**Servlet Life Cycle**

the life cycle of the **Servlet** with an example. There is a **[Servlet interface](https://www.javaguides.net/2019/02/servlet-interface-example.html" \t "_blank)** in Servlet API which is the central abstraction of the Java Servlet API. All servlets implement this interface either directly, or more commonly, by extending a class that implements the interface. The **[Servlet interface](https://www.javaguides.net/2019/02/servlet-interface-example.html" \t "_blank)** provides a life cycle methods to initialize a servlet, to service requests, and to remove a servlet from the server.

These **[Servlet interface](https://www.javaguides.net/2019/02/servlet-interface-example.html" \t "_blank)** methods are central to the life cycle of a servlet.

1. **init( )**
2. **service()**
3. **destroy( )**

Let us consider a typical user scenario to understand when these methods are called.

1. First, assume that a user enters a *Uniform Resource Locator (URL)* to a web browser. The browser then generates an HTTP request for this URL. This request is then sent to the appropriate server.
2. Second, this HTTP request is received by the web server. The server maps this request to a particular servlet. The servlet is dynamically retrieved and loaded into the address space of the server.
3. Third, the server invokes the *init( )* method of the servlet. This method is invoked only when the servlet is first loaded into memory. It is possible to pass initialization parameters to the servlet so it may configure itself.
4. Fourth, the server invokes the *service( )* method of the servlet. This method is called to process the HTTP request. You will see that it is possible for the servlet to read data that has been provided in the HTTP request. It may also formulate an HTTP response for the client.
5. The servlet remains in the server’s address space and is available to process any other HTTP requests received from clients. The *service( )* method is called for each HTTP request.
6. Finally, The servlet is taken out of service, then destroyed with the *destroy()* method, then garbage collected and finalized.

Below diagram shows a life cycle methods of Servlet interface:

**[](https://2.bp.blogspot.com/-QNP-eBNyBc4/XG4QAY6IbDI/AAAAAAAAFpA/Er5XD2SFYRotwH2h3z5nLka0WJP12dBLwCLcBGAs/s1600/servlet-interface-coding-example.PNG)**

**Servlet Life Cycle Example**

In this example, we create ServletImpl which implements Servlet interface and it's all methods. Let's use @WebServlet annotation to define Servlet component so that web.xml is not required.

package net.javaguides.servlet.tutorial;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.Servlet;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.annotation.WebServlet;

@WebServlet("/ServletInterface")

public class ServletImpl implements Servlet {

private ServletConfig config = null;

@Override

public void init(ServletConfig config) throws ServletException {

this.config = config;

System.out.println("iniside init method : " + config.getServletName());

}

@Override

public ServletConfig getServletConfig() {

return this.config;

}

@Override

public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException {

System.out.println("inside service method");

res.setContentType("text/html");

PrintWriter out = res.getWriter();

out.print("<html><body>");

out.print("<b> Servlet Interface Example</b>");

out.print("</body></html>");

}

@Override

public String getServletInfo() {

return "Demonstrated implementaton of Servlet interface";

}

@Override

public void destroy() {

System.out.println("inside destroy method");

}

}

**Output:**

Let's deploy above servlet in a web container like tomcat 8 and hit [**http://localhost:8080/java-servlet-tutorial/ServletInterface**](http://localhost:8080/java-servlet-tutorial/ServletInterface) link in a browser will result below web page:

**[](https://4.bp.blogspot.com/-2WRZYHkvsiw/XG4VK2GDkUI/AAAAAAAAFpM/Dmi7fDusKxw8ZOA5TBab5ZcaHi13jjdbACLcBGAs/s1600/servlet-interface-output.PNG)**