The Servlet Container manages the servlet component life cycle.The important actions that are performed with respect to servlet comp in the life cycle of servlet component are called servlet life cycle events. In the servlet component life cycle 3 life cycle events will be raised by servlet container.

1. Instantiation event: This event raises when servlet container creates our servlet component object.
2. Request Processing:- This event raises when servlet container keeps our servlet component object ready for request processing.
3. Destruction event:- This event raises when servlet container is ready to destroy our servlet component object.

The method that will be called automatically by underlying server for life cycle event that is raised is called **life cycle method**.

**1.Life Cycle Methods:-**  Servlet container calls 3 life cycle methods automatically for 3 life cycle events.

**a. int(-):-**

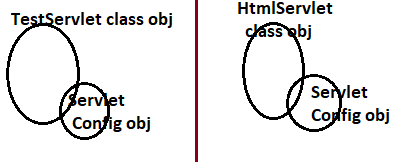


This method execution is done only once in Servlet life cycle. So it is one time execution method. we generally place initialization logics in the init(-) method like creating jdbc con objection,Initializing the received ServletConfig object to our servlet class obj.

When servlet component class object is created, servlet container invokes the init(-) method. Before invoking the init(-) method, servlet container creates the “servletConfig” object.

The ServletConfig object is 1 per servlet comp class object. Using this object we can pass inputs to servlet component and we can gather more info about servlet component.

EX:- SerletConfig object to ServletComponent class obj is like PA to Minister.



**b. service(-):-**  The Servlet container automatically calls this method for every request that is given servlet component. So we generally place request processing logics and response generation logics in this method.

**C. Destory(-):-** It is one time executing block in servlet life cycle. The servlet container automatically calls this method when servlet container is about to destroy out servlet component class object. But this method does not remove it. To remove, the servlet container will not depends on JVM provided garbage collector. The servlet container have its own garbage collector. It removes the Servlet component class object.

Generally place unintialization logics in the destroy() method. These logics are like

a.Closing jdbc objs

b. closing io streams

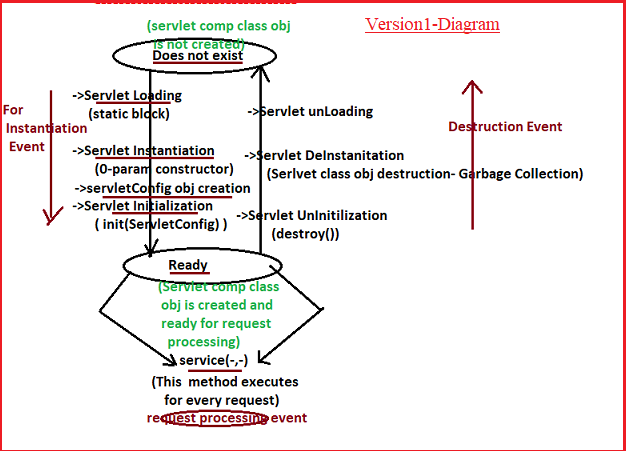
c. nullifying other important objects.

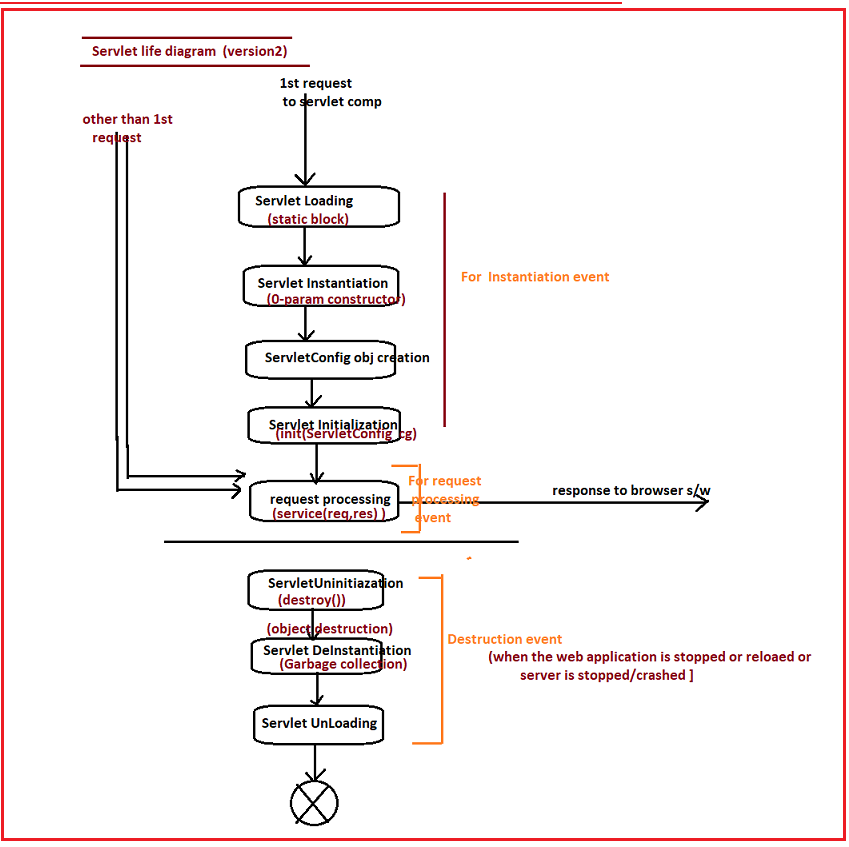
(assigning null values to reference variable).

In the following cases, ServletContainer will destroy the servlet comp class object:

1. When the web application is stopped or reloaded or undeployed.
2. When the underlying web server is stopped or restarted or crashed.
3. If created servlet component class object is idle from long time.

2. **Servlet LifeCycle Diagram:-**





Q)what happens if u call init(-) and destroy(-) method from the service(-) method?

The servlet container will executes logic of init(-) and destroy(-) method along with service(-) method. In this case, servlet container will not treat those two methods as servlet life cycle methods.

Example:- This program demonstrates servlet life cycle.

**FileName: LifeCycle**

**package** com.serv.comp;

**import** jakarta.servlet.ServletConfig;

**import** jakarta.servlet.http.HttpServlet;

**import** jakarta.servlet.http.HttpServletRequest;

**import** jakarta.servlet.http.HttpServletResponse;

**public** **class** LifeCycle **extends** HttpServlet {

**static** {

System.***out***.println("Servlet static block");

}

**public** LifeCycle(){

System.***out***.println("Servlet 0-param constructor");

}

**public** **void** init(ServletConfig sfg) {

System.***out***.println("Servlet init method");

}

**public** **void** service(HttpServletRequest req,HttpServletResponse res) {

System.***out***.println("Servlet service method");

}

@Override

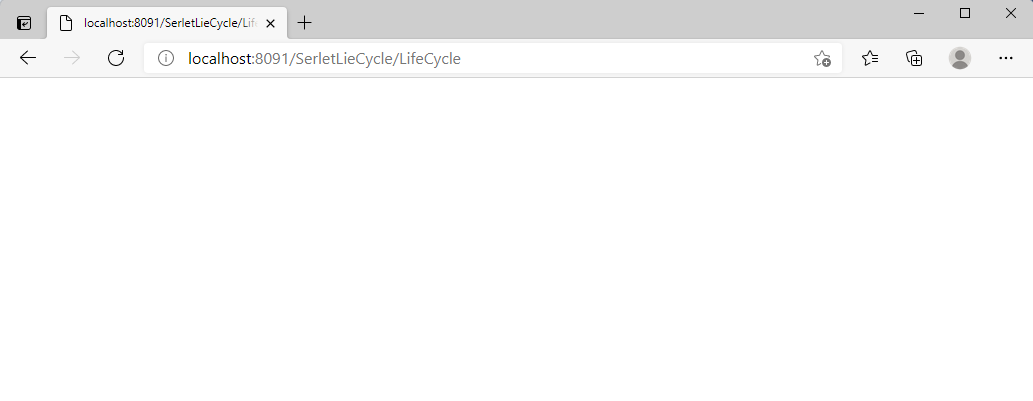
**public** **void** destroy() {

System.***out***.println("Servlet destroy method");

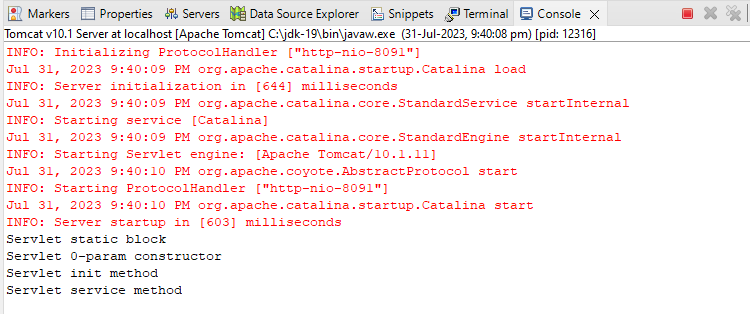
}

}

**Output:-**

****

When I gave first request for **LifeCycle servlet comp class**, we saw below outpur on server console.

****

