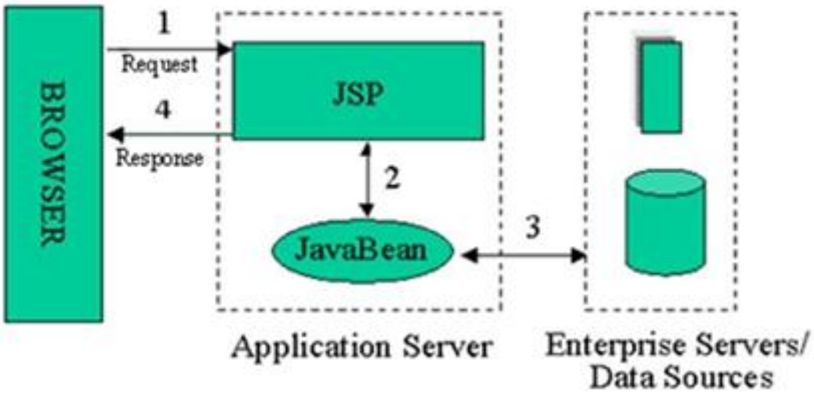
Prac 7 : Create an employee information form by using JSP

JSP is an extension to Servlets and is used to create dynamic web pages. JSP document consists of HTML and JSP tags. HTML is used to create static web pages and JSP tags are used to create dynamic web pages. JSP pages are translated into Java Servlets by translator. The Java servlet is the compiled and executed to generate an output for the client(browser).

JSP Architecture:

- The web browser directly acces JSP page. The JSP page interacts with web containers JavaBean, which represents the application model then response is sent back to the browser.
- If the generated response requires database access, JSP uses JavaBean, which gets required data from the database.



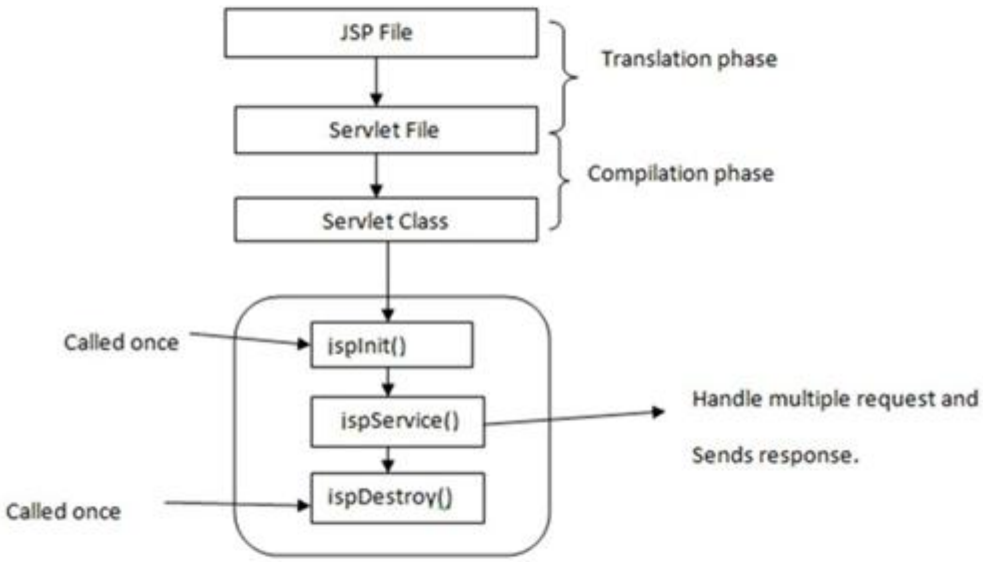
JSP – Lifecycle

A JSP life cycle is defined as the process from its creation till the destruction. This is similar to a servlet life cycle with an additional step which is required to compile a JSP into servlet.

The JSP pages follow these phases:

- Translation of JSP Page o Compilation of JSP Page
- Classloading (the classloader loads class file)
- Instantiation (Object of the Generated Servlet is created).
- Initialization (the container invokes jspInit() method).
- Request processing (the container invokes _jspService() method).
- Destroy (the container invokes jspDestroy() method).

JSP Life Cycle



1. **Translation of JSP Page:**
   In this, the Web container translates JSP document into an equivalent Java code, i.e., a Servlet. Servlet contains Java code with some mark-up language tags such as HTML/XML. This is the first step of the JSP life cycle. This translation phase deals with the Syntactic correctness of JSP. Ex: Test.jsp file is translated into Tesp.java (servlet file)

2. **Compilation of JSP Page:**
   In this, the JSP container compiles the Java code for the corresponding servlet and converts into Java byte (class) code. After the compilation stage, the servlet is ready to be loaded and initialized.

3. **Class loading** :
   Servlet class which has been loaded (created) from the JSP source is now loaded into the container. **Instantiation**:
   Here an instance of the class is generated. The container manages one or more instances by providing responses to requests.

4. **Initialization:**
   jspInit() method is called only once during the life cycle immediately after the generation of Servlet instance from JSP.

5. **Request processing**:
   jspService() method is used to serve the raised requests by JSP. It takes request and response objects as parameters. This method cannot be overridden**.**

6. **JSP Clean-up:**
   In order to remove the JSP from the use by the container or to destroy the method for servlets jspDestroy()method is used. This method is called once, if you need to perform any cleanup task like closing open files, releasing database connections jspDestroy() can be overridden.

JSP elements :

a. JSP Scripting elements(tags)
   - JSP Scriptlet tag
   - JSP expression tag
   - JSP declaration tag

b. JSP Directives
   - JSP page directive
   - JSP include directive
   - JSP taglib directive

c. Implicit objects
   - out
   - request
   - response
   - config
   - application
   - session
   - pageContext
   - page
   - exception

d. JSP Action elements

   - jsp:forward
   - jsp:include
   - java bean class
   - jsp:useBean
   - jsp:setProperty and jsp:getProperty

e. JSP Comments

<% -- JSP Comments %>

(Write your own program’s directory Structure here)