ICON OF JAVA

ADV. JAVA

JSP

3. JSP LIFECYCLE

Mr. Nagoor Babu M.Tech

[ICON of JAVA]
(Sun certified & Realtime Expert)

Ex. HCL Employee

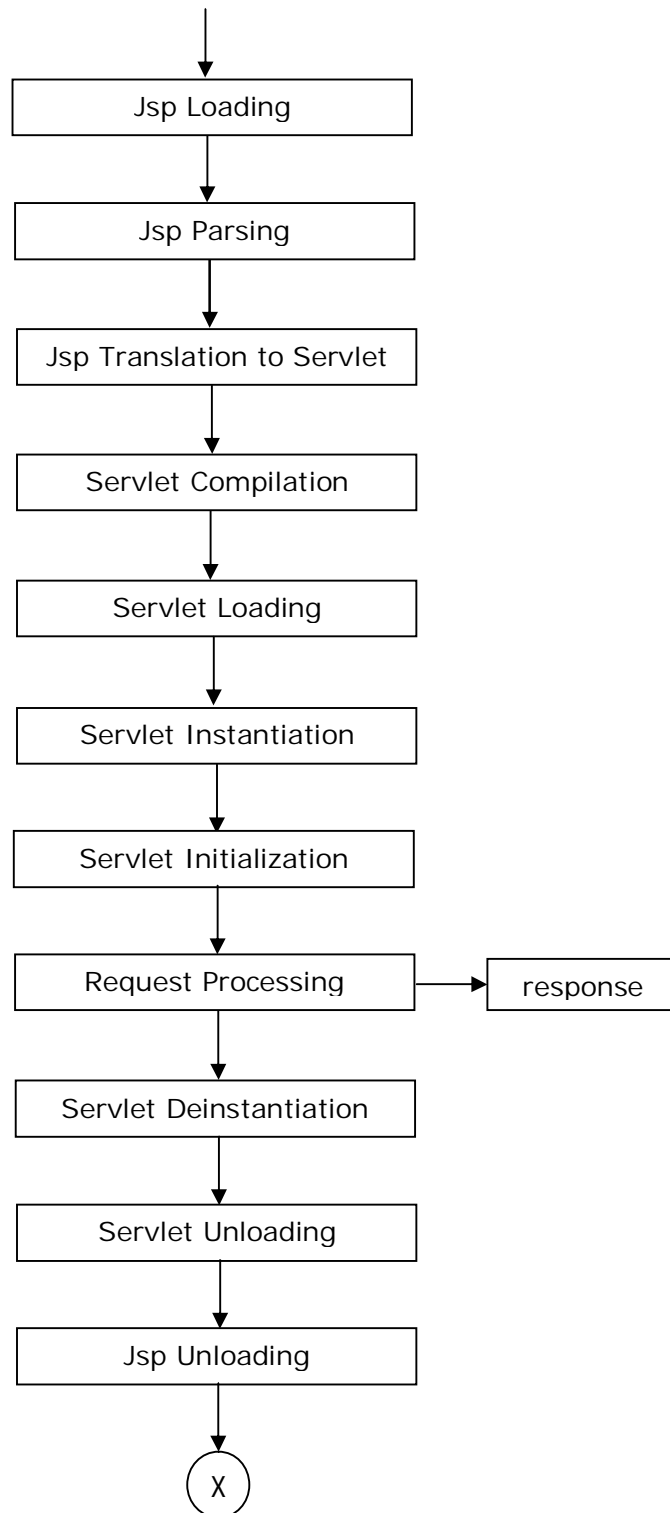Trained Lakhs of Students
for last 14 years across INDIA

India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

JSP LifeCycle

request

```
          │
          ▼
   ┌─────────────────────┐
   │    Jsp Loading      │
   └─────────────────────┘
          │
          ▼
   ┌─────────────────────┐
   │    Jsp Parsing      │
   └─────────────────────┘
          │
          ▼
   ┌─────────────────────┐
   │ Jsp Translation to Servlet │
   └─────────────────────┘
          │
          ▼
   ┌─────────────────────┐
   │  Servlet Compilation │
   └─────────────────────┘
          │
          ▼
   ┌─────────────────────┐
   │   Servlet Loading    │
   └─────────────────────┘
          │
          ▼
   ┌─────────────────────┐
   │ Servlet Instantiation │
   └─────────────────────┘
          │
          ▼
   ┌─────────────────────┐
   │ Servlet Initialization │
   └─────────────────────┘
          │
          ▼
   ┌─────────────────────┐           ┌──────────────┐
   │  Request Processing  │──────────▶│   response   │
   └─────────────────────┘           └──────────────┘
          │
          ▼
   ┌─────────────────────┐
   │ Servlet Deinstantiation │
   └─────────────────────┘
          │
          ▼
   ┌─────────────────────┐
   │  Servlet Unloading   │
   └─────────────────────┘
          │
          ▼
   ┌─────────────────────┐
   │    Jsp Unloading     │
   └─────────────────────┘
          │
          ▼
         ( X )
```

When we send request from client to server for a particular Jsp page then container will pick up the request, identify the requested Jsp pages and perform the following life cycle actions.

1. Jsp Loading:

Here container will load Jsp file to the memory from web application directory structure.

2. Jsp Parsing:

Here container will check whether all the tags available in Jsp page are in well-formed format or not.
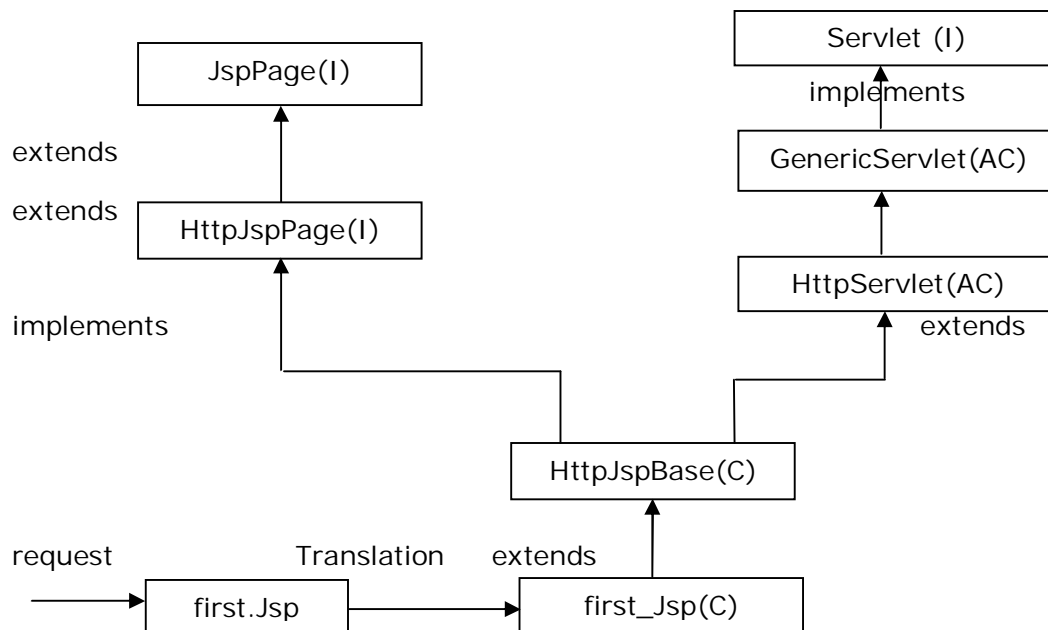
3. Jsp Translation to Servlet:

After the Jsp parsing container will translate the loaded Jsp page into a particular servlet.

While executing a Jsp page Tomcat container will provide the translated servlet in the following location at Tomcat Server.

C;\Tomcat7.0\work\catalina\localhost\org\apache\Jsp\first_Jsp.java

If the Jsp file name is first.jsp then Tomcat Server will provide a servlet with name first_jsp. By default all the translated servlets provided by Tomcat container are final.

The default super class for translated servlet is **HttpJspBase**.

```
                JspPage(I)                              Servlet (I)
                                                         ↑
         extends                                         implements
                ↑                               GenericServlet(AC)
         extends   HttpJspPage(I)                        ↑
                ↑                                  HttpServlet(AC)
                                                         ↑
      implements                                   extends
                  └──────────────┐          ┌──────┘
                              HttpJspBase(C)
                                   ↑
   request        Translation   extends
      →   first.Jsp  ──────────→   first_Jsp(C)
```

Where **JspPage** interface has declared the following methods.

public void _JspInit()

public void _JspDestroy()

Where **HttpJspPage** interface has provided the following method.

public void _JspService(HttpServletRequest req, HttpServletResponse res)

For the above 3 abstract methods HttpJspBase class has provided the default implementation but _JspService(_,_) method would be overridden in first_jsp class with the content what we provided in first.jsp file.

4. Servlet Compilation:

After getting the translated servlet container will compile servlet java file and generates the respective .class file.

5. Servlet Loading:

Here container will load the translated servlet class byte code to the memory.

6. Servlet Instantiation:

Here container will create object for the loaded servlet.

7. Servlet Initialization:

Here container will access _JspInit() method to initialize the servlet.

8. Creating request and response objects:

After the servlet initialization container will create a thread to access _JspService(_,_) method, for this container has to create HttpServletRequest and HttpServletResponse.

9. Generating Dynamic response:

After getting request and response objects container will access _JspService(_,_) method, by executing its content container will generate some response on response object.

10. Dispatching Dynamic response to Client:

When container generated thread reached to the ending point of _JspService(_,_) method then that thread will be in Dead state, with this container will dispatch dynamic response to client through the Response Format prepared by the protocol.

11. Destroying request and response objects:

When the dynamic response reached to client protocol will terminate its virtual socket connection, with this container will destroy request and response objects.

12. Servlet Deinstantiation:

After destroying request and response objects container will be in waiting state depends on the container, then container identifies no further request for the same resource then container will destroy servlet object, for this container will execute _JspDestroy() method.

13. Servlet Unloading and Jsp Unloading:

After the servlet deinstantiation container will eliminate the translated servlet byte code and Jsp code from memory.