

# **Java & JEE Training**

**Day 28 – Servlets - Part 4**

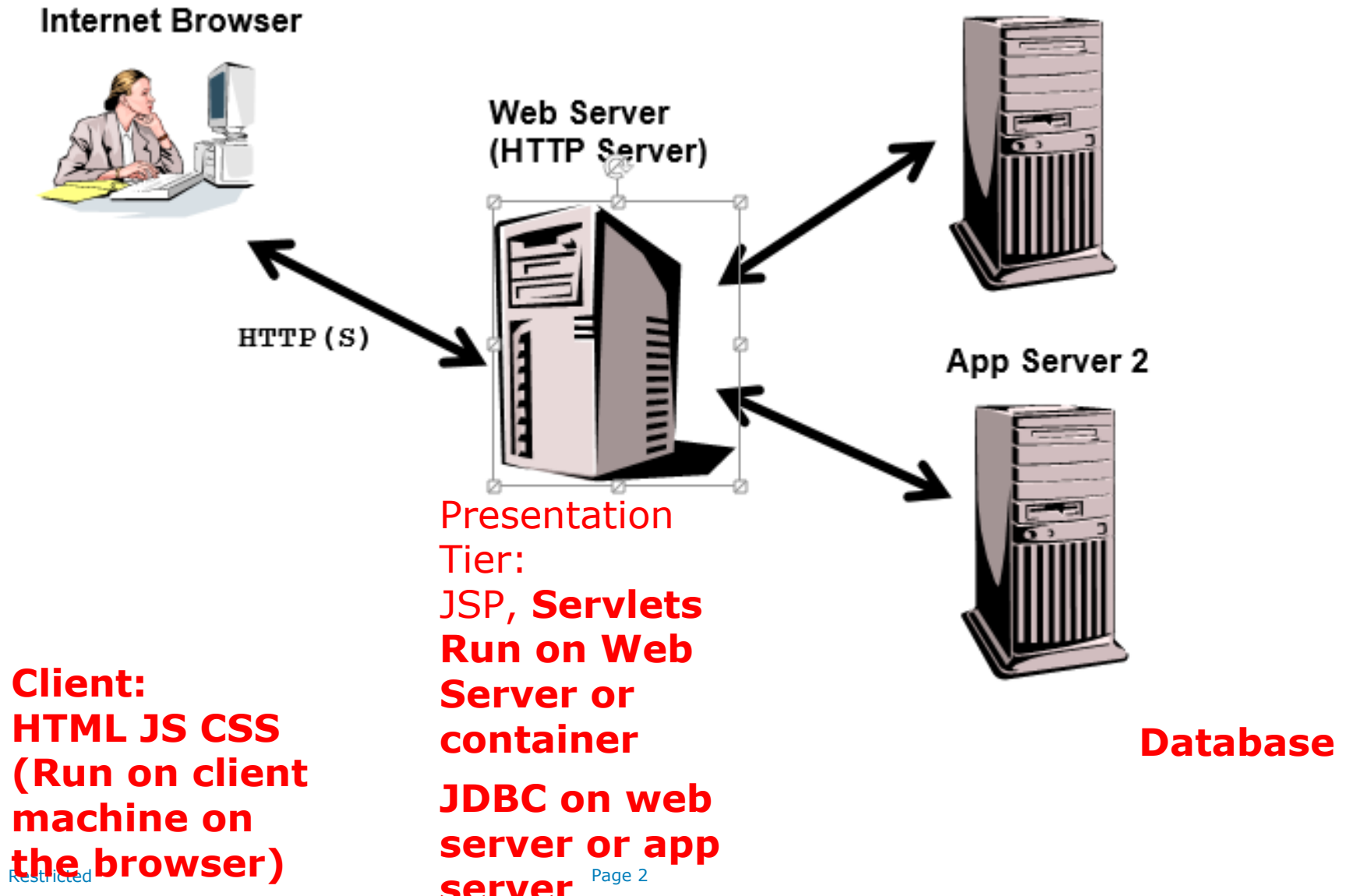
**MindsMapped Consulting**

# Review of previous session...

---

...

# Web Server and Application Server



## Demo of complete Web app flow...

---

- Form which takes two fields... Student ID and Student Name...
- Output should show whether that row exists or not, and if it exists, how many records in the table match the criteria (id, name)
- If match found, then print login successful, else login failed.

# **Agenda of today's session**

# Agenda...

---

- SQL Injection continued... + Exercises
- Validating forms
- Redirecting requests to other sources

# **Validation in web apps**

# Validating forms using JavaScript..

---

## Best practice

- MUST always validate both on client side as well as server side.
- Validation in the client side is done using JavaScript.
- Validation in the server side is done using Java.
- Demo of frontend and backend validation.
- Example of JS validation.
- Check if value entered is a number or not:  

```
if (data === parseInt(data, 10))  
    alert("data is integer")  
else  
    alert("data is not an integer");
```



## More JS Validation Examples....

---

- Let us look at some validation examples...

## Check for non-empty field...

---

```
// If the length of the element's string is 0 then display helper message
function required(inputtx) {
    if (inputtx.value.length == 0) {
        alert("message");
        return false;
    }
    return true;
}
```

## Check for restricting length of input field...

---

```
function lengthRange(inputtxt, minlength, maxlength)
{
    var userInput = inputtxt.value;
    if(userInput.length >= minlength && userInput.length <= maxlength){
        return true;
    }
    else{
        alert("Please input between " +minlength+ " and " +maxlength+ "
characters");
        return false;
    }
}
```

## Reference for more validation examples...

---

- A good reference for form validation..

<http://www.w3resource.com/javascript/form/javascript-form-validation.php>

**Redirecting requests to other sources**

**MindsMapped Consulting**

# Redirecting Requests to Other Resources

---

- Servlet `RedirectServlet`
  - Redirects the request to a different resource
  - Why not just have links to those resources?
    - Could determine browser type and choose link accordingly
    - Could add parameters before redirecting (those sent originally are included automatically)
  - `response.sendRedirect(URL);`
    - redirects to that URL
    - immediately terminates current program
    - Use relative paths whenever possible to make Website portable

## RedirectServlet redirecting requests to other resources.

```
1  // RedirectServlet.java
2  // Redirecting a user to a different web page.
3
4  import javax.servlet.*;
5  import javax.servlet.http.*;
6  import java.io.*;
7
8  public class RedirectServlet extends HttpServlet {
9
10     // process "get" request from client
11     protected void doGet( HttpServletRequest request,
12         HttpServletResponse response )
13         throws ServletException, IOException
14     {
15         String location = request.getParameter( "page" );
16
17         if ( location != null )
18
19             if ( location.equals( "deitel" ) )
20                 response.sendRedirect( "http://www.deitel.com" )
21             else
22                 if ( location.equals( "welcome1" ) )
23                     response.sendRedirect( "welcome1" );
24     }
```

Obtains the page parameter from the request.

Redirects the request to www.deitel.com.

Redirects the request to the servlet welcomeServlet.

# CRUD...

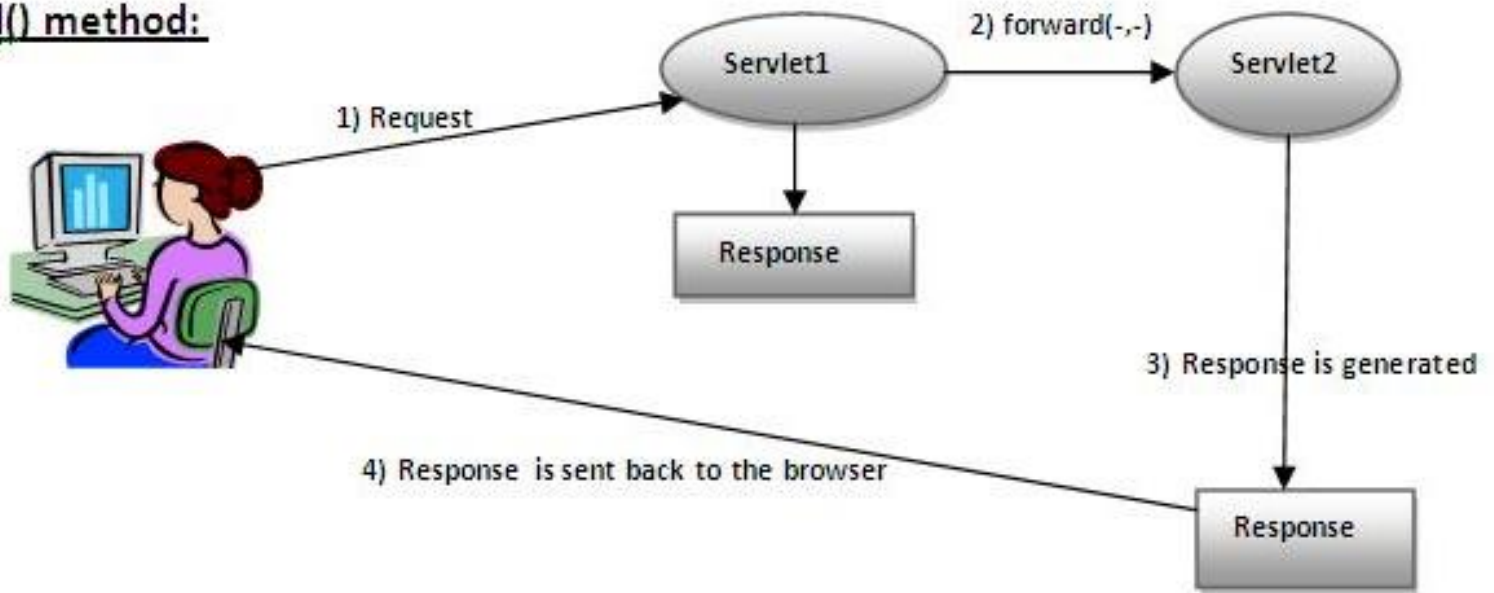
---

- Select
  - Update
  - Insert
  - Delete
- 
- 1. /Hello?op=Insert OR /Hello?op=Delete
  - 2. /Hello/Insert OR /Hello/Delete



# RequestDispatcher forward()

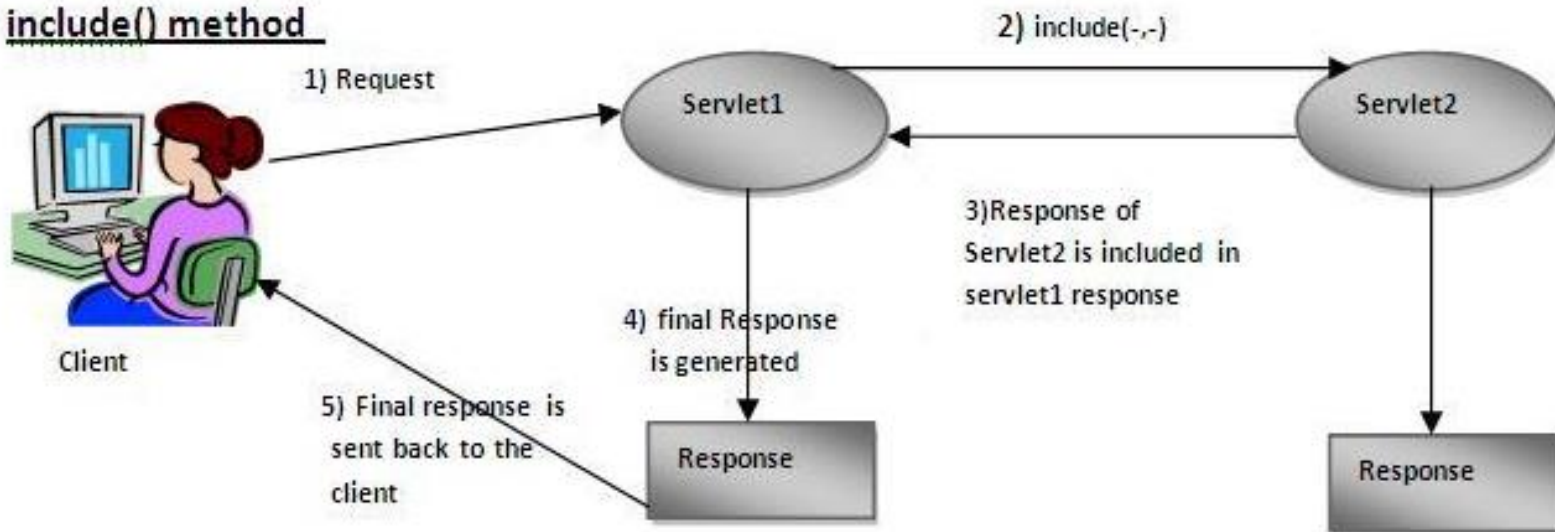
forward() method:



```
RequestDispatcher rd=request.getRequestDispatcher("<another servlet>");  
rd.forward(request, response);
```

# RequestDispatcher include()

## include() method



```
out.print("<Your message to include goes here>");  
RequestDispatcher rd=request.getRequestDispatcher("nameOfHTMLorJSP");  
rd.include(request, response);
```

# Request Dispatcher Demo

