

ServletConfig, ServletContext, Request/Response Objects

1. ServletConfig – Dedicated object for a particular servlet
2. ServletContext – Shared object throughout your application
3. Request/Response – Dedicated object for a particular request

Variable Scope & Category- Parameters vs Attributes

	ServletContext Attribute/Parameter	Request Attribute	Session Attribute	ServletConfig Parameter	JSP PageContext
Scope	Application (Global)	Request	Session	Servlet	JSP Page (pageScope)

Web Forms under Servlet

Basic Form Elements	Description	Possible values
name	Specifies the name of a form	Anything
method	Specifies the HTTP method (request method) to use when sending form-data.	GET, POST
action	Specifies where to send the form-data when a form is submitted.	Servlet URL
enctype	Specifies how the form-data should be encoded when submitting it to the server	application/x-www-form-urlencoded (default), multipart/form-data

Forwarding, Redirection and Including Request to another Resource

	Forward	Redirect	Include
What it is ?	Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.	Redirects a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.	Includes the content of a resource (servlet, JSP page, or HTML file) in the response.
APIs	HttpServletRequest-> getRequestDispatcher()->forward()	HttpServletResponse->sendRedirect()	HttpServletRequest-> getRequestDispatcher()->include()
Differences	<ol style="list-style-type: none">1. The request will be further processed on the server side2. The client isn't impacted by forward, URL in a browser stays the same3. Request and response objects will remain the same object after forwarding. Request-scope objects will be still available4. Supports POST & GET both	<ol style="list-style-type: none">1. The request is redirected to a different resource2. The client will see the URL change after the redirect3. A new request is created4. Supports only GET	NA
Usage	If the previous scope is required, or the user doesn't need to be informed, but the application also wants to perform an internal action then use forwarding.	To discard the scope or if the new content isn't associated with the original request – such as a redirect to a login page or completing a form submission – then use redirecting.	To include the content of a resource in response.

Problem with Servlets – Why JSP ?

- ✓ **Processing the request and generating the response** are both **handled by a single servlet class**.
- ✓ The processing **code and the HTML elements are lumped together**.
- ✓ **Changing the look and feel** of the application requires the servlet code to be updated and **recompiled**.
- ✓ It's hard to take advantage of web page development tools when **designing the application interface**. If such tools are used to develop the web page layout, the generated HTML must then be manually **embedded into the servlet code**, a process which is time consuming, error prone, and extremely boring.

Why JSP ?

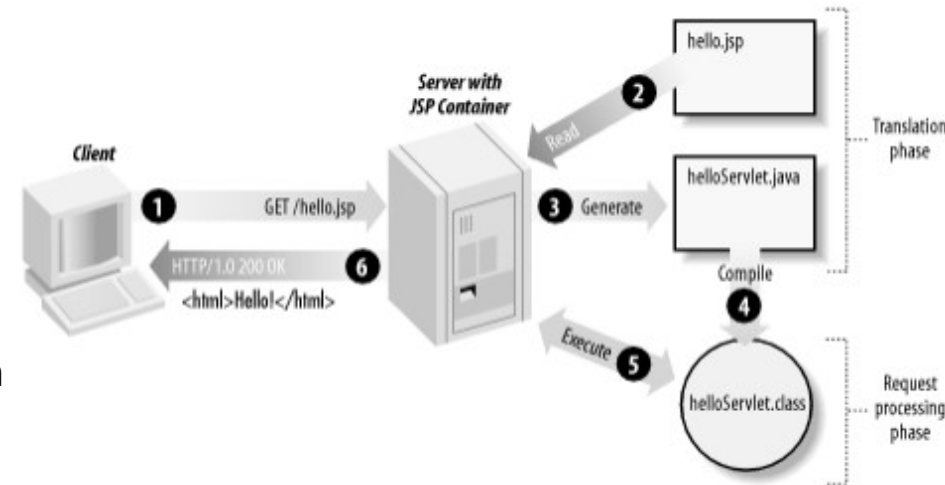
Adding JSP to the puzzle lets you solve these problems by **separating the request processing and business logic code from the presentation**. Instead of embedding HTML in the code, place all static HTML in a JSP page, just as in a regular web page, and add a few **JSP elements to generate the dynamic parts of the page**. The request processing can remain the domain of the servlet, and the business logic can be handled by JavaBeans.

Advantages -

- ✓ **Separating the request processing and business logic from presentation** makes it possible to divide the development tasks among people with different skills.
- ✓ It possible to **change different aspects of the application independently**

How JSP works?

1. Translation of JSP Page
2. Compilation of JSP Page
3. Classloading (the classloader loads class file)
4. Instantiation (Object of the Generated Servlet is created).
5. Initialization (the container invokes `jspInit()` method).
6. Request processing (the container invokes `_jspService()` m
7. Destroy (the container invokes `jspDestroy()` method).



A JSP page is handled exactly like a regular servlet; it's loaded once and called repeatedly, until the server is shut down. A JSP page inherits all the advantages of a servlet i.e. platform and vendor independence, integration, efficiency, scalability, robustness, and security.