

---

# JSP & Servlet Life Cycle

Kevin Nilson, <http://www.javaclimber.com>

## Abstract

Interview Presentation for: Part-time - Computer and Information Science Instructor, Monday  
December 19th, 2005

## Table of Contents

Additional Resources .....	1
JSP to Servlet Source Code Generation (Translation) .....	1
.....	1
Servlet Life Cycle .....	3
Overview .....	3
Constructor .....	4
Init .....	4
Service .....	4
Destroy .....	4
Session Management .....	5
HTTPSession Creation .....	5
Review .....	5
Thank You .....	7
How was this presentation made? .....	7

## Additional Resources

- Java™ 2 Platform Enterprise Edition, v 1.4 API Specification Packages, Sun Microsystems [<http://java.sun.com/j2ee/1.4/docs/api/index.html>]
- Story of a Servlet: An Instant Tutorial, Sun Microsystems [<http://java.sun.com/products/servlet/articles/tutorial/>]
- Ant JSPC [<http://ant.apache.org/manual/OptionalTasks/jspc.html>]
- Java Servlet Programming, O'Reilly, Chapter 3: The Servlet Life Cycle [<http://www.oreilly.com/catalog/jservlet/chapter/ch03.html>]
- The J2EE 1.4 Tutorial, Sun Microsystems [<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>]

## JSP to Servlet Source Code Generation (Translation)

- Container Generates Servlet Source Code from JSP

- Source is often generated & compiled once upon first request (Tomcat)
- Source is generated & compiled at deployment
- Source is often generated (Translation) & compiled before deployment(JSPC)

```
<html>
  <head><title>Hello World</title></head>
  <body>
    <h1>Hello World!!!</h1>
    <br/>
    Today is <%=new java.util.Date()%>
  </body>
</html>
```

```
*** public abstract class HttpJspBase extends javax.servlet.http.HttpServlet implements
javax.servlet.jsp.HttpJspPage

package org.apache.jsp;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;

public final class HelloWorld_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {

    private static java.util.Vector _jspx_dependants;

    public java.util.List getDependants() {
        return _jspx_dependants;
    }

    public void _jspService(HttpServletRequest request, HttpServletResponse response
        throws java.io.IOException, ServletException) {

        JspFactory _jspxFactory = null;
        PageContext pageContext = null;
        HttpSession session = null;
        ServletContext application = null;
        ServletConfig config = null;
        JspWriter out = null;
        Object page = this;
        JspWriter _jspx_out = null;
        PageContext _jspx_page_context = null;

        try {
            _jspxFactory = JspFactory.getDefaultFactory();
            response.setContentType("text/html");
            pageContext = _jspxFactory.getPageContext(this, request, response,
                null, true, 8192, true);
            _jspx_page_context = pageContext;
            application = pageContext.getServletContext();
            config = pageContext.getServletConfig();
            session = pageContext.getSession();
            out = pageContext.getOut();
```

```
        _jspx_out = out;

        out.write("<html>\r\n");
        out.write("  <head><title>Hello World</title><head>\r\n");
        out.write("  <body>\r\n");
        out.write("    <h1>Hello World!!!</h1>\r\n");
        out.write("    <br/>\r\n");
        out.write("    Today is ");
        out.print(new java.util.Date());
        out.write("\r\n");
        out.write("  </body>\r\n");
        out.write("</html>");
    } catch (Throwable t) {
        if (!(t instanceof SkipPageException)){
            out = _jspx_out;
            if (out != null && out.getBufferSize() != 0)
                out.clearBuffer();
            if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
        } finally {
            if (_jspxFactory != null) _jspxFactory.releasePageContext(_jspx_page_context);
        }
    }
}
```

# Servlet Life Cycle

## Overview

- constructor
- init
- service
- destroy

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class HelloWorldServlet extends HttpServlet {
    public HelloWorldServlet() {
    }

    public void init() throws ServletException {
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // HttpSession session = request.getSession();
        response.setContentType("text/html");
    }
}
```

```
        PrintWriter out = response.getWriter();

        out.println("<html><head><title>HelloWorld</title></head>");
        out.println("<body>HelloWorld!!!</body>");

        out.close();
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doGet(request, response);
    }

    public void destroy() {
    }
}
```

## Constructor

- Typically not used
- Container may call on startup or with first request
- One Servlet instance or a pool handles all requests

## Init

- Called right before first request is processed
- Used for one time initializations:
  - Reading & loading configuration
  - ???
- `jspInit()` in JSP

## Service

- Called with every request
- service handles request and delegates to other methods: `doPost`, `doGet`, ...???
  - `doPost` handles POST request
  - `doGet` handles GET request
- `_jspService(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse response)` in JSP, body of JSP

## Destroy

- Called once when finished using Servlet
- Used to clean up resources
- `jspDestroy()` in JSP

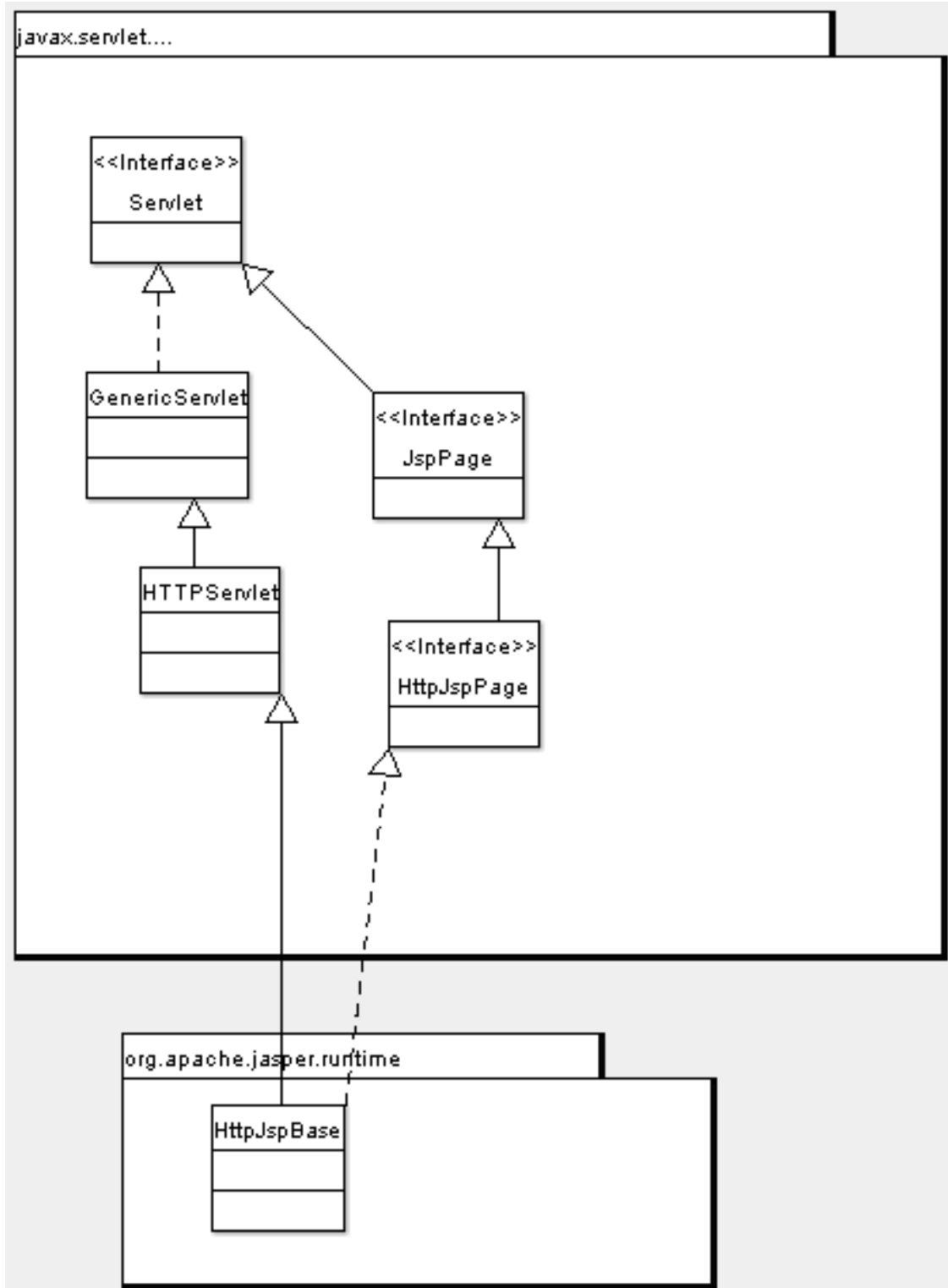
## Session Management

### HTTPSession Creation

- JSP
  - On request it will create an instance of `HTTPSession` for the user if one does not exist
  - If you do not want an instance of `HTTPSession` to be created use: `<%@ page session="false" %>`
- Servlet
  - On request it will not create an instance of `HTTPSession`
  - `HTTPSession` can be created with `request.getSession()` or `request.getSession(true)`

## Review

**Figure 1. JSP & Servlet UML1**



### Essay Questions

- What is the purpose of init and destroy?
- When is a JSP converted to a Servlet?
- When are HttpSession Instances created?
- When are the steps in the Servlet Life Cycle?

## Thank You

Any Questions?

Feel Free to contact me anytime at <http://www.javaclimber.com> [<http://www.javaclimber.com>]

## How was this presentation made?

DocBook

According to the official DocBook web site,

DocBook is a general purpose XML and SGML document type particularly well suited to books and papers about computer hardware and software (though it is by no means limited to these applications).

—DocBook.org