



SERVLET TECHNOLOGY



Gmail for Work

Look more professional with
custom Gmail from Google

Start free trial



[SEE THE INDEX](#)



Introduction to Request Dispatcher

RequestDispatcher is an interface, implementation of which defines an object which can dispatch request to any resources (such as HTML, Image, JSP, Servlet) on the server.

Methods of RequestDispatcher

RequestDispatcher interface provides two important methods

Methods	Description
void <code>forward(ServletRequest request, ServletResponse response)</code>	forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server
void <code>include(ServletRequest request, ServletResponse response)</code>	includes the content of a resource (servlet, JSP page, HTML file) in the response

How to get an Object of RequestDispatcher

`getRequestDispatcher()` method of **ServletRequest** returns the object of **RequestDispatcher**.

```
RequestDispatcher rs = request.getRequestDispatcher("hello.html");  
rs.forward(request, response);
```

ServletRequest object → `request`

resource name → `"hello.html"`

```
RequestDispatcher rs = request.getRequestDispatcher("hello.html");  
  
rs.forward(request, response);
```

↑ **forward the request and response to "hello.html" page**

OR

```
RequestDispatcher rs = request.getRequestDispatcher("hello.html");  
rs.include(request,response);
```

The diagram shows two lines of Java code with red arrows pointing to specific parts. The first line is `RequestDispatcher rs = request.getRequestDispatcher("first.html");`. An arrow points from the text "ServletRequest object" to the `request` parameter. Another arrow points from the text "Resource name" to the string `"first.html"`. The second line is `rs.include(request, response);`. An arrow points from the text "include the response of 'first.html' page in current servlet response" to the `include` method call.

```
RequestDispatcher rs = request.getRequestDispatcher("first.html");  
  
rs.include(request, response);
```

Example demonstrating usage of RequestDispatcher

In this example, we will show you how RequestDispatcher is used to **forward** or **include** response of a resource in a Servlet. Here we are using **index.html** to get username and password from the user, **Validate** Servlet will validate the password entered by the user, if the user has entered "studytonight" as password, then he will be forwarded to **Welcome** Servlet else the user will stay on the index.html page and an error message will be displayed.

Files to be created :

- **index.html** will have form fields to get user information.
- **Validate.java** will validate the data entered by the user.
- **Welcome.java** will be the welcome page.
- **web.xml** , the deployment descriptor.

index.html

```
<form method="post" action="Validate">  
Name:<input type="text" name="user" /><br/>  
Password:<input type="password" name="pass" ><br/>  
<input type="submit" value="submit">  
</form>
```

Validate.java

```

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

public class Validate 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 name = request.getParameter("user");
            String password = request.getParameter("pass");

            if(password.equals("studytonight"))
            {
                RequestDispatcher rd = request.getRequestDispatcher("Welcome");
                rd.forward(request, response);
            }
            else
            {
                out.println("<font color='red'><b>You have entered incorrect password</b></font>");
                RequestDispatcher rd = request.getRequestDispatcher("index.html");
                rd.include(request, response);
            }
        }finally {
            out.close();
        }
    }
}

```

Welcome.java

```

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

public class Welcome extends HttpServlet {

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

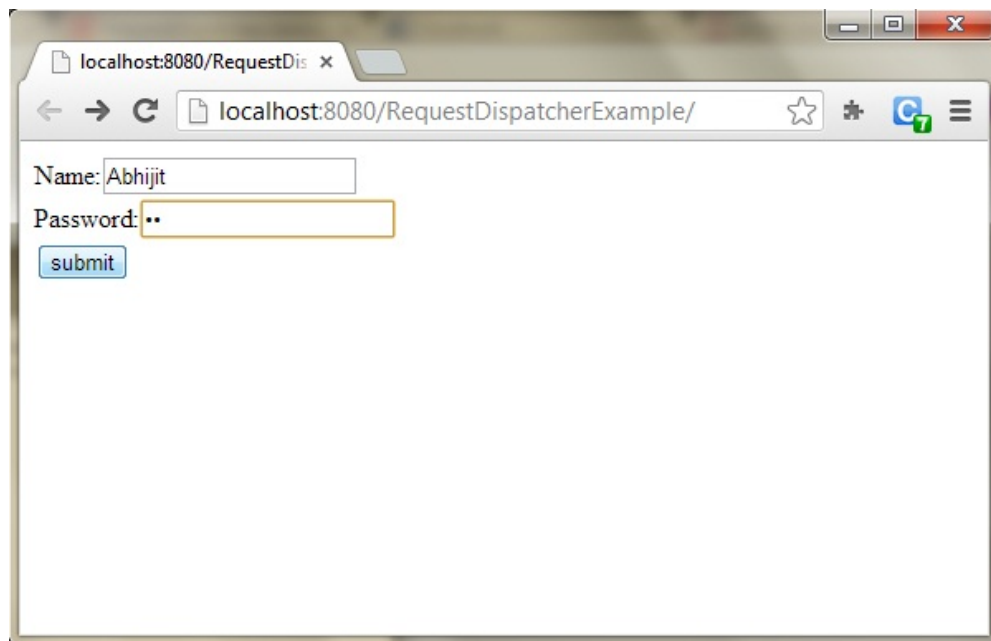
            out.println("<h2>Welcome user</h2>");
        } finally {
            out.close();
        }
    }
}

```

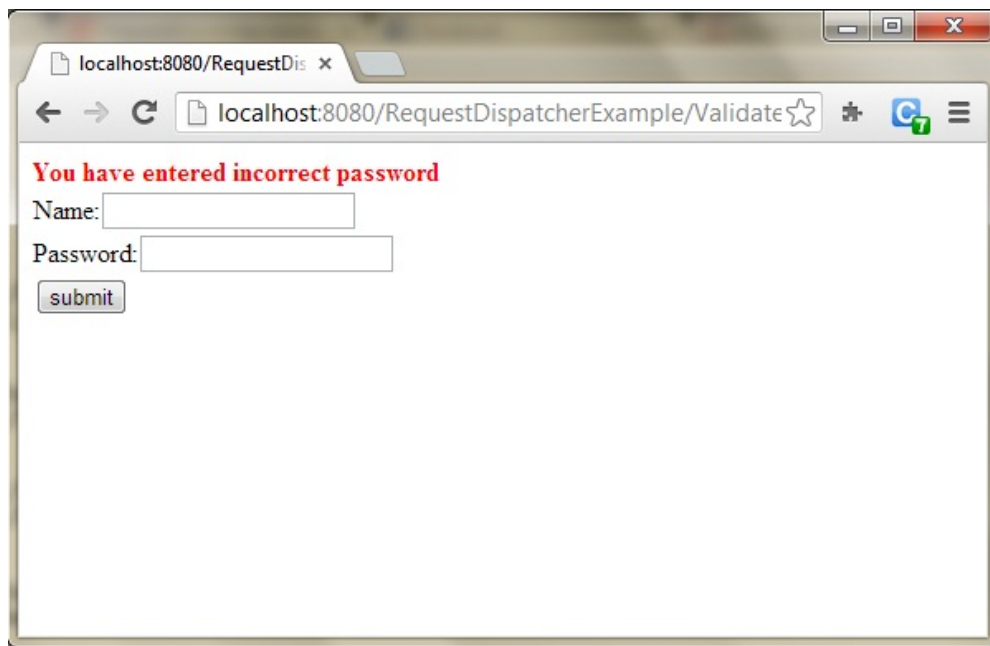
web.xml

```
<web-app>
  <servlet>
    <servlet-name>Validate</servlet-name>
    <servlet-class>Validate</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>Welcome</servlet-name>
    <servlet-class>Welcome</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Validate</servlet-name>
    <url-pattern>/Validate</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>Welcome</servlet-name>
    <url-pattern>/Welcome</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>
</web-app>
```

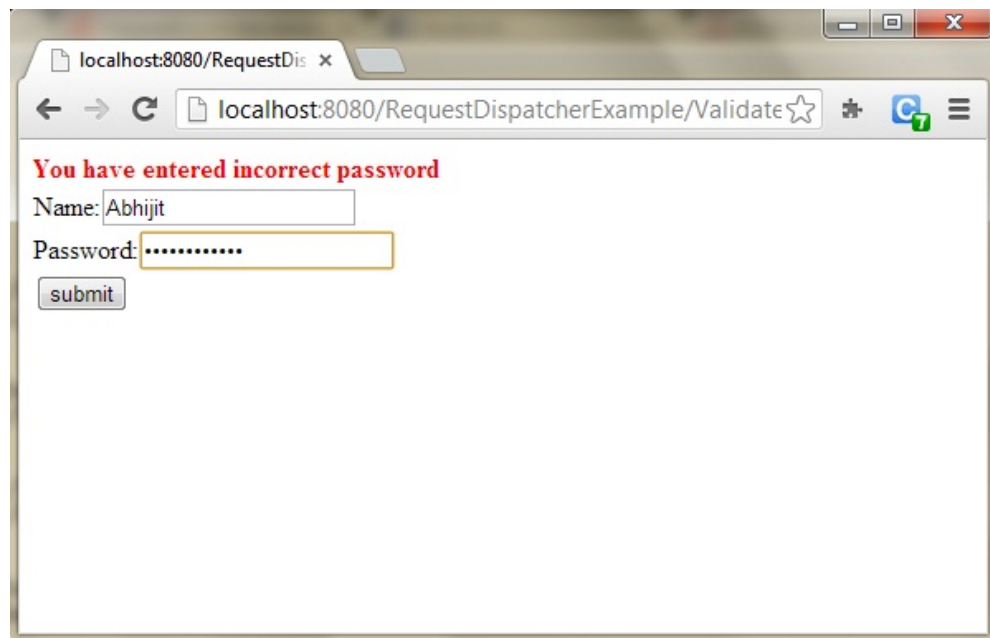
This will be the first screen. You can enter your Username and Password here.



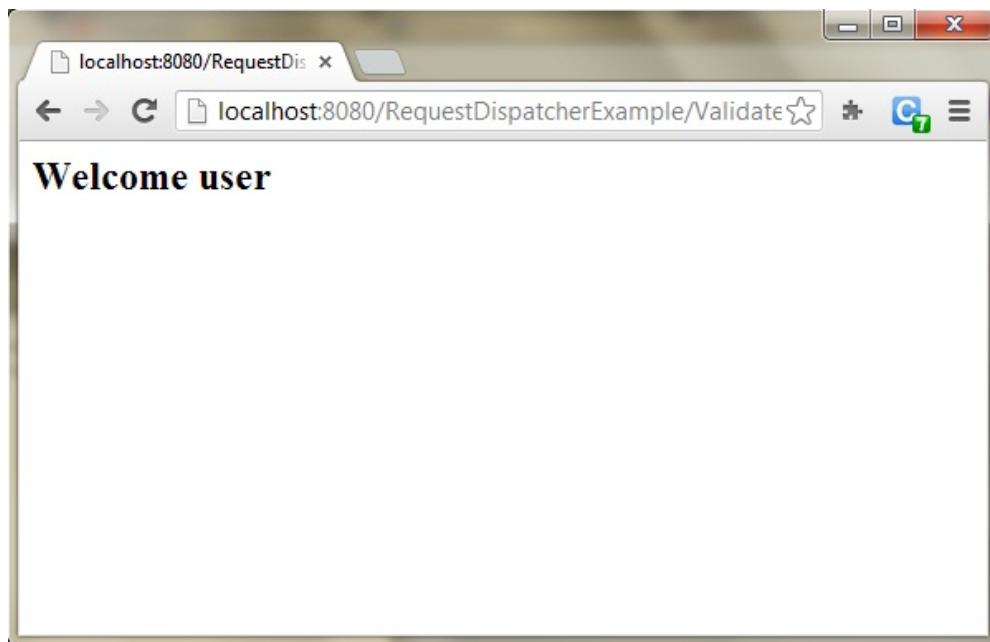
When you click on Submit, Password will be validated, if it is not 'studytonight' , error message will be displayed.



Enter any Username, but enter 'studytontight' as password.



Password will be successfully validated and you will be directed to the Welcome Servlet.



[← Prev \(servlet-response\)](#)

[Next → \(sendredirect-method\)](#)

© Studytonight 2016 · ❤ Handcrafted with Love

About Us (<http://www.studytonight.com/about>) · Suggest (<http://www.studytonight.com/suggest>) · Terms (<http://www.studytonight.com/terms>) · Contact Us (<http://www.studytonight.com/contact>) · Collaborate (<http://www.studytonight.com/collaborate/>) · Authors (<http://www.studytonight.com/authors/>) · Blog (<http://blog.studytonight.com/>)

Java (Hiring)



₹250-600 per hour No Experience Required.

