The **JSP API (JavaServer Pages API)** is a part of the **Java EE (Jakarta EE)** platform that allows developers to create **dynamic web content** using HTML, XML, or other types of documents, with embedded Java code.

**🔧 Key Components of the JSP API**

1. **javax.servlet.jsp package** (older versions)  
   In modern versions, part of jakarta.servlet.jsp.
2. **Core Interfaces and Classes:**

| **Interface/Class** | **Description** |
| --- | --- |
| JspPage | Base interface for all JSP pages. |
| HttpJspPage | Extension of JspPage for HTTP-based JSPs. |
| JspFactory | Factory class to get JSP engine components (like PageContext). |
| PageContext | Encapsulates context information about a JSP page (request, response, session, etc.). |
| JspWriter | Used to send output to the client (like out in JSP). |
| JspException / SkipPageException | Exception classes specific to JSPs. |
| Tag, TagSupport | Used for implementing custom tag libraries (JSP tag extensions). |
| BodyTag, IterationTag | Interfaces for advanced tag behavior (e.g., looping, body content). |

**⚙️ JSP Lifecycle (Servlet Behind-the-Scenes)**

1. **Translation** – JSP is converted to a Java Servlet.
2. **Compilation** – The servlet class is compiled.
3. **Initialization** – jspInit() is called.
4. **Request handling** – service() is invoked (maps to \_jspService()).
5. **Destruction** – jspDestroy() is called on shutdown.

**💡 Common Built-in Objects (Available in JSP)**

| **Object** | **Type** | **Description** |
| --- | --- | --- |
| request | HttpServletRequest | Client request data |
| response | HttpServletResponse | Used to send response |
| session | HttpSession | Session data |
| application | ServletContext | Shared application scope |
| out | JspWriter | Writes to output stream |
| config | ServletConfig | Servlet configuration |
| pageContext | PageContext | Context for the JSP page |
| exception | Throwable | Exception caught (error pages only) |

**🏗️ Example JSP Page**

<%@ page language="java" contentType="text/html;charset=UTF-8" %>

<html>

<head><title>Hello JSP</title></head>

<body>

<% String name = request.getParameter("name"); %>

<h1>Hello, <%= name %>!</h1>

</body>

</html>

Here's a simple example that shows how a **JSP file is translated into a Java servlet** behind the scenes during the **JSP lifecycle**. This is done by the JSP engine (like Apache Tomcat).

**🌐 Example JSP File (hello.jsp)**

<%@ page language="java" contentType="text/html; charset=UTF-8" %>

<%

String name = request.getParameter("name");

if (name == null) {

name = "Guest";

}

%>

<html>

<head><title>Hello Page</title></head>

<body>

<h1>Hello, <%= name %>!</h1>

</body>

</html>

**⚙️ Translated Servlet (hello\_jsp.java)**

The container (like Tomcat) automatically translates the above JSP into a servlet. Here’s what the **translated servlet** might look like:

package org.apache.jsp;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.jsp.\*;

public final class hello\_jsp extends org.apache.jasper.runtime.HttpJspBase

implements org.apache.jasper.runtime.JspSourceDependent {

public void \_jspService(HttpServletRequest request, HttpServletResponse response)

throws java.io.IOException, ServletException {

// Setup

response.setContentType("text/html; charset=UTF-8");

PageContext pageContext = null;

HttpSession session = null;

ServletContext application = null;

ServletConfig config = null;

JspWriter out = null;

Object page = this;

try {

pageContext = JspFactory.getDefaultFactory().getPageContext(

this, request, response,

null, true, 8192, true);

application = pageContext.getServletContext();

config = pageContext.getServletConfig();

session = request.getSession();

out = pageContext.getOut();

// Translated scriptlet

String name = request.getParameter("name");

if (name == null) {

name = "Guest";

}

// Static HTML + expression output

out.write("<html>\n");

out.write("<head><title>Hello Page</title></head>\n");

out.write("<body>\n");

out.write("<h1>Hello, ");

out.write(name); // Output the name

out.write("!</h1>\n");

out.write("</body>\n");

out.write("</html>");

} catch (Throwable t) {

if (!(t instanceof SkipPageException)) {

out = pageContext.getOut();

if (out != null && out.getBufferSize() != 0)

out.clearBuffer();

if (pageContext != null) pageContext.handlePageException(t);

}

} finally {

if (pageContext != null) pageContext.release();

}

}

}

**🧠 Key Concepts**

* JSP gets converted to a servlet class extending HttpJspBase.
* HTML is rendered via out.write().
* Scriptlets (<% %>) become plain Java code inside \_jspService.
* <%= ... %> becomes inline output via out.write(...).
* JspFactory, PageContext, and JspWriter are JSP API components.

Here's a **complete, line-by-line translation** of your sample JSP into equivalent **Java servlet code** as it would be generated by a JSP engine like Tomcat.

**✅ Original JSP Code (example.jsp)**

<%@ page import="java.util.Date" %>

<%!

public String getGreeting() {

return "Welcome!";

}

%>

<%

String user = request.getParameter("user");

if (user == null) user = "Guest";

%>

<html>

<body>

<h1><%= getGreeting() %></h1>

<p>Hello, <%= user %>. Today is <%= new Date() %></p>

</body>

</html>

**🔄 Translated Servlet Code (example\_jsp.java)**

package org.apache.jsp;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.jsp.\*;

import java.util.Date; // from <%@ page import="..." %>

public final class example\_jsp extends org.apache.jasper.runtime.HttpJspBase

implements org.apache.jasper.runtime.JspSourceDependent {

// <%! ... %> → Declaration outside \_jspService()

public String getGreeting() {

return "Welcome!";

}

public void \_jspService(HttpServletRequest request, HttpServletResponse response)

throws IOException, ServletException {

// Boilerplate JSP setup

response.setContentType("text/html;charset=UTF-8");

PageContext pageContext = null;

HttpSession session = null;

ServletContext application = null;

ServletConfig config = null;

JspWriter out = null;

try {

pageContext = JspFactory.getDefaultFactory().getPageContext(

this, request, response, null, true, 8192, true);

application = pageContext.getServletContext();

config = pageContext.getServletConfig();

session = request.getSession();

out = pageContext.getOut();

// <% ... %> → Scriptlet (executed logic)

String user = request.getParameter("user");

if (user == null) user = "Guest";

// HTML and expressions <%= ... %> → out.write / out.print

out.write("<html>\n");

out.write("<body>\n");

out.write("<h1>");

out.print(getGreeting()); // <%= getGreeting() %>

out.write("</h1>\n");

out.write("<p>Hello, ");

out.print(user); // <%= user %>

out.write(". Today is ");

out.print(new Date()); // <%= new Date() %>

out.write("</p>\n");

out.write("</body>\n");

out.write("</html>\n");

} catch (Throwable t) {

if (!(t instanceof SkipPageException)) {

if (out != null && out.getBufferSize() != 0) out.clearBuffer();

if (pageContext != null) pageContext.handlePageException(t);

}

} finally {

if (pageContext != null) pageContext.release();

}

}

public java.util.List getDependants() {

return null;

}

}

**🔍 Summary Mapping**

| **JSP Construct** | **Servlet Equivalent** |
| --- | --- |
| <%@ page import="..." %> | import statement |
| <%! method or variable %> | Class-level method/variable (outside method) |
| <% Java code %> | Inside \_jspService(...) |
| <%= expression %> | out.print(expression) |
| HTML content | out.write("...") |