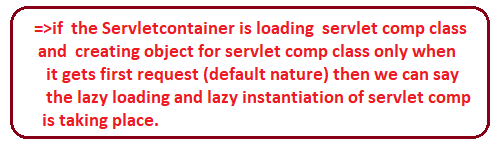
When First request is given for servlet component, The following activities will be done.

1. Servlet loading.
2. Servlet instatiation (Servlet object creation)
3. Servlet initialization(calling init(-) method)
4. Request processing and response generation(calling service,(or)doGet(-) method)

Where as other than 1st request is given for servlet component, Only the following single activity will be done.

1. Request processing and response Generation(calling service(-) or doGet(-) method.

Lazy loading:-



**The Response time( The time taken to process the request and to generate the response) of First request is bit higher compare to other than First request respons time.**

To overcome this problem, we have to enable the early/eager/pre-instantiation/loading on servlet component. It can be enabled by **<load-on-startup> tag.** We are making the servlet container to load the servlet comp class, to create the servlet comp class object, to initialize the servlet comp class object during **either server startup or deployment of web application**.

Syntax:



This tag should be written in servlet configuration in web.xml file.

if multiple servlet comps of a web application are enabled with <load-on-startup> then their order of early loading either furing server startup or during the deployment of web application will be decided based on the <load-on-startup> priority value.

* High value indicate low priority.
* Low value indicates high priority.
* -ve value ignore the enabled <loa-on-startup>

Enable <load-on-startup> on such servlet comp that will be requested guaranteely afte the deployment of web application.

Ex:- 1. Servlet comp gives homepage

2.servlet comp gives log-in page

…etc.

We should not enable <load-on-starup> on following types of servlets.

EX:- 1. Servlet which gives contact\_us page.

1. servlet which gives about\_us page.
2. Servlet which give terms and condition page.
3. …etc

Example:- This application demonstrates the early-loading.

FileName:LifeCycle.java

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");

}

}

**Web.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"https://jakarta.ee/xml/ns/jakartaee"* xsi:schemaLocation=*"https://jakarta.ee/xml/ns/jakartaee https://jakarta.ee/xml/ns/jakartaee/web-app\_6\_0.xsd"* version=*"6.0"*>

<servlet>

<description></description>

<display-name>LifeCycle</display-name>

<servlet-name>LifeCycle</servlet-name>

<servlet-class>com.serv.comp.LifeCycle</servlet-class>

**<load-on-startup>1</load-on-startup>**

</servlet>

<servlet-mapping>

<servlet-name>LifeCycle</servlet-name>

<url-pattern>/LifeCycle</url-pattern>

</servlet-mapping>

<display-name>SerletLieCycle</display-name>

<welcome-file-list>

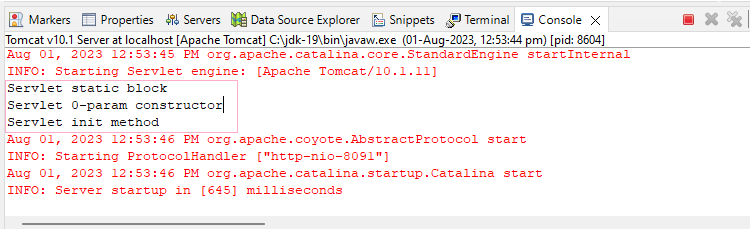
<welcome-file>index.html</welcome-file>

</welcome-file-list>

</web-app>

**Output:-**

When I started server, LifeCyle component class has been loaded,instantiated, initialized.



When I gave first request to same servlet component class, servlet container will invokes the only service(-) method.

