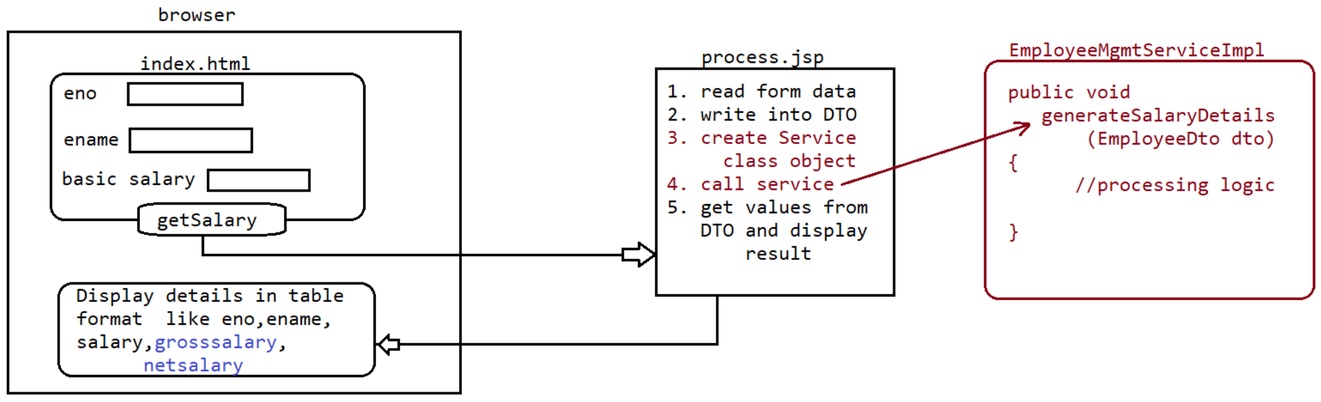

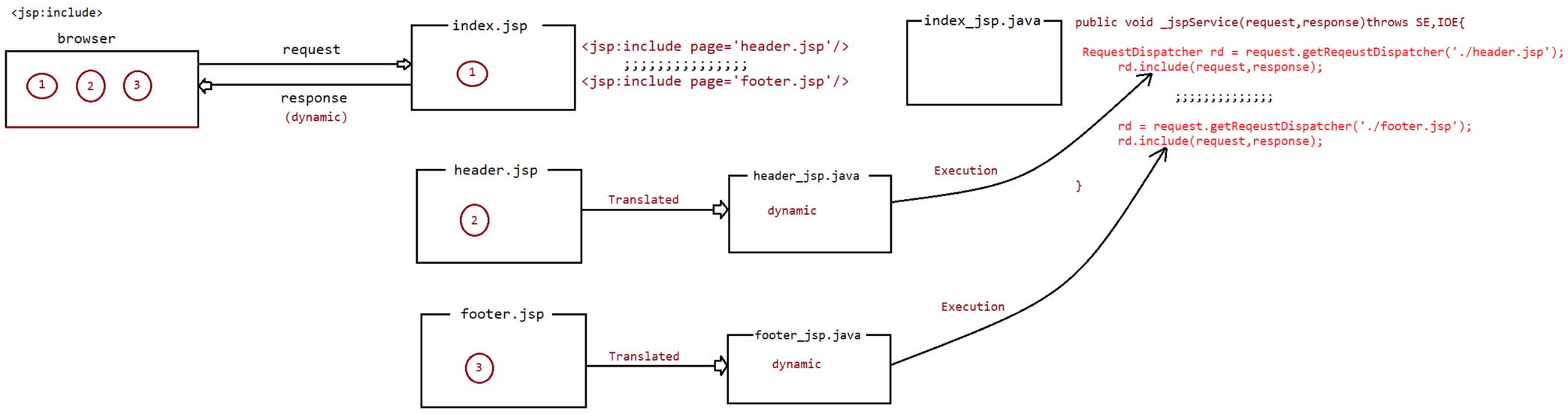

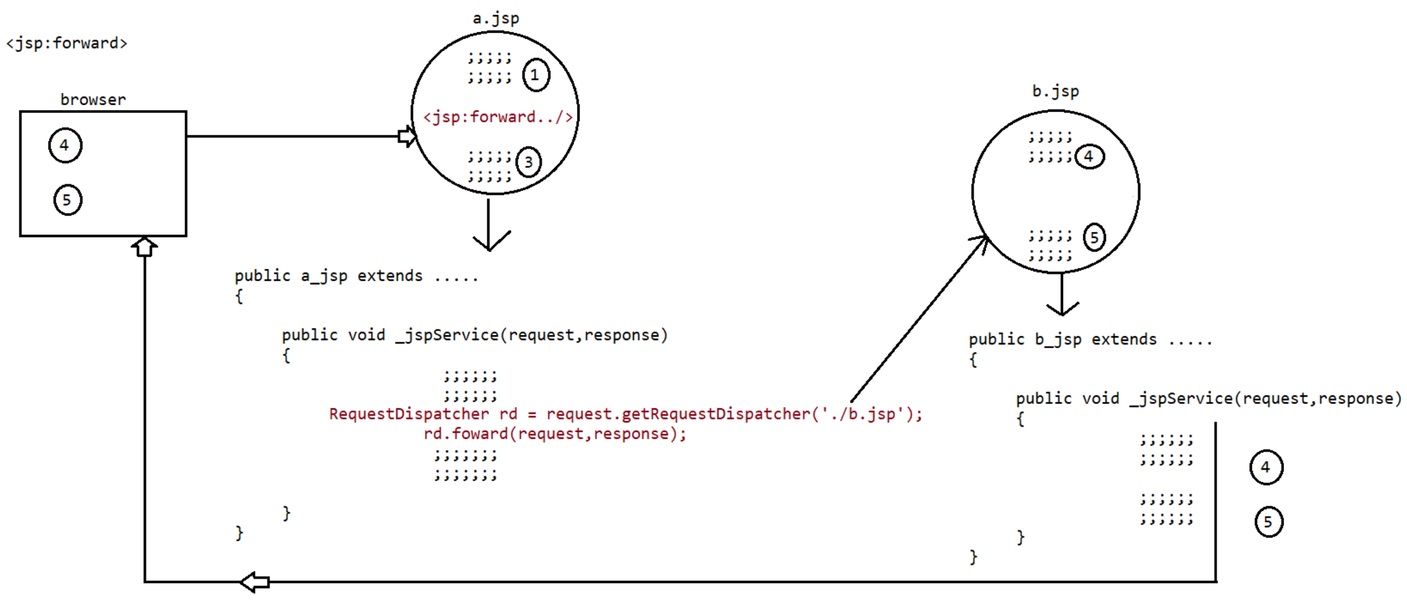

<jsp:param>

a.jsp

```
<b>Start of JSP</b>
    <%
        float billAmount = 300.0f+(300.0f * 0.03f);
    %>
    <br>
        <jsp:forward page="b.jsp">
            <jsp:param value="divya" name="customerName"/>
            <jsp:param value="<%=billAmount%>" name="billAmount"/>
        </jsp:forward>
    <br>
<b>End of JSP</b>
```

request → customerName=divya, billAmount=3000.0f

b.jsp

```
<body>
    <h1>
        <b>from b.jsp</b><br/>
        <%= new java.util.Date() %><br/><br/>
    </h1>

    <h1 style='color: blue; text-align: center;'>
        CustomerName is ::<%= request.getParameter("customerName") %><br />
        BillAmount is ::<%= request.getParameter("billAmount") %><br />
    </h1>
</body>
</html>
```

JSP Standard Actions
==========

JSP -> Java Server pages

It is given by SUNMS to give a technology which would attract front end developers and main theme of jsp is to avoid java code as much as possible.

index.jsp

```
<%@ page language="java" %>
<%!
	public int squareIt(int x){
		return x*x;
	}
%>
<h1>
	The Square of 5 is :: <%= squareIt(5) %><br/>
	The Square of 6 is :: <%= squareIt(6) %><br/>
</h1>
```

This approach has several disadvantages

1. There is no clear cut seperation b/w presentation logic and business logic.
2. Person who is writing this jsp should compulsorily have knowledge of java and html which may not be possible always.
3. It won't promote code ReUsability.
4. It reduces the readability of the code also.

We can resolve this problem by encapsulating total buisness logic inside java bean.

Calculator.java

```
package in.ineuron.bean;
public class Calculator {
	static {
		System.out.println("Calculator.class file is loading...");
	}

	public Calculator() {
		System.out.println("Calculator class object is created...");
	}

	public int squareIt(int x) {
		return x * x;
	}
}
```

index.jsp

```
<%@ page language="java"%>
<jsp:useBean id="calculator" class="in.ineuron.bean.Calculator"/>
<h1>
	The Square of 5 is :: <%= calculator.squareIt(5)%><br/>
	The Square of 6 is :: <%= calculator.squareIt(6)%><br/>
</h1>
```

Advantages of this approach is

1.Seperation of Buisness logic and presentation logic

Presentation logic is available only in jsp where as Buisness logic is available in java class so readability of the code is

improved.
2. Seperation of responsibilities
Java developers can focus on buisness logic where as HTML page designer can concentrate on presentation logic.
As a result of which both can work simultaneously and we can reduce the project development time.
3. It Promotes ReUsability of the code
where ever squareIt() functionality is required we can resuse it in jsp pages.

JspStandardactions
==========
The main purpose of standard actions is to avoid java code as much as possible inside jsp.
Instead of java code ,we use tags and those tags internally will do some java logic and make front end developer work less.
1. <jsp:useBean id =' ' class=' ' scope=' '><jsp:useBean>
This tag is used to create an object/locate an object for a particular class.
2. <jsp:setProperty name = ' ' property = ' ' value=''/>
This tag is used to set the values for the properties of the bean.
3. <jsp:getProperty name=' ' property=' '/>
This tag is used to get the values from the properites of the bean.

To set the values to the jsp page
http://localhost:9999/JspStandardAction-02/set_value.jsp?sid=18&sname=kohli&saddress=RCB&sage=35

To get the values from the jsp page
http://localhost:9999/JspStandardAction-02/get_values.jsp

refer: JspStandardAction-02, JspStandardAction-03

2. Reusable components
1. <jsp:include> ====> refer: JspStandardAction-04

Different ways of performing include in jsp
==========
1. <@include file='header.jsp'>
2. <jsp:include file='header.jsp'>
3. <%
RequestDispatcher rd =reqeust.getRequestDispatcher('./header.jsp');
rd.include(request,response);
%>
4. <%
pageContext.include('header.jsp');
%>

2. <jsp:forward> ======> JspStandardAction-05

Different ways of performing forward in jsp
==========
1. <jsp:forward file='header.jsp'>
2. <%
RequestDispatcher rd =reqeust.getRequestDispatcher('./header.jsp');
rd.forward(request,response);
%>
3. <%

```
            pageContext.forward('header.jsp');
      %>


3. <jsp:param> =====> refer: JspStandardAction-06
      1. It should be used as subtag for <jsp:include> or <jsp:forward> tags
      2. It is very useful to send the data as the addition request params values
from source jsp page to destination
            web components.
      3. syntax:
                  <jsp:param name='' value=''/>
```