sendRedirect() method redirects the response to another resource. This method actually makes the client(browser) to create a new request to get to the resource. The client can see the new url in the browser.

**sendRedirect()** also used to navigate from one servlet to another servlet.

Sample : Navigation from one servlet to another

**Web.xml**

<web-app>

<servlet>

<servlet-name>first</servlet-name>

<servlet-class>FirstServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>first</servlet-name>

<url-pattern>/first</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>second</servlet-name>

<servlet-class>SecondServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>second</servlet-name>

<url-pattern>/second</url-pattern>

</servlet-mapping>

</web-app>

**FirstServlet**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class FirstServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

try {

response.sendRedirect("http://localhost:8090/SendRedirect/second");

}finally {

out.close();

}

}

}

**Second servlet**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class SecondServlet extends HttpServlet {

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

ServletConfig sc=getServletConfig();

out.println("This is the second servlet");

}

}

**Difference between sendRedirect and Forward**

**sendRedirect :** send request to different server

**Forward :** send request to same server