Introduction to Servlet Request

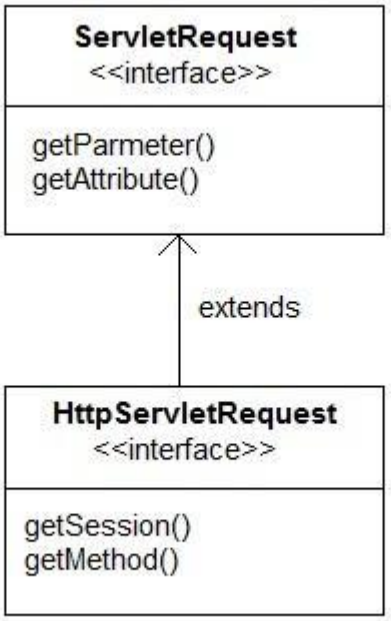
True job of a Servlet is to handle client request. Servlet API provides two important interfaces **javax.servlet.ServletRequest** and **javax.servlet.http.HttpServletRequest** to encapsulate client request. Implementation of these interfaces provide important information about client request to a servlet.

Some Important Methods of ServletRequest

Methods	Description
Object `getAttribute(String name)`	return attribute set on request object by name
Enumeration `getAttributeName()`	return an Enumeration containing the names of the attributes available inthis request
int `getContentLength()`	return size of request body
int `getContentType()`	return media type of request content
ServletInputStream `getInputStream()`	returns a input stream for reading binary data
String `getParameter(String name)`	returns value of parameter by name
String `getLocalAddr()`	returns the Internet Protocol(IP) address of the interface on which the request was received
Enumeration `getParameterNames()`	returns an enumeration of all parameter names
String[] `getParameterValues(String name)`	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 `getServletContext()`	return the servlet context of current request.
String `getServerName()`	returns the host name of the server to which the request was sent
int `getServerPort()`	returns the port number to which the request was sent
boolean `isSecure()`	returns a boolean indicating whether this request was made using a secure channel, such as HTTPS.
void `removeAttribute(String name)`	removes an attribute from this request
void `setAttribute(String name, Object o)`	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

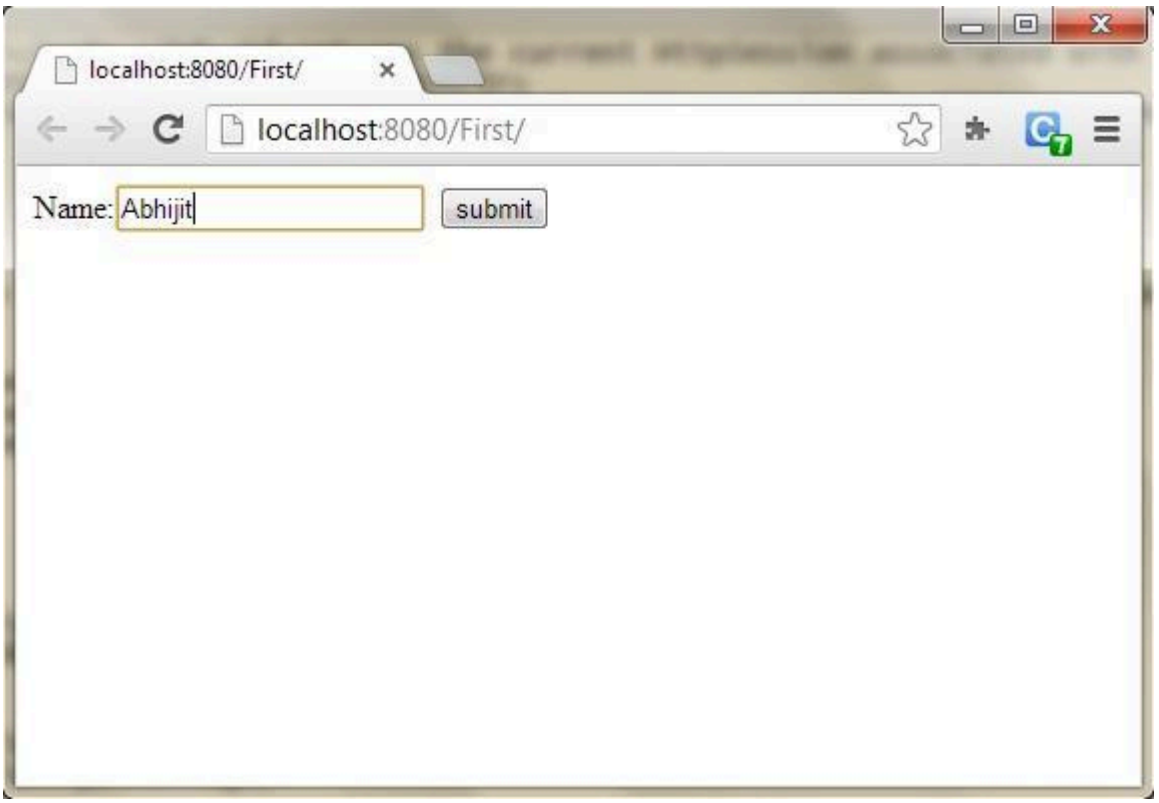
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

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

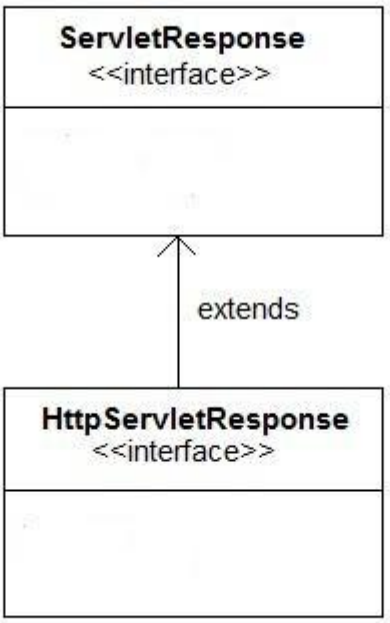
Servlet API provides two important interfaces **ServletResponse** and **HttpServletResponse** to assist in sending response to client.

Some Important Methods of ServletResponse

Methods	Description
PrintWriter `getWriter()`	returns a PrintWriter object that can send character text to the client.
void `setBufferSize(int size)`	Sets the preferred buffer size for the body of the response
void `setContentLength(int len)`	Sets the length of the content body in the response In HTTP servlets, this method sets the HTTP Content-Length header
void `setContentType(String type)`	sets the content type of the response being sent to the client before sending the respond.
void `setBufferSize(int size)`	sets the preferred buffer size for the body of the response.
boolean `isCommitted()`	returns a boolean indicating if the response has been committed
void `setLocale(Locale loc)`	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
void `addCookie(Cookie cookie)`	adds the specified cookie to the response.
void `sendRedirect(String location)`	Sends a temporary redirect response to the client using the specified redirect location URL and clears the buffer
int `getStatus()`	gets the current status code of this response
String `getHeader(String name)`	gets the value of the response header with the given name.
void `setHeader(String name, String value)`	sets a response header with the given name and value
void `setStatus(int sc)`	sets the status code for this response
void `sendError(int sc, String msg)`	sends an error response to the client using the specified status and clears the buffer