc)</code>	sets the locale of the response, if the response has not been committed yet.

HttpServletResponse Interface

HttpServletResponse interface adds the methods that relates to the **HTTP** response.



Some Important Methods of HttpServletResponse

Methods	Description
<code>void addCookie(Cookie cookie)</code>	adds the specified cookie to the response.
<code>void sendRedirect(String location)</code>	Sends a temporary redirect response to the client using the specified redirect location URL and clears the buffer
<code>int getStatus()</code>	gets the current status code of this response
<code>String getHeader(String name)</code>	gets the value of the response header with the given name.
<code>void setHeader(String name, String value)</code>	sets a response header with the given name and value
<code>void setStatus(int sc)</code>	sets the status code for this response
<code>void sendError(int sc, String msg)</code>	sends an error response to the client using the specified status and clears the buffer