**Starting with first Servlet Application**

To get started with Servlets, let’s first start with a simple Servlet application *i.e* **LifeCycle** application, that will demonstrate the implementation of the **init()**, **service()** and **destroy()** methods.  
First of all it is important to understand that if we are developing any Servlet application, it will handle some client’s request so, whenever we talk about Servlets we need to develop a index.html page (can be any other name also) which will request a particular Servlet to handle the request made by the client (in this case index.html page).  
To be simple, lets first describe the steps to develop the **LifeCycle** application :

* Creating the index.html page
* Creating the **LifeCycle** Servlet
* Creating deployment descriptor

**Creating the index.html page**

For the sake of simplicity, this page will just have a button **invoke life cycle**. When you will click this button it will call **LifeCycleServlet** (which is mapped according to the entry in web.xml file).

* HTML

|  |
| --- |
| <**html**>      <**form** action="LifeCycleServlet">          <**input** type="submit" value="invoke life cycle servlet">      </**form**>  </**html**> |

The name of the Servlet is given in action attribute of form tag to which the request will be send on clicking the button, in this case **FirstServlet**.

**Creating the Servlet (FirstServlet)**

Now, its time to create the LifeCycleServlet which implements **init()**, **service()** and **destroy()** methods to demonstrate the lifecycle of a Servlet.

* Java

|  |
| --- |
| // importing the javax.servlet package  // importing java.io package for PrintWriter  **import** javax.servlet.\*;  **import** java.io.\*;    // now creating a servlet by implementing Servlet interface  **public** **class** LifeCycleServlet **implements** Servlet {        ServletConfig config = **null**;        // init method  **public** **void** init(ServletConfig sc)      {          config = sc;          System.out.println("in init");      }        // service method  **public** **void** service(ServletRequest req, ServletResponse res)  **throws** ServletException, IOException      {          res.setContenttype("text/html");          PrintWriter pw = res.getWriter();          pw.println("<h2>hello from life cycle servlet</h2>");          System.out.println("in service");      }        // destroy method  **public** **void** destroy()      {          System.out.println("in destroy");      }  **public** String getServletInfo()      {  **return** "LifeCycleServlet";      }  **public** ServletConfig getServletConfig()      {  **return** config; // getServletConfig      }  } |

**Creating deployment descriptor(web.xml)**

As discussed in other posts about **web.xml** file we will just proceed to the creation of it in this article.

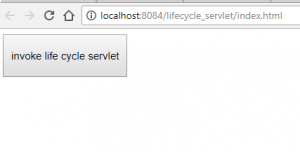
* XML

|  |
| --- |
| <?**xml** version="1.0" encoding="UTF=8"?>  <**web-app**>      <**servlet**>           <**servlet-name**>LifeCycleServlet</**servlet-name**>           <**servlet-class**>LifeCycleServlet</**servlet-class**>      </**servlet**>      <**servlet-mapping**>           <**servlet-name**>LifeCycleServlet</**servlet-name**>           <**url-pattern**>/LifeCycleServlet</**url-pattern**>      </**servlet-mapping**>      <**session-config**>           <**session-timeout**>               30           </**session-config**>  </**web-app**> |

To know the working and use of the above **web.xml** file its recommended to read the next post.

**How to run the above program ?**

It is important to make sure that you have some server like **Apache Tomcat**installed and configured with the IDE of your choice like Netbeans.  
Now, if the above condition is fulfilled then you can simply create the above three files under **Web application** project and then simply run the above application.  
First of all the **index.html** file gets executed and then when the button is clicked the request goes to the Servlet, in this case LifeCycleServlet and the service() method handles the request.



When the above **invoke life cycle servlet** button is clicked the code under service() method of LifeCycleServlet is executed and the below output is obtained :

