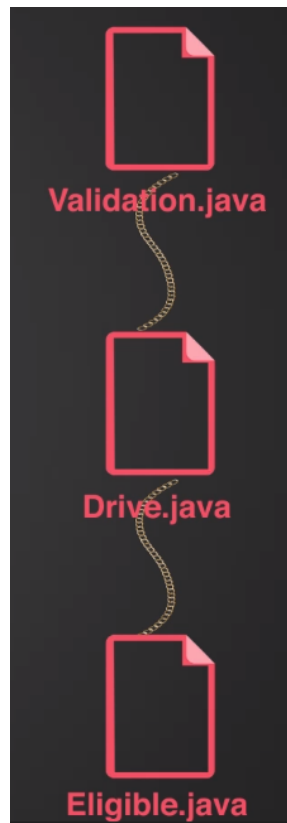


## Day - 9

### forward( ) v/s include( )

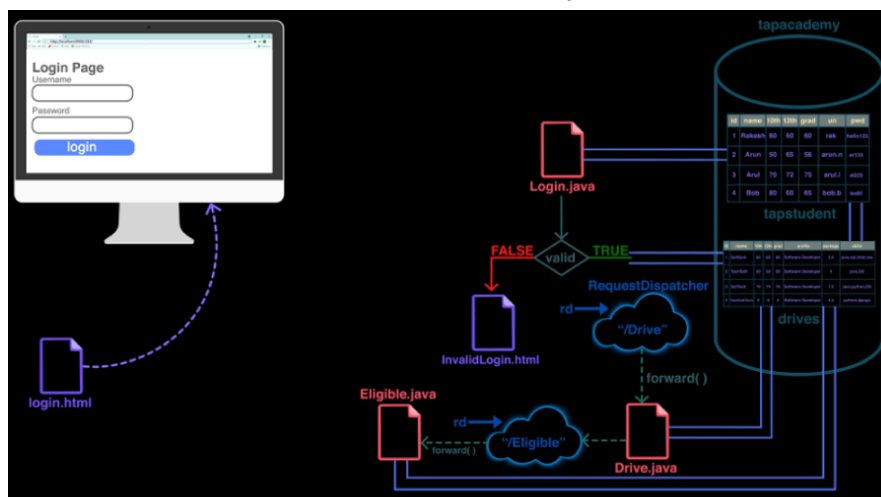
In the previous session, we had learnt about **Single Level Servlet Chaining**. In this session let us try to understand **Two Level Servlet Chaining**.

In the last session, **Validation.java** servlet had chained to **Drive.java** servlet, now we will introduce a new servlet called **Eligible.java** whose duty is to display all the drives the candidate is eligible for and Drive.java servlet will chain to Eligible.java servlet.



We know that whenever a request from one resource should be forwarded to another resource, we have to make use of the **RequestDispatcher** object and the RequestDispatcher object will forward the request to **Eligible.java** from **Drive.java**.

**Eligible.java** will now further connect to the database and display all the drives that the user is eligible to.



Let us now try to implement this in our code-

Create a servlet **Eligible.java** in the **com.tap.student** package and type in the following-

```
package com.tap.student;

public class Eligible extends HttpServlet {
    Connection con = null;
    String url = "jdbc:mysql://localhost:3306/tapacademy";
    String un = "root";
    String pwd = "root";

    @Override
    public void init() throws ServletException {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(url, un, pwd);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        String username = req.getParameter("username");
        String password = req.getParameter("password");

        PrintWriter writer = resp.getWriter();

        try {
            String query1 = "select * from tapstudent where un = ? and pwd = ?";
            PreparedStatement pstmt1 = con.prepareStatement(query1);
            pstmt1.setString(1, username);
            pstmt1.setString(2, password);
            ResultSet res1 = pstmt1.executeQuery();
            res1.next();

            int ten = res1.getInt(3);
            int twe = res1.getInt(4);
            int grad = res1.getInt(5);

            String query2 = "select * from drives where 10th <= ? "
```

```

        + "and 12th <= ? and grad <= ?";

PreparedStatement pstmt2 = con.prepareStatement(query2);
pstmt2.setInt(1, ten);
pstmt2.setInt(2, tve);
pstmt2.setInt(3, grad);
ResultSet res2 = pstmt1.executeQuery();

writer.println("<h3>" + res1.getString(2) + ""
        + "drives you are eligible for:</h3>");

writer.println("<table border=\"1\">\r\n"
        + "\r\n"
        + "    <tr>\r\n"
        + "        <th>Id</th>\r\n"
        + "        <th>Name</th>\r\n"
        + "        <th>10th</th>\r\n"
        + "        <th>12th</th>\r\n"
        + "        <th>Grad</th>\r\n"
        + "        <th>Profile</th>\r\n"
        + "        <th>Package</th>\r\n"
        + "        <th>Skills</th>    \r\n"
        + "    </tr>");

while(res2.next()==true){
    int id = res2.getInt(1);
    String name = res2.getString(2);
    int ten1 = res2.getInt(3);
    int tve1 = res2.getInt(4);
    int grad1 = res2.getInt(5);
    String profile = res2.getString(6);
    float pac = res2.getFloat(7);
    String skills = res2.getString(8);

    writer.println("<tr>\r\n"
        + "        <td>" + id + "</td>\r\n"
        + "        <td>" + name + "</td>\r\n"
        + "        <td>" + ten1 + "</td>\r\n"
        + "        <td>" + tve1 + "</td>\r\n"
        + "        <td>" + grad1 + "</td>\r\n"
        + "        <td>" + profile + "</td>\r\n"
        + "        <td>" + pac + "</td>\r\n"
        + "        <td>" + skills + "</td>\r\n"
        + "    </tr>");
}
writer.println("</table>");

```

```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Now that the Eligible servlet is ready, add the code shown below in the Drive servlet, so that the Drive servlet can chain to Eligible servlet with the help of **RequestDispatcher** object.

```
req.getRequestDispatcher("/eligible").forward(req, resp);
```

And also register the Eligible servlet in the deployment descriptor(web.xml)

```

<servlet>
    <servlet-name>Eligible</servlet-name>
    <servlet-class>com.tap.student.Eligible</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>Eligible</servlet-name>
    <url-pattern>/eligible</url-pattern>
</servlet-mapping>

```

Once we deploy the server in the browser, we get the following output:

### Sagar drives you are eligible for:

Id	Name	10th	12th	Grad	Profile	Package	Skills
1	Softsoft	60	60	60	software developer	3.8	java,sql,html,css
2	Techsoft	50	50	50	software developer	4.0	java,DS
4	Techsofttech	0	0	0	softwaredeveloper	4.2	python,django

But we only got the result from Eligible servlet and not from Drive servlet, even though Drive.java had only chained to Eligible.java.

#### Why did this happen?

To understand this, we must first understand how the RequestDispatcher object works.

The RequestDispatcher has two methods-

- **forward( )**
- **include( )**

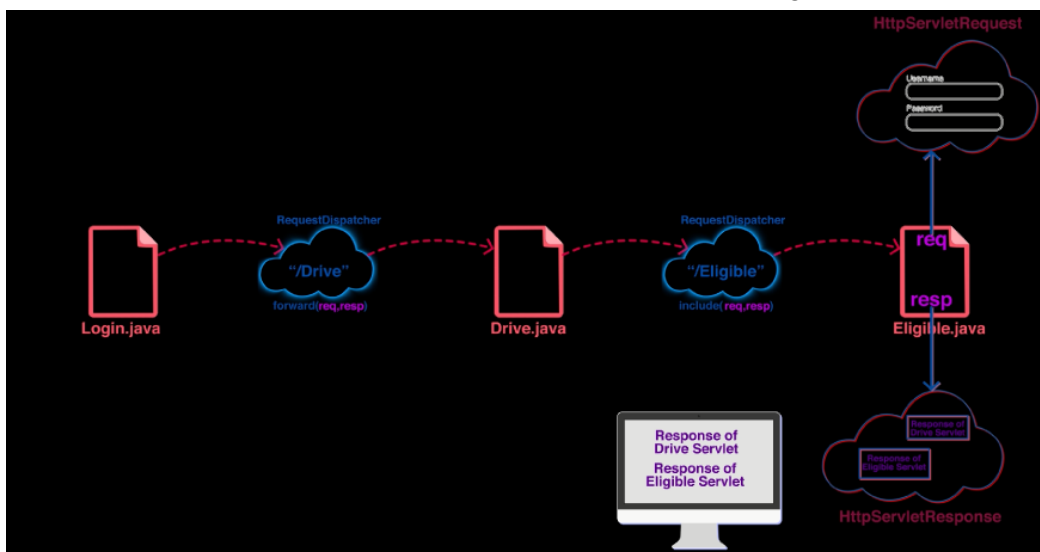
We have been using the forward( ), and the duty of the forward( ) is to forward the request from one resource to another. But the forward( ) before forwarding the request, it will delete the response of the previous

resource. So that is the reason we got the response of only Eligible servlet.



But we wanted the output of both **Eligible.java** and **Drive.java** servlets.

For that we have to make use of the 2nd method of `RequestDispatcher` i.e., `include()` and the `include()` will delete the responses of the previous servlets while forwarding the request.



So let us try to change the `forward()` to `include()` in `Drive.java` file and see if we get the response of both the servlets.

Once we deploy the server, we will get the expected output.

### Drives conducted at TapAcademy:

<b>Id</b>	<b>Name</b>	<b>10th</b>	<b>12th</b>	<b>Grad</b>	<b>Profile</b>	<b>Package</b>	<b>Skills</b>
1	Softsoft	60	60	60	software developer	3.8	java,sql,html,css
2	Techsoft	50	50	50	software developer	4.0	java,DS
3	Softech	70	70	70	software developer	7.5	java,python,DS
4	Techsofttech	0	0	0	softwaredeveloper	4.2	python,django

### Sagar drives you are eligible for:

<b>Id</b>	<b>Name</b>	<b>10th</b>	<b>12th</b>	<b>Grad</b>	<b>Profile</b>	<b>Package</b>	<b>Skills</b>
1	Softsoft	60	60	60	software developer	3.8	java,sql,html,css
2	Techsoft	50	50	50	software developer	4.0	java,DS
4	Techsofttech	0	0	0	softwaredeveloper	4.2	python,django

And if we want to include the response of Validation servlet too, then we have to change the forward( ) to include( ) in Validation.java

### Welcome to TapAcademy:

### Drives conducted at TapAcademy:

<b>Id</b>	<b>Name</b>	<b>10th</b>	<b>12th</b>	<b>Grad</b>	<b>Profile</b>	<b>Package</b>	<b>Skills</b>
1	Softsoft	60	60	60	software developer	3.8	java,sql,html,css
2	Techsoft	50	50	50	software developer	4.0	java,DS
3	Softech	70	70	70	software developer	7.5	java,python,DS
4	Techsofttech	0	0	0	softwaredeveloper	4.2	python,django

### Sagar drives you are eligible for:

<b>Id</b>	<b>Name</b>	<b>10th</b>	<b>12th</b>	<b>Grad</b>	<b>Profile</b>	<b>Package</b>	<b>Skills</b>
1	Softsoft	60	60	60	software developer	3.8	java,sql,html,css
2	Techsoft	50	50	50	software developer	4.0	java,DS
4	Techsofttech	0	0	0	softwaredeveloper	4.2	python,django