HARAMAYA UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF SOFTWARE ENGINEERING

ASSIGNMENT FOR JAVA

NAME: YUSUF KEDIR
ID NUMBER: 3737/14
SECTION: B

Life Cycle of a Servlet

The entire life cycle of a Servlet is managed by the Servlet container which uses the javax.servlet.Servlet interface to understand the Servlet object and manage it. So, before creating a Servlet object, let's first understand the life cycle of the Servlet object which is actually understanding how the Servlet container manages the Servlet object.

Stages of the Servlet Life Cycle: The Servlet life cycle mainly goes through four stages:

1. Loading a Servlet.
2. Initializing the Servlet.
3. Request handling.
4. Destroying the Servlet.

1. **Loading a Servlet**: The first stage of the Servlet lifecycle involves loading and initializing the Servlet by the Servlet container. The Web container or Servlet Container can load the Servlet at either of the following two stages:

Initializing the context, on configuring the Servlet with a zero or positive integer value.

If the Servlet is not preceding stage, it may delay the loading process until the Web container determines that this Servlet is needed to service a request.

The Servlet container performs two operations in this stage:

Loading: Loads the Servlet class.

Instantiation: Creates an instance of the Servlet. To create a new instance of the Servlet, the container uses the no-argument constructor.

2. **Initializing a Servlet**: After the Servlet is instantiated successfully, the Servlet container initializes the instantiated Servlet object. The container initializes the Servlet object by invoking the **Servlet.init(ServletConfig)** method which accepts ServletConfig object reference as parameter.

The Servlet container invokes the **Servlet.init(ServletConfig)** method only once, immediately after the **Servlet.init(ServletConfig)** object is instantiated successfully. This method is used to initialize the resources, such as JDBC data source.

Now, if the Servlet fails to initialize, then it informs the Servlet container by throwing the **ServletException** or **UnavailableException**.

3. **Handling request**: After initialization, the Servlet instance is ready to serve the client requests. The Servlet container performs the following operations when the Servlet instance is located to service a request :

It creates the **ServletRequest** and **ServletResponse** objects. In this case, if this is a HTTP request, then the Web container creates **HttpServletRequest** and **HttpServletResponse** objects which are subtypes of the **ServletRequest** and **ServletResponse** objects respectively.

After creating the request and response objects it invokes the Servlet.service(ServletRequest, ServletResponse) method by passing the request and response objects.

The **service()** method while processing the request may throw the **ServletException** or **UnavailableException** or **IOException**.

4. **Destroying a Servlet**: When a Servlet container decides to destroy the Servlet, it performs the following operations,

It allows all the threads currently running in the service method of the Servlet instance to complete their jobs and get released.

After currently running threads have completed their jobs, the Servlet container calls the **destroy()** method on the Servlet instance.

After the **destroy()** method is executed, the Servlet container releases all the references of this Servlet instance so that it becomes eligible for garbage collection.

Servlet Life Cycle Methods

There are three life cycle methods of a Servlet :

- init()
- service()
- destroy()

Let's look at each of these methods in details:

1. init() method: The Servlet.init() method is called by the Servlet container to indicate that this Servlet instance is instantiated successfully and is about to put into service.

```
//init() method

public class MyServlet implements Servlet{
```

```
public void init(ServletConfig config) throws ServletException {
//initialization code }
//rest of code
}
```

2. **service() method**: The **service()** method of the Servlet is invoked to inform the Servlet about the client requests.

This method uses **ServletRequest** object to collect the data requested by the client.

This method uses **ServletResponse** object to generate the output content.

```
// service() method

public class MyServlet implements Servlet{
  public void service(ServletRequest res, ServletResponse res)
  throws ServletException, IOException {
      // request handling code
  }
  // rest of code
}
```

3. destroy() method: The destroy() method runs only once during the lifetime of a Servlet and signals the end of the Servlet instance.

```
//destroy() method
public void destroy()
```

As soon as the destroy() method is activated, the Servlet container releases the Servlet instance.

The following is an example of java servlets program that accepts two numbers from the user and displays their addition, our project name in which the files are saved is “CalServlet”. the name of the java program is “AddServlet.java”, and the name of html program is “Cal.html”.

The below is AddServlet.java

```
import jakarta.servlet.ServletException;
```

```
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/AddServlet")
public class AddServlet extends HttpServlet {
private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
int number1 = Integer.parseInt(request.getParameter("num1"));
int number2 = Integer.parseInt(request.getParameter("num2"));
int sum = number1 + number2;

response.setContentType("text/html");
PrintWriter pw = response.getWriter();
pw.println("The sum of the two numbers is: " + sum);
}
}
```

The below is Cal.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script type="text/javascript">
function Add() {
var num1 = document.getElementById("num1").value;
var num2 = document.getElementById("num2").value;

// Make an AJAX request to the servlet
var xhr = new XMLHttpRequest();
xhr.open("GET", "AddServlet?num1=" + num1 + "&num2=" + num2, true);
xhr.onreadystatechange = function() {
if (xhr.readyState === 4 && xhr.status === 200) {
// Update the page content with the response
document.getElementById("result").innerHTML = xhr.responseText;
}
};
xhr.send();
}
</script>
</head>
<body>
<form action="AddServlet" method="get" align="center">
Enter first number: <input type="text" name="num1" id="num1" align="center">
Enter second number: <input type="text" name="num2" id="num2" align="center">
<input type="button" value="Add" align="center" onclick="Add()">
```

```
</form>
<div id="result"></div>
</body>
</html>
```