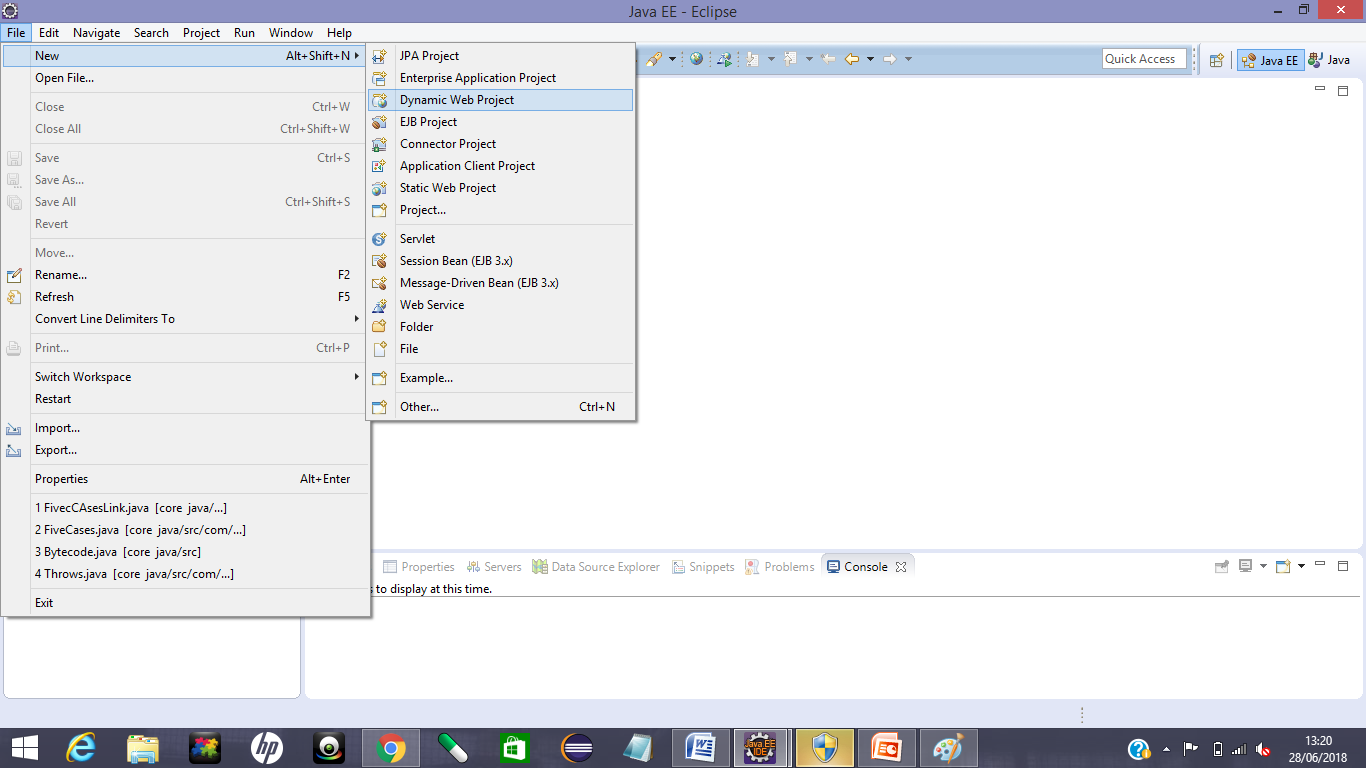
**Client-server (Request- Response) programming:-**

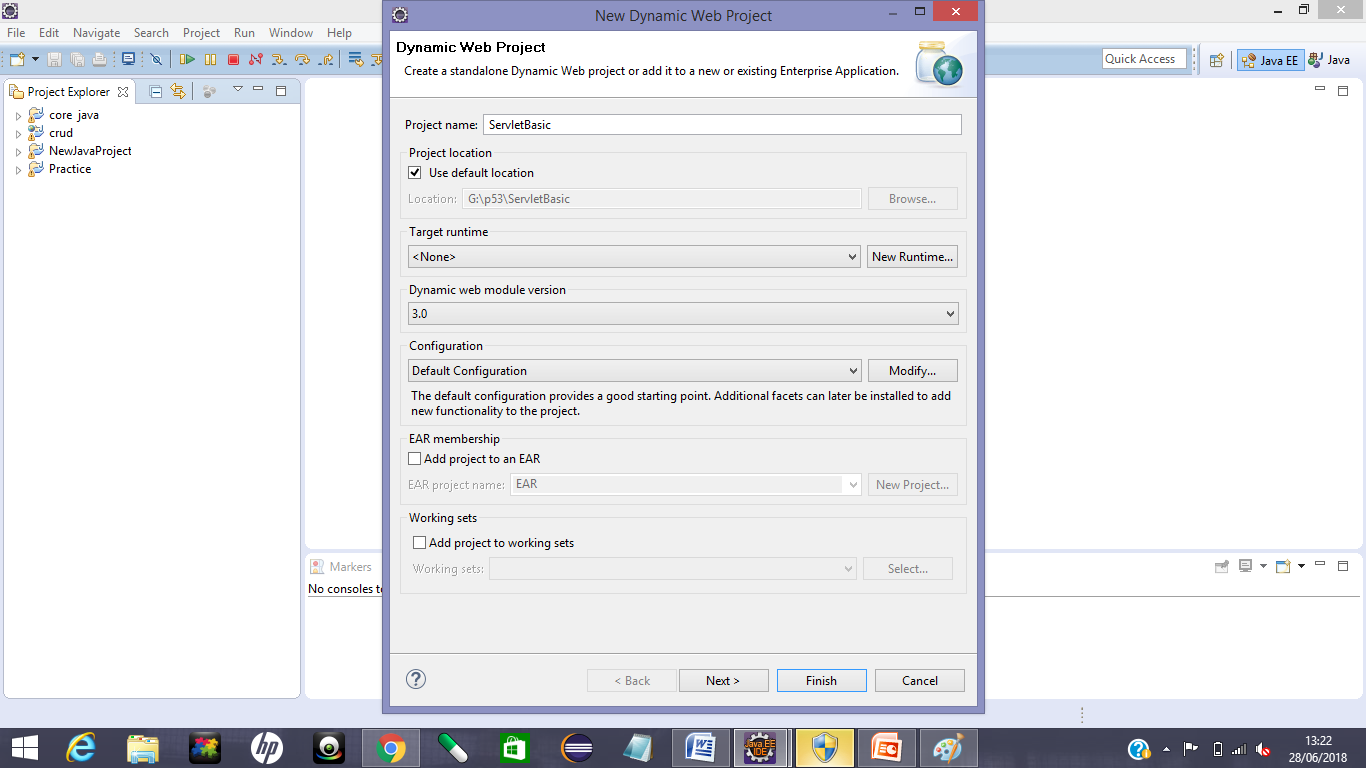
**Index -**

1. **How to Create web dynamic project**
2. **How to Create a servlet**
3. **Deploy project on server**
4. **Execute project on server**
5. **Web Dynamic project setup:**

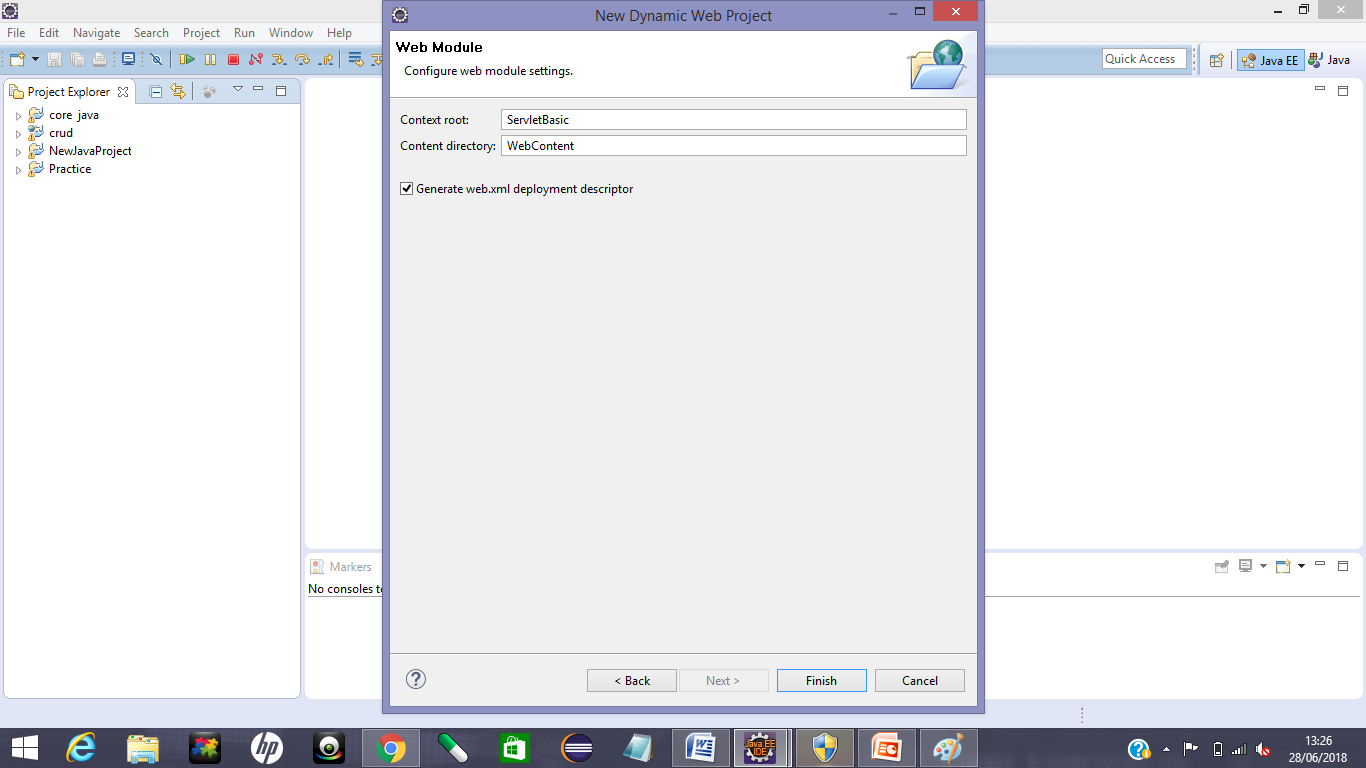
**Step 1:** click

****

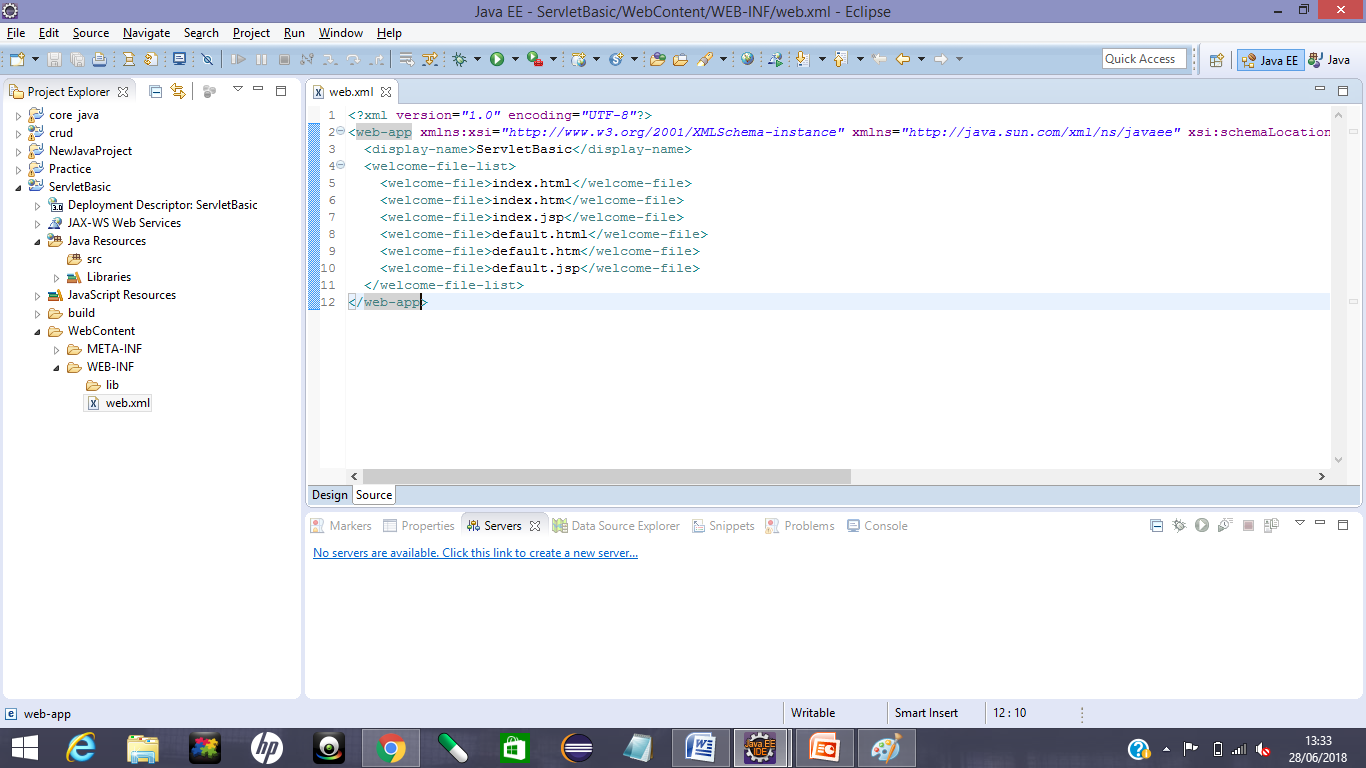
**Step 2:** Title any name to project and click on next.

****

**Step 3: click on the check box to create web.xml deployment descriptor and then finish.**

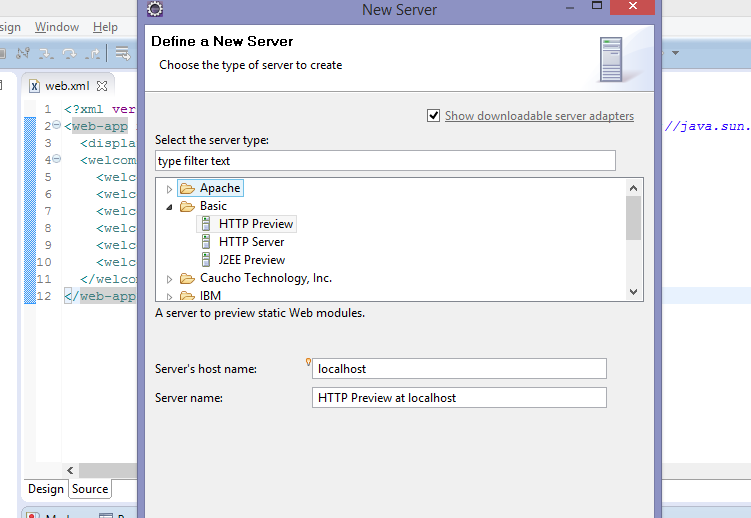
****

**Step 4:To add server**.

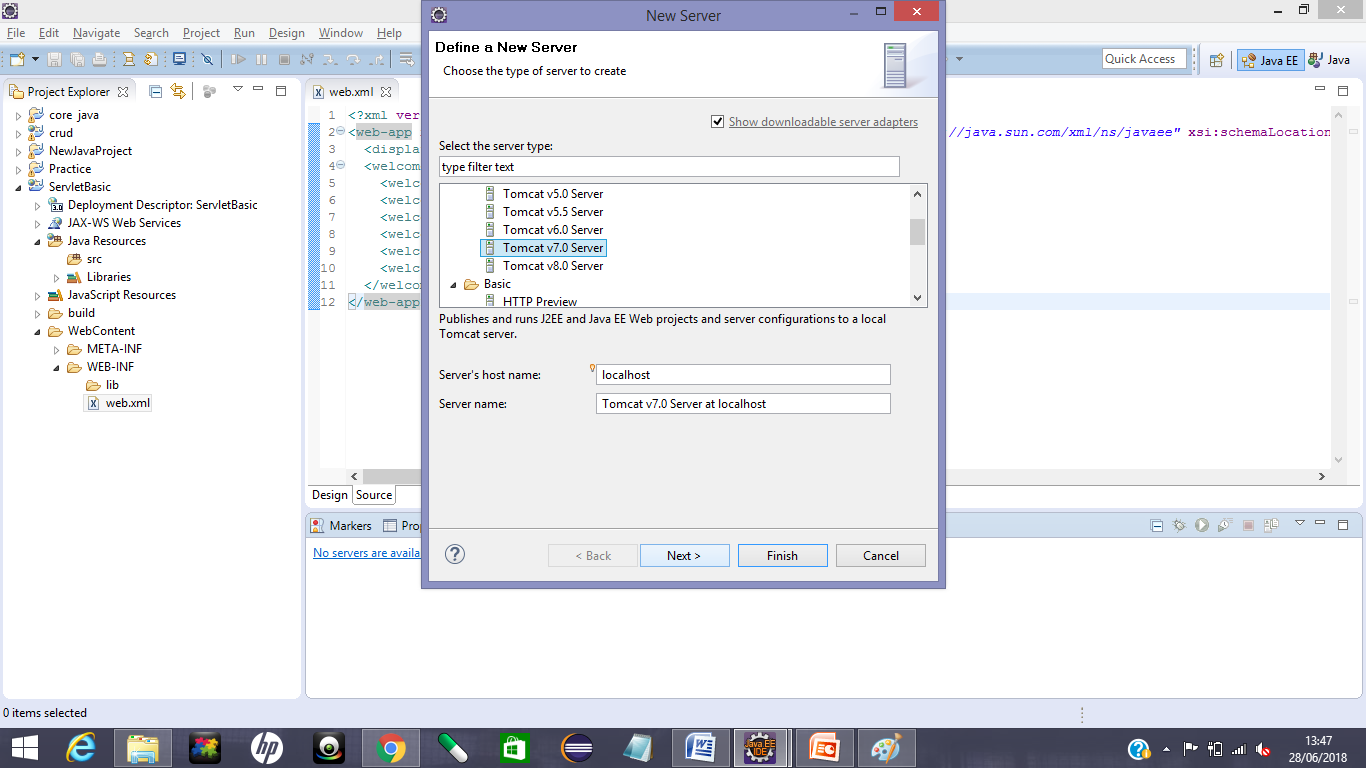
****

There is no servers are added, so we click on link to add server.

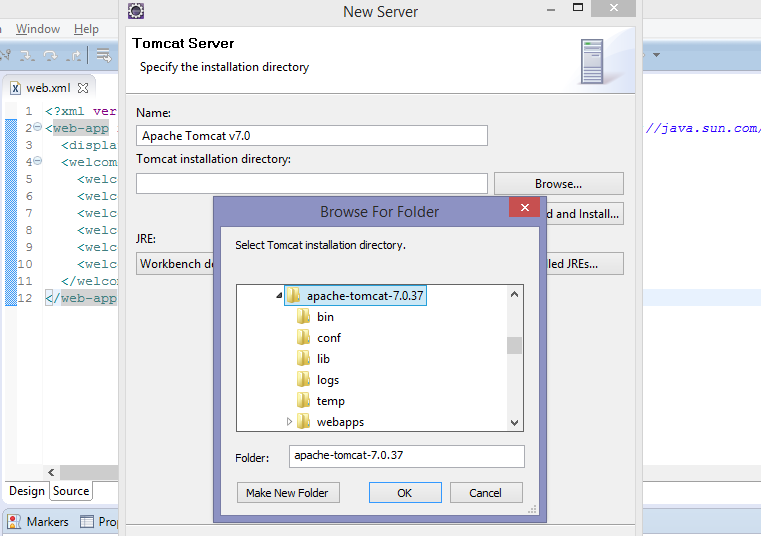
**Step 5: 5.1:** select Apache.

****

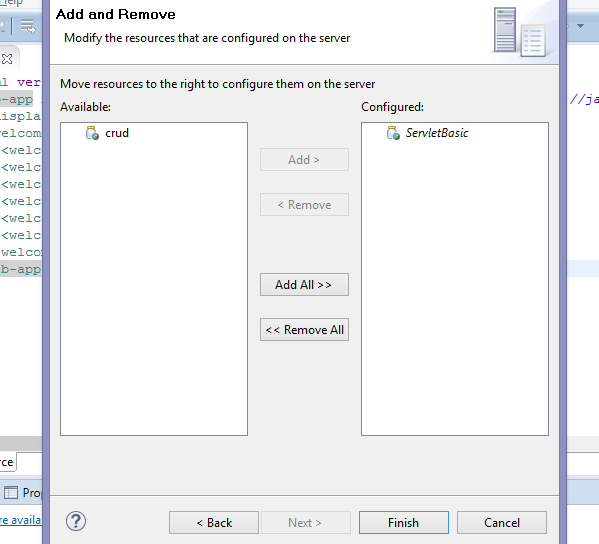
**5.2:**click on the version which you are using and click next.

****

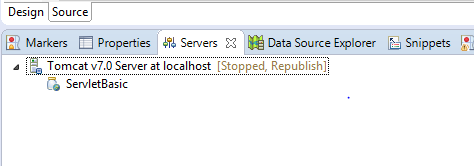
**5.3:** Browse the installation directory of tomcat and click on OK and Next.

****

**5.4:** Add your project to the server and Finish.

****

Now you can see your project is added on the server.

****

**Server:** Server serves some services and functionalities to clients or users. There are two type of server:

1. Web Server: web Server only contain web container.
2. Application Server: Application server contain web container and ejb-connector.

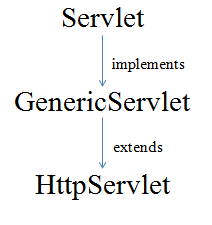
**Note\*\*\* Web container creates object of Servlet and it is responsible to manage lifecycle of Servlet.**

**What is Servlet?**

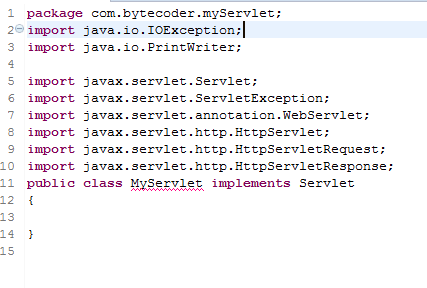
**Servlet**: Servlet is aserver side technology which is used to create web application or web dynamic pages.

There are 3 ways we can create servlet:

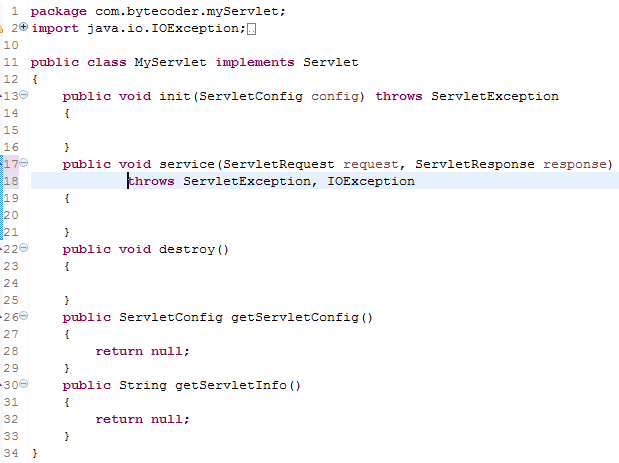
1. By implement Servlet interface
2. By extends Generic class
3. By extends HttpServlet



1. **By implement Servlet interface:**



Here, we created a class MyServlet which implements **Servlet** interface, but there is compile time error. To remove this we have to override five methods of **servlet interface.**

****

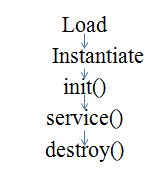
**In Servlet there are 5 methods:**

1. init()
2. service()
3. destroy()
4. getServletInfo()
5. getServletConfig()

init(), service() and destroy() are **Life-cycle methods** and getServletInfo(), getServletConfig are **Non Life-cycle methods.**

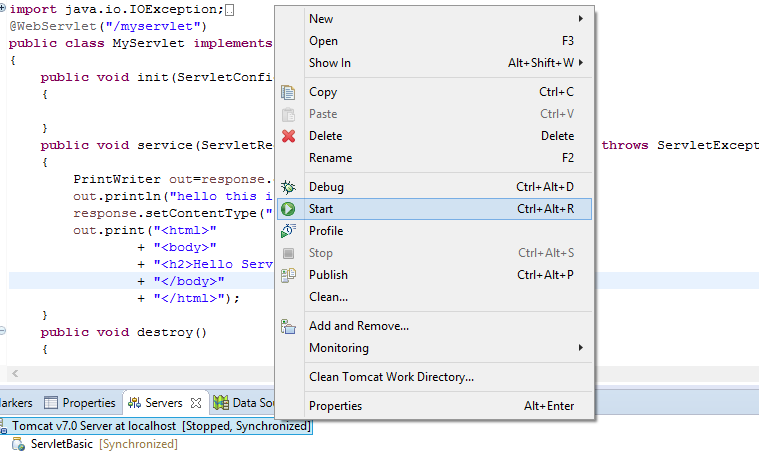
**Note: init() and destroy() execute at once while service() executes on every request.**

**Life-cycle of Servlet:** When user send request to server by hitting URL in browser, this request goes to web container of server. Web container handle life cycle of servlet.

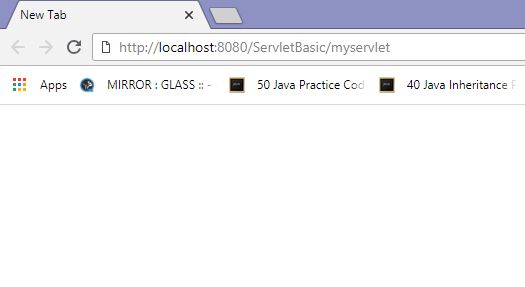
****

1. Load: Servlet is load at first request of that servlet by user to server.Class loader is responsible to load class.
2. Instantiation: After class loading instance or object of Servlet is created by web container. Object of servlet is created onlyonce time in the lifecycle of Servlet.
3. init(): init() is used to initialize instance of Servlet. Web container calls init().init method is call only one time for all request
4. service():

**Example 1: 1.1: previously we have deployed our project on server. Now start the Server.**

****

**1.2: Open the browser and post the URL on URL bar:**

****

**Output:**

****

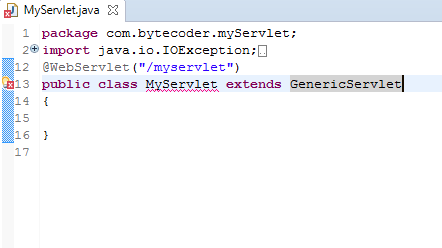
**Note:**

**URL:** localhost:portno./projectname/url\_name

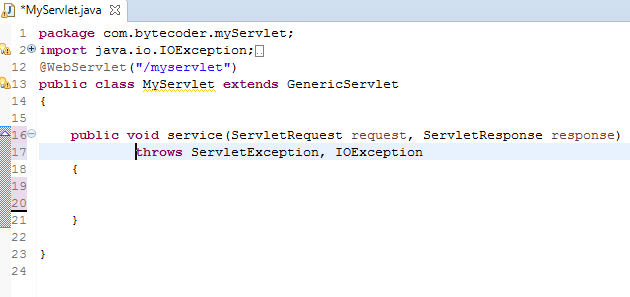
**MIME type:** MIME is acronym for Multiple Internet Mark Extension. It shows content type in response. Ex:text/html, text/plain etc.

**Annotation:** @webservlet(“/url\_name”)

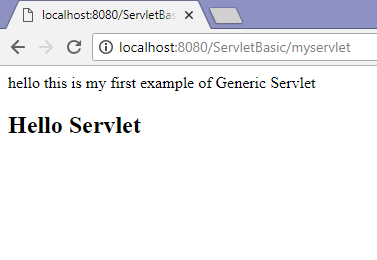
1. **By extending GenericServlet class:**With extending GenericServlet class there is also compile time error.

****

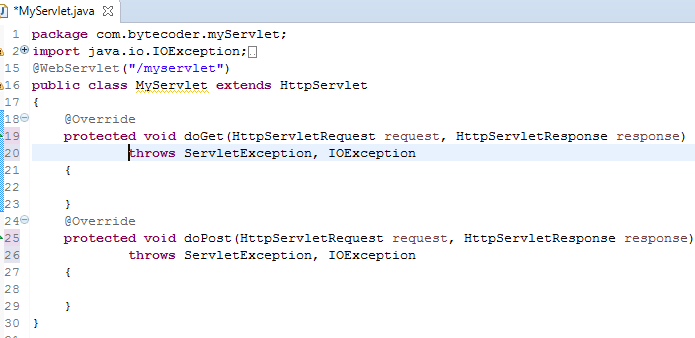
We have to override service()

****

**Output:**

****

1. **By extending HttpServlet:**

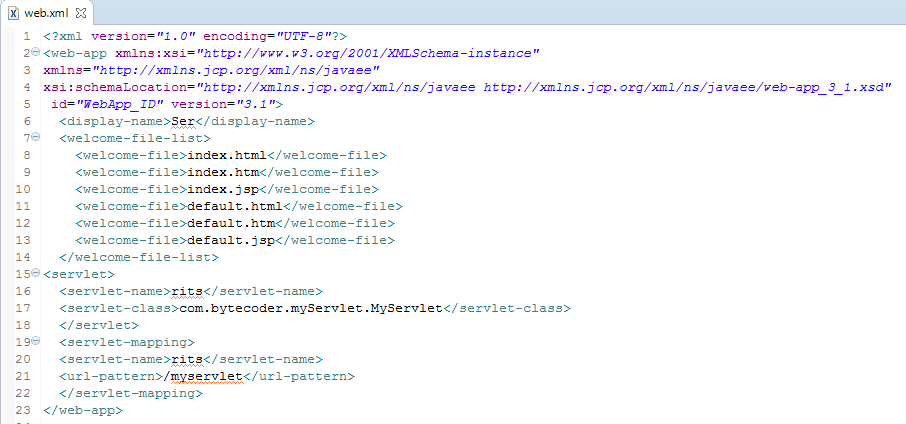
****

**Output:**

****

**Deployment descriptor:** web.xml is known as deployment descriptor.

Before servlet 3.0 we use xml file for mapping:



**Lazy Loading**: Lazy loading is on demand loading, when user made a request on servlet then it servlet is being loaded.

**Eager Loading:** If you have multiple servlets and you want to load any servlet on your priority so itloads the servlet with the loading time of server.We use one tag for eager loading. Container loads the value according to priority of given integer in body of tag.

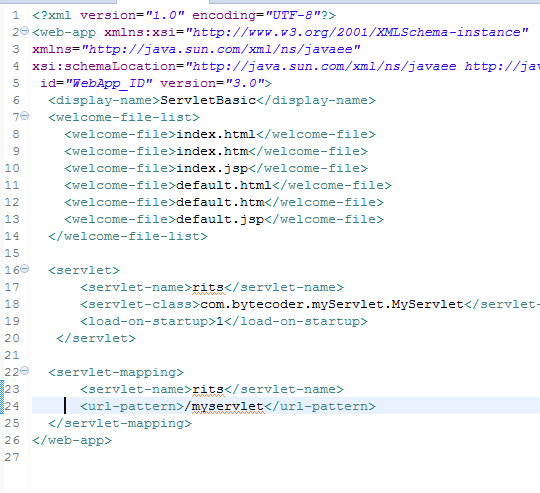
<load-on-startup>1</load-on-startup>

**Annotation for load-on-startup:**

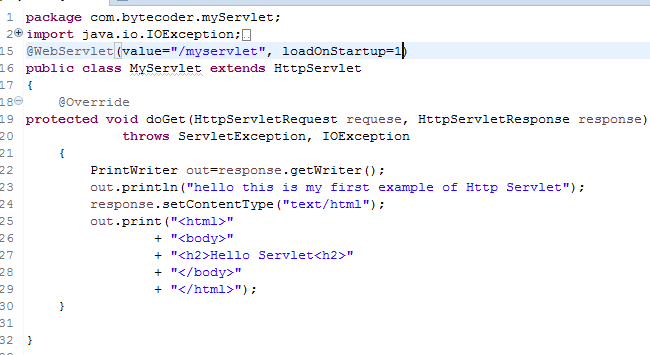
**@**webServlet(value=”/url\_name”, loadonstartup=1)

**Note\*\*\* if you pass negative value, servlet will loaded at the time of first request.**

**\*\*\*Use of <load-on-startup> tag:**



**\*\*\*Use of load-on-startup annotation:**



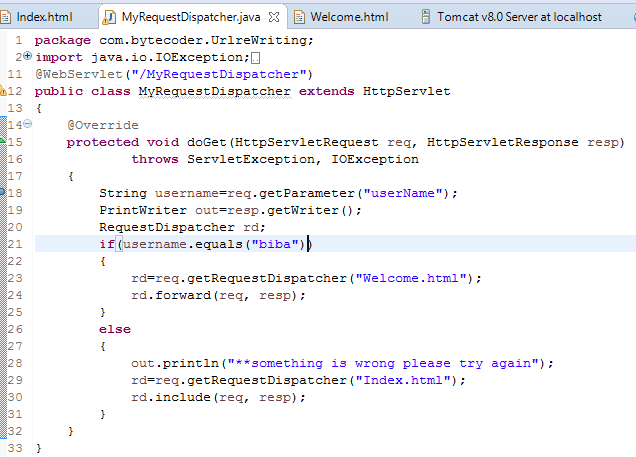
**\*\*Request Dispatcher**:

* Forward
* Include

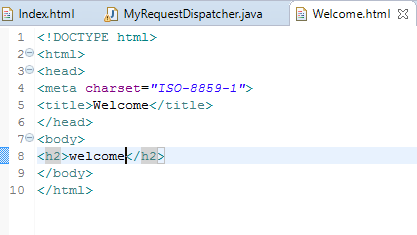
**Index.html**



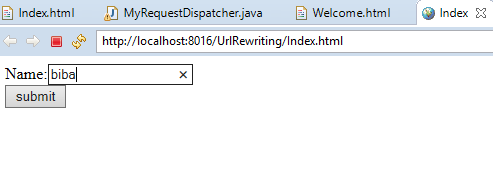
**MyRequestDispatcher.java**



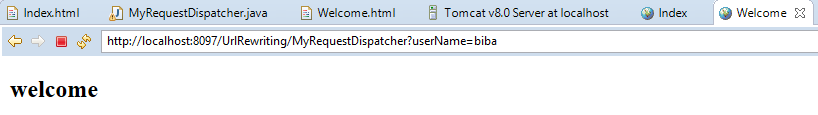
**Welcome.html:**

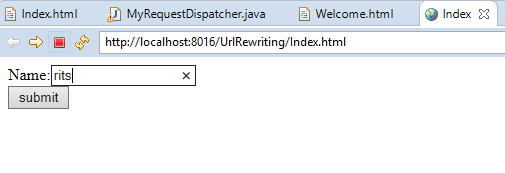


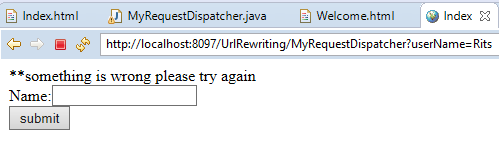
**Output:**



We entered biba as username, so it forwarded on welcome page.



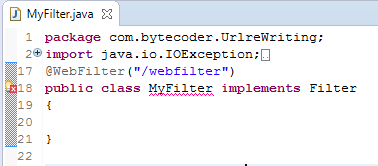




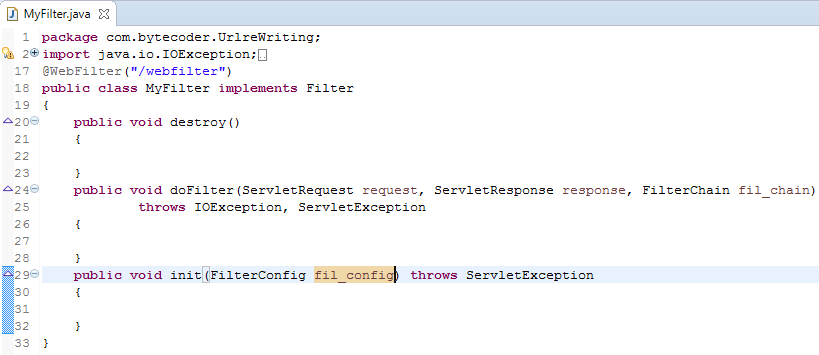
**Filter:** Filter is used for pre and post processing. Filter is pluggable. For using functionality of filter we have to implements Filter interface.

**Annotation for filter:** @webfilter(“/url\_name”)

**Example:**

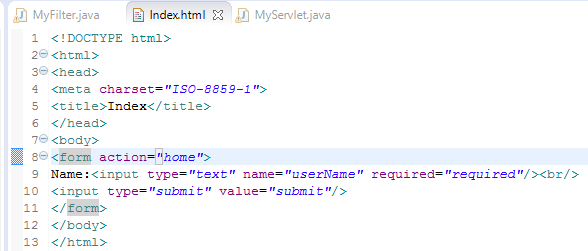
****

For removing this error we have to override its methods:

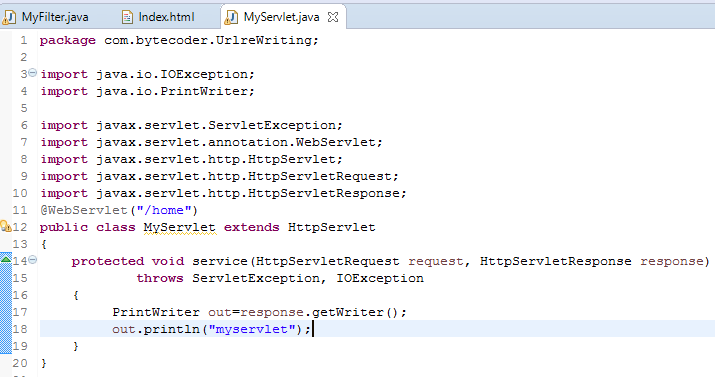


**Example of filter:**

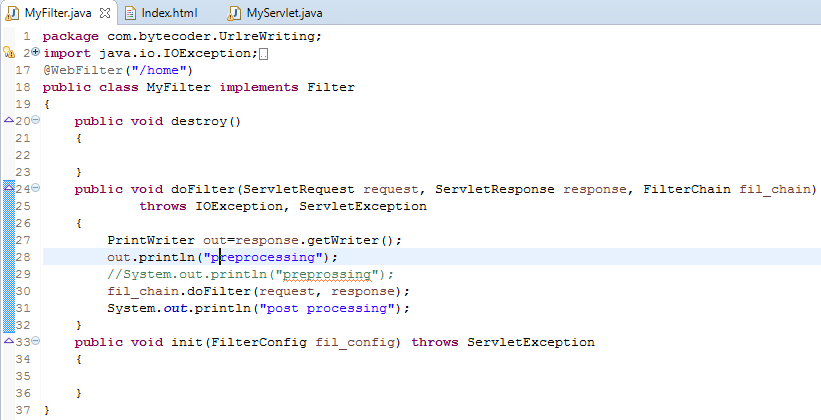
**Index.html**

****

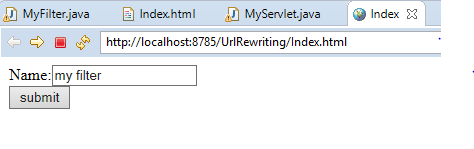
**MyServlet.java**

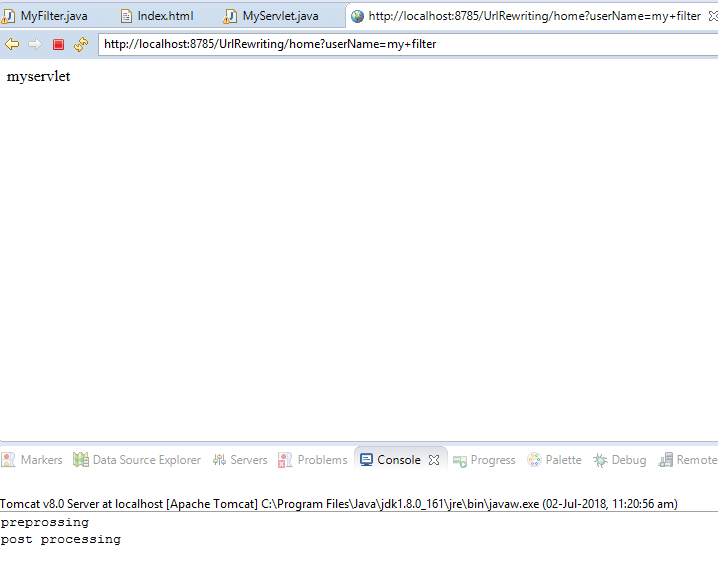
****

**MyFilter.java**

****

**Output:**

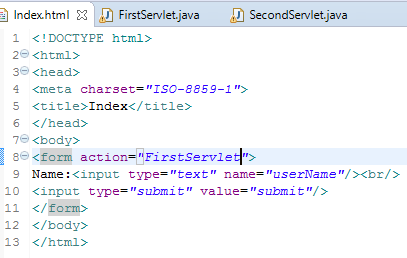
****

****

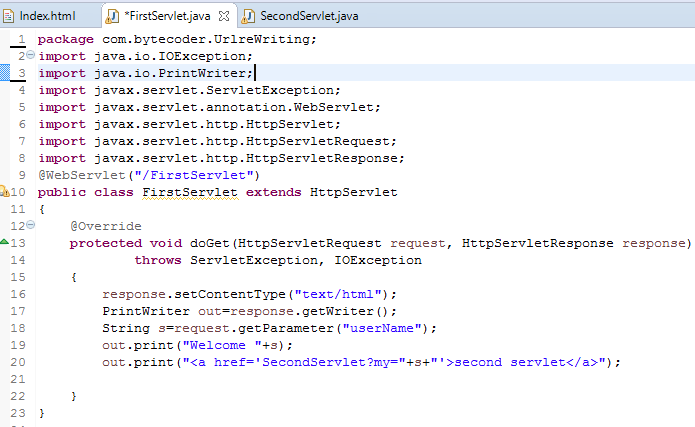
**Session tracking technique:**

1. **URL rewriting**
2. **HTTP session**
3. **Hidden from field**
4. **Cookies**
5. **URL rewriting:**

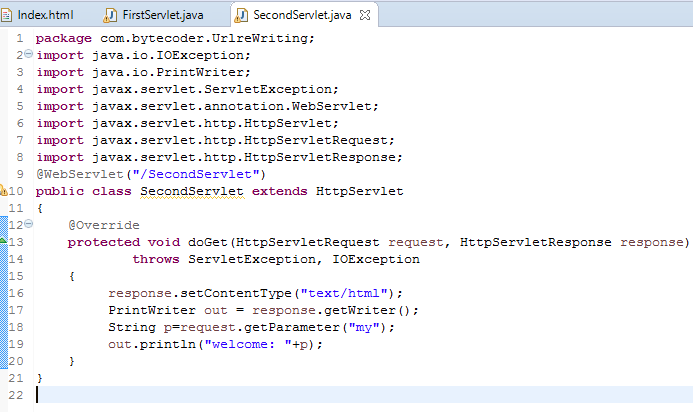
**Index.html**

****

**FirstServlet:**

****

**Second Servlet:**

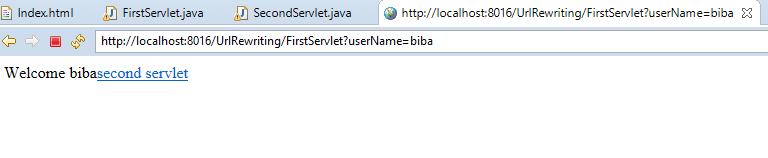
****

**Output:**

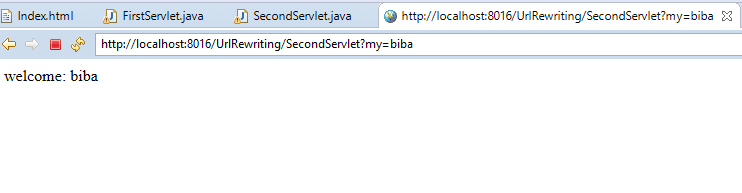
**1.**

****

**2.**

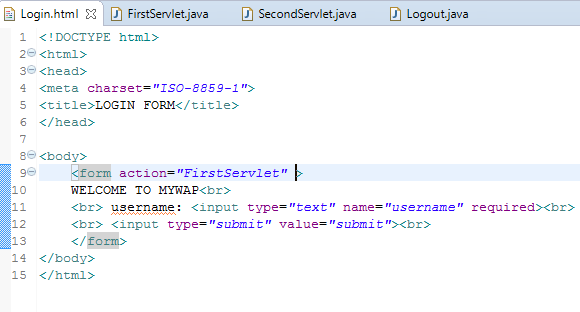
****

**3.**

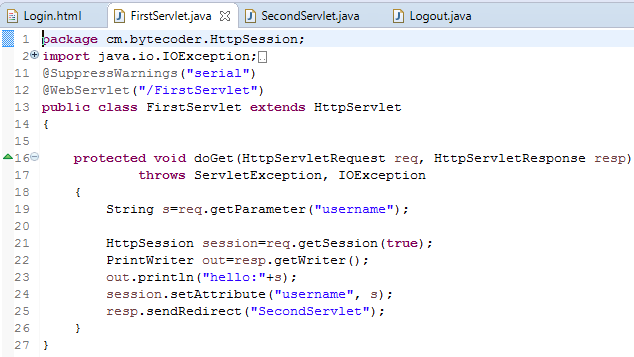
****

1. **HTTP session:**

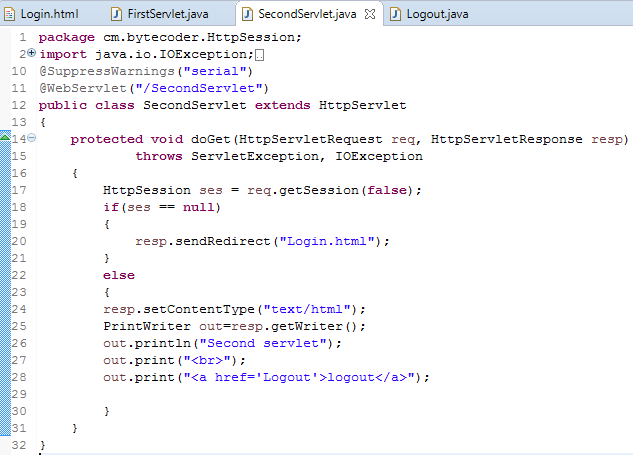
**Login.html**

****

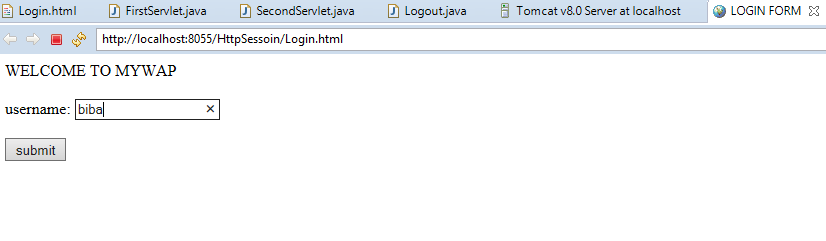
**First Servlet:**

****

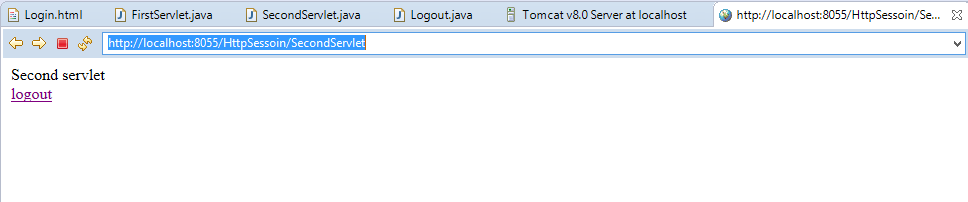
**Second Servlet:**

****

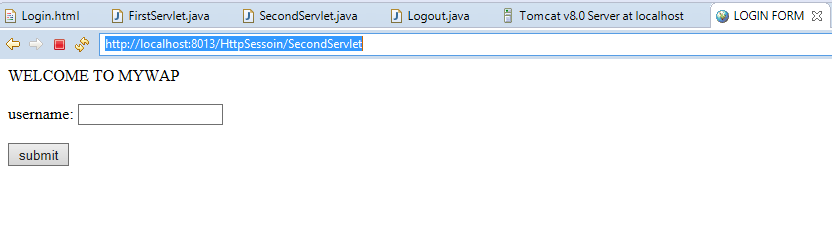
**Output:**

****

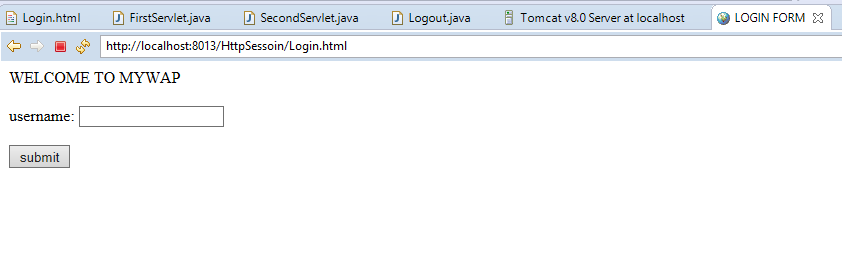
Copy the path of second servlet:

****

Paste the copied path here, and press enter.

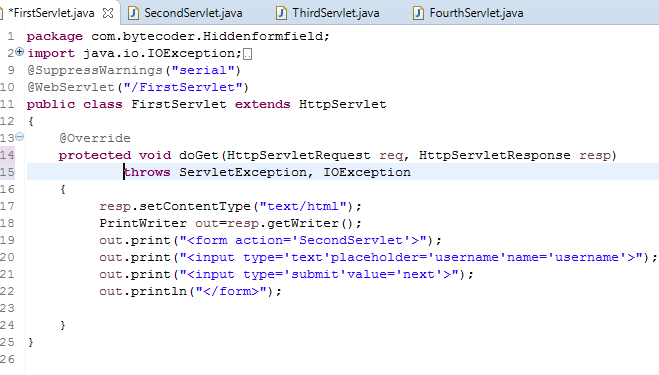
****

It will again sent you to login page if once you log out

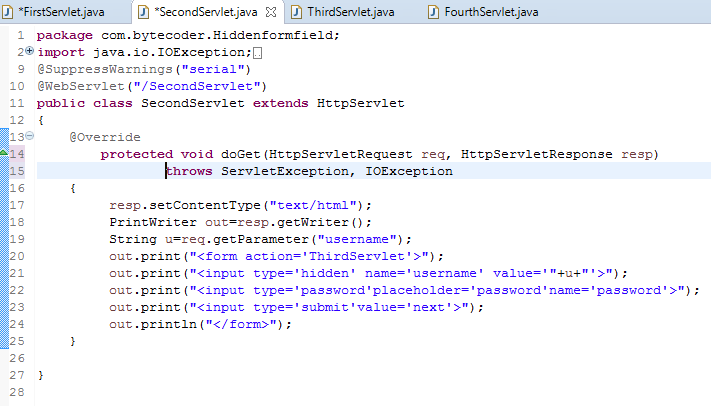
****

1. **Hidden from field:**

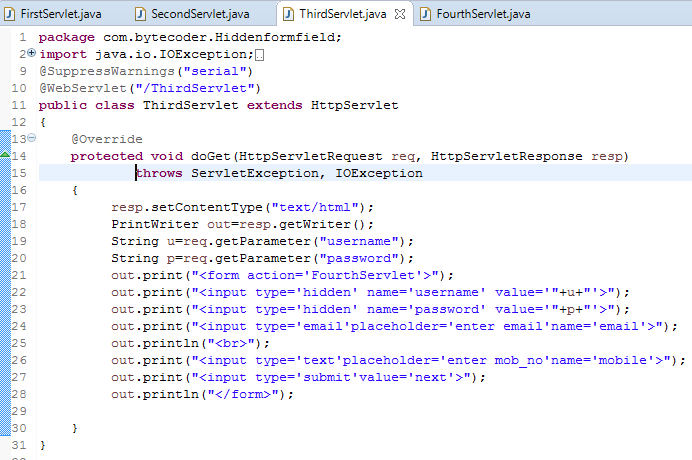
**Servlet 1:**

****

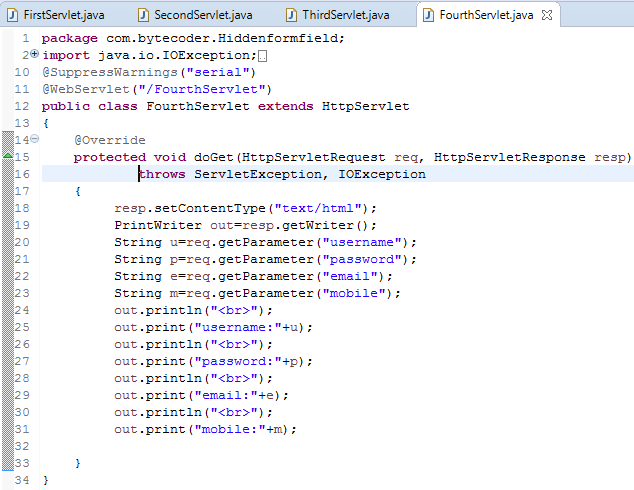
**Servlet 2:**

****

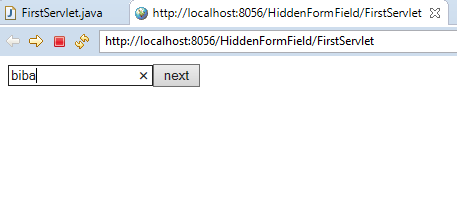
**Servlet 3:**

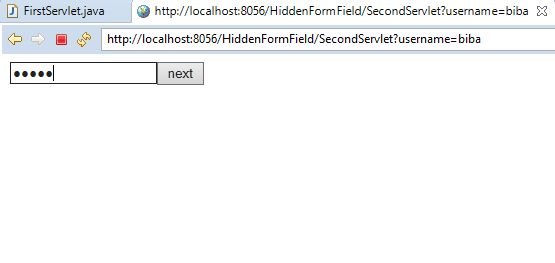
****

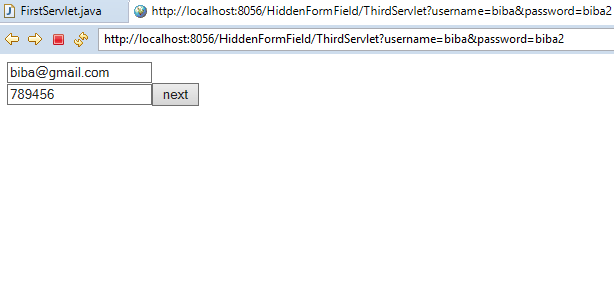
**Servlet 4:**

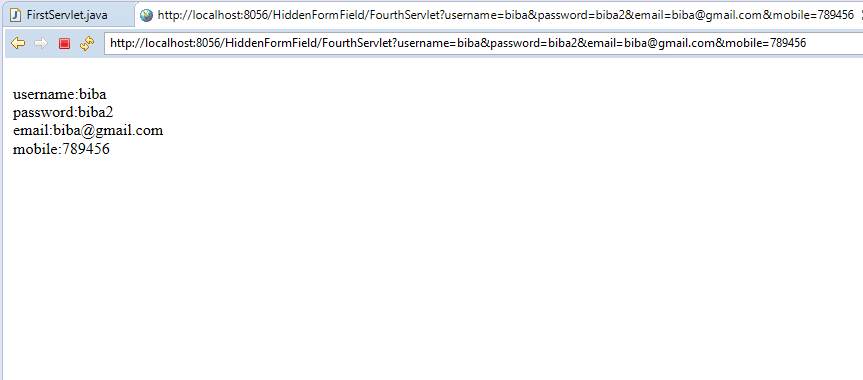
****

**Output:**

****

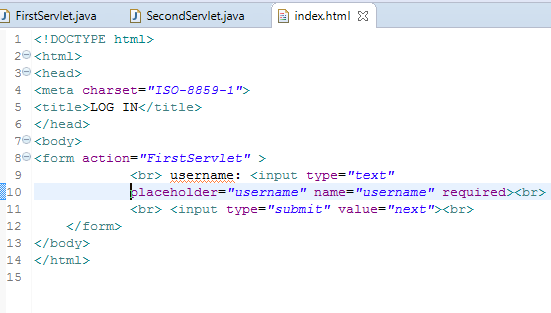
****

****

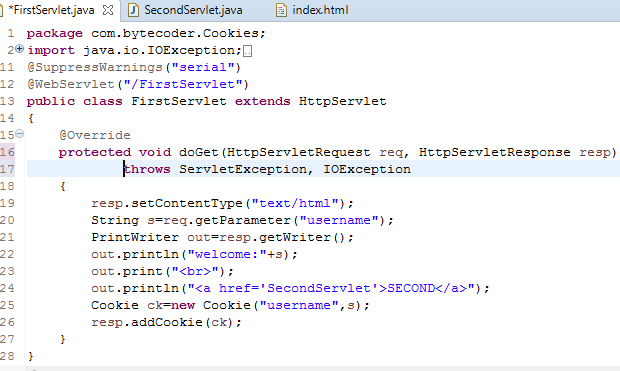
****

1. **Cookies:**

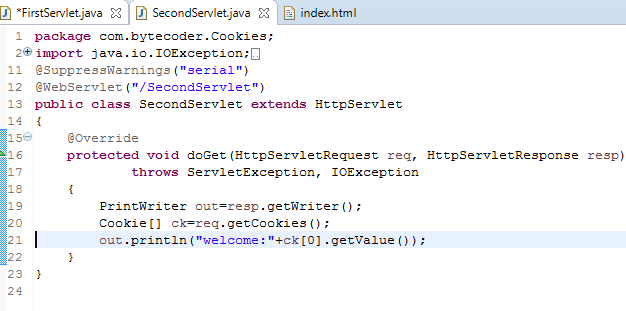
**4.1: Index.html**

****

**4.2: first servlet**

****

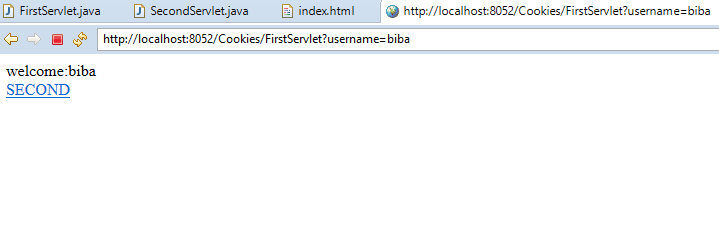
**4.3: second servlet**

****

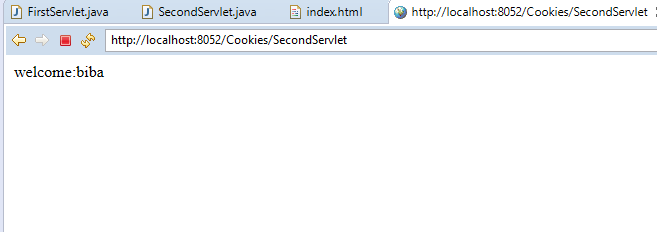
**Output :1**

****

**2**

****

**3**

****