

# JSP - Lifecycle

In this chapter, we will discuss the lifecycle of JSP. The key to understanding the low-level functionality of JSP is to understand the simple life cycle they follow.

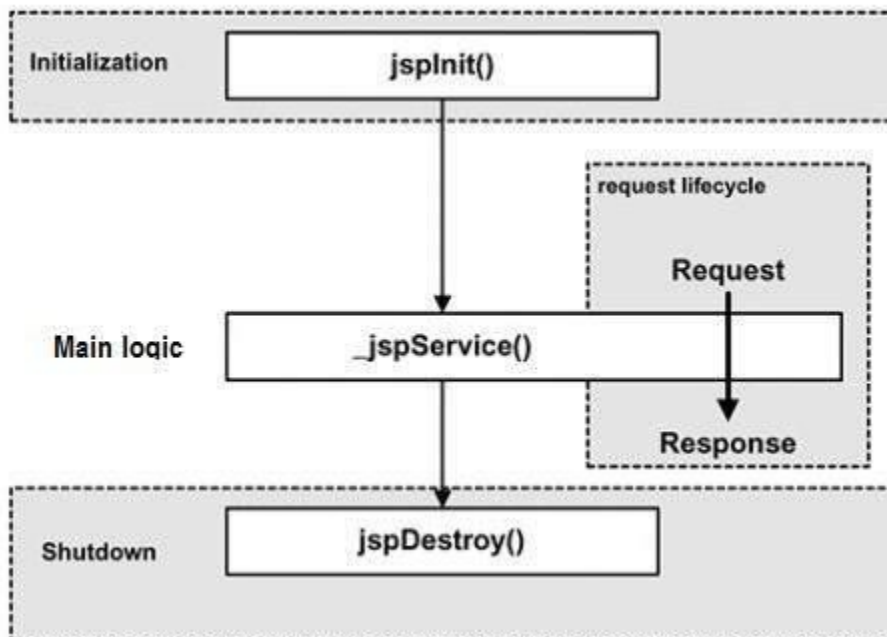
A JSP life cycle is defined as the process from its creation till the destruction. This is similar to a servlet life cycle with an additional step which is required to compile a JSP into servlet.

## Paths Followed By JSP

The following are the paths followed by a JSP –

- Compilation
- Initialization
- Execution
- Cleanup

The four major phases of a JSP life cycle are very similar to the Servlet Life Cycle. The four phases have been described below –



## JSP Compilation

When a browser asks for a JSP, the JSP engine first checks to see whether it needs to compile the page. If the page has never been compiled, or if the JSP has been modified since it was last compiled, the JSP engine compiles the page.

The compilation process involves three steps –

- Parsing the JSP.

- Turning the JSP into a servlet.
- Compiling the servlet.

## JSP Initialization

When a container loads a JSP it invokes the **jspInit()** method before servicing any requests. If you need to perform JSP-specific initialization, override the **jspInit()** method –

```
public void jspInit(){  
    // Initialization code...  
}
```

Typically, initialization is performed only once and as with the servlet init method, you generally initialize database connections, open files, and create lookup tables in the jspInit method.

## JSP Execution

This phase of the JSP life cycle represents all interactions with requests until the JSP is destroyed.

Whenever a browser requests a JSP and the page has been loaded and initialized, the JSP engine invokes the **\_jspService()** method in the JSP.

The **\_jspService()** method takes an **HttpServletRequest** and an **HttpServletResponse** as its parameters as follows –

```
void _jspService(HttpServletRequest request, HttpServletResponse response) {  
    // Service handling code...  
}
```

The **\_jspService()** method of a JSP is invoked on request basis. This is responsible for generating the response for that request and this method is also responsible for generating responses to all seven of the HTTP methods, i.e, **GET**, **POST**, **DELETE**, etc.

## JSP Cleanup

The destruction phase of the JSP life cycle represents when a JSP is being removed from use by a container.

The **jspDestroy()** method is the JSP equivalent of the destroy method for servlets. Override jspDestroy when you need to perform any cleanup, such as releasing database connections or closing open files.

The jspDestroy() method has the following form –

```
public void jspDestroy() {  
    // Your cleanup code goes here.  
}
```