

MongoDbServlet

1 Day Work

- 1-What is Web Application
- 2-Types Of Server
- 3-Servlet Introduction
- 4-Ways For Creating Servlet
- 5-Servlet First Demo Program
- 6-Attach Html with Servlet
- 7-Data Transfer From One to Another Servlet

Web Application

A Web Application or Web App is Software that runs on a Server.

Types Of Server

1-Web Server

2-Application Server

Web Server

Web server contains only web or servlet container. It can be used for servlet, jsp, struts, jsf etc. It can't be used for EJB.

Application Server

Application server contains Web and EJB containers. It can be used for servlet, jsp, struts, jsf, ejb etc. It is a component based product that lies in the middle-tier of a server centric architecture.

Servlet Introduction

Servlet technology is used to create a web application (resides at server side and generates a dynamic web page).

Ways For Creating Servlet

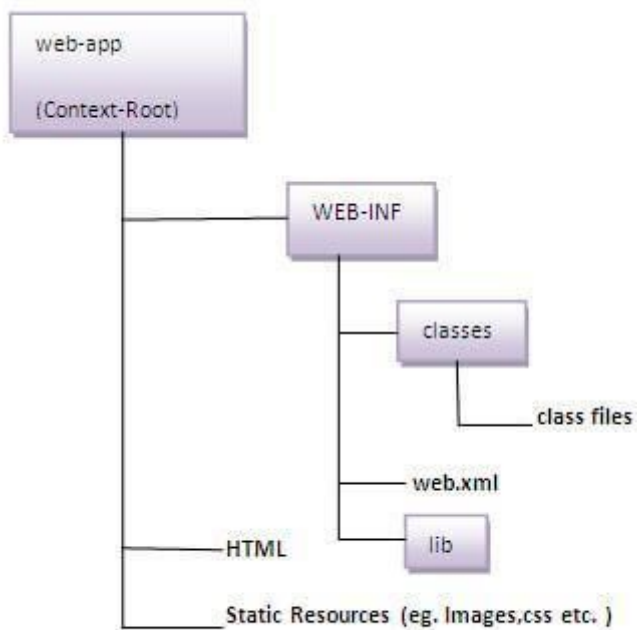
- 1- By Implementing Servlet class
- 2- By extending Servlet class

3- By extending HttpServlet

Servlet First Demo Program

Story 1-

Create Directory Structure



Task 1-Install Tomcat Server In your System

[Link of Tomcat Installation Process](#)

Task 2-Download jar file of Servlet form this link

[Link to download Servlet.jar file](#)

Task 3-Set classpath of servlet jar file using this command open terminal in your root Folder and run this command and give the path of jar file

export CLASSPATH=\$CLASSPATH:/home/kls/workspace/servlet jar file name:.

Here /home/kls/workspace your location of your jar file

Task 4- Here Servlet Demo is our root folder



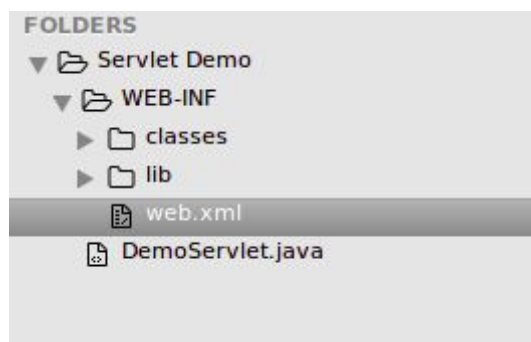
Task 6- In root folder we have one java file and WEB-INF folder



Task 7- In WEB-INF folder we have two folders classes and lib and one xml file name web.xml

classes folder: In this folder we put our java .class files

lib: In this folder we put our jar files



web.xml: in this file we map our servlet .

Story 2-

Run your first Servlet program

Task 1- Create .java file in root folder name DemoServlet

In this file we import packages as HttpServlet, io and servlet and extends our class from HttpServlet class. Here we must throws Exception

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;
public class DemoServlet extends HttpServlet{
public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
{
res.setContentType("text/html");//setting the content type
PrintWriter pw=res.getWriter();//get the stream to write the data
//writing html in the stream
pw.println("<html><body>");
pw.println("Welcome to servlet");
pw.println("</body></html>");
pw.close();//closing the stream }}
```

Task 2- Create web.xml file parallel to classes folder in WEB-INF. Here <web-app> our root tag in the xml we mapped our servlet

```
<web-app>
<servlet>
<servlet-name>DemoServlet</servlet-name>
<servlet-class>DemoServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>DemoServlet</servlet-name>
<url-pattern>/demo</url-pattern>
</servlet-mapping>
</web-app>
```

Task 3- Open terminal(in your root folder) and run this command to
Compile your . java file

```
javac -d WEB-INF/classes DemoServlet.java
```

Task 4- After creating .class file we run this command on same terminal to make
war file

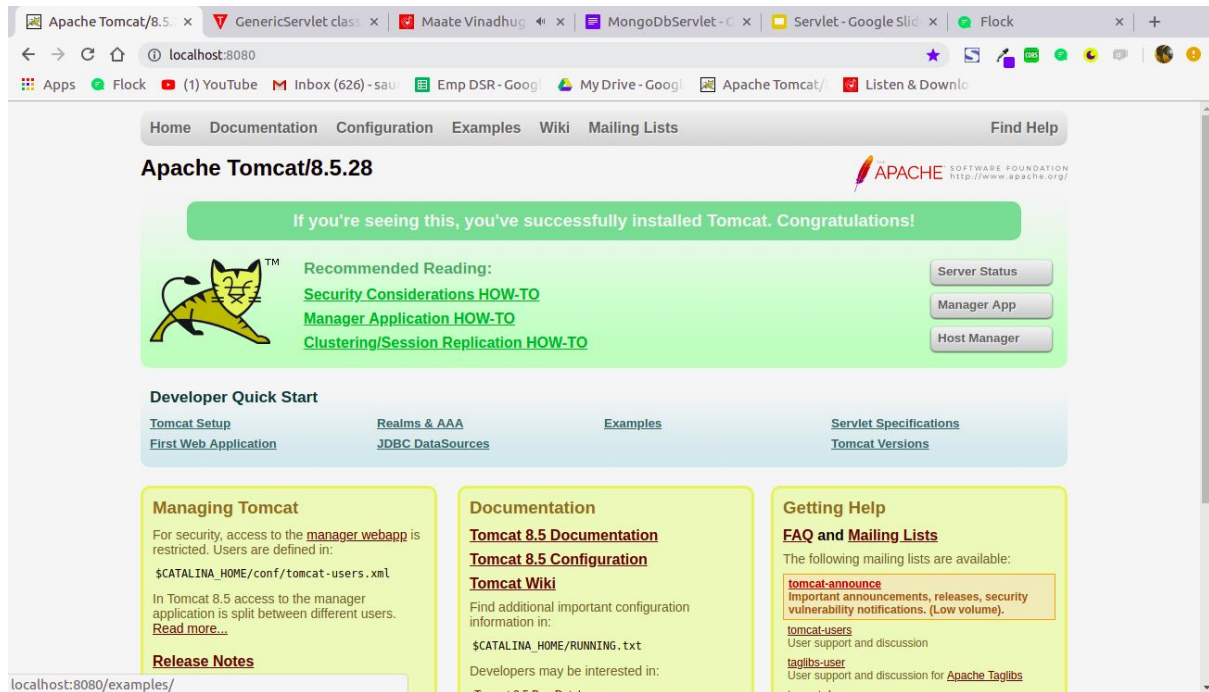
```
$ jar -cvf demo.war *
```

Task 5- After making war file we Start Tomcat Server by using this command

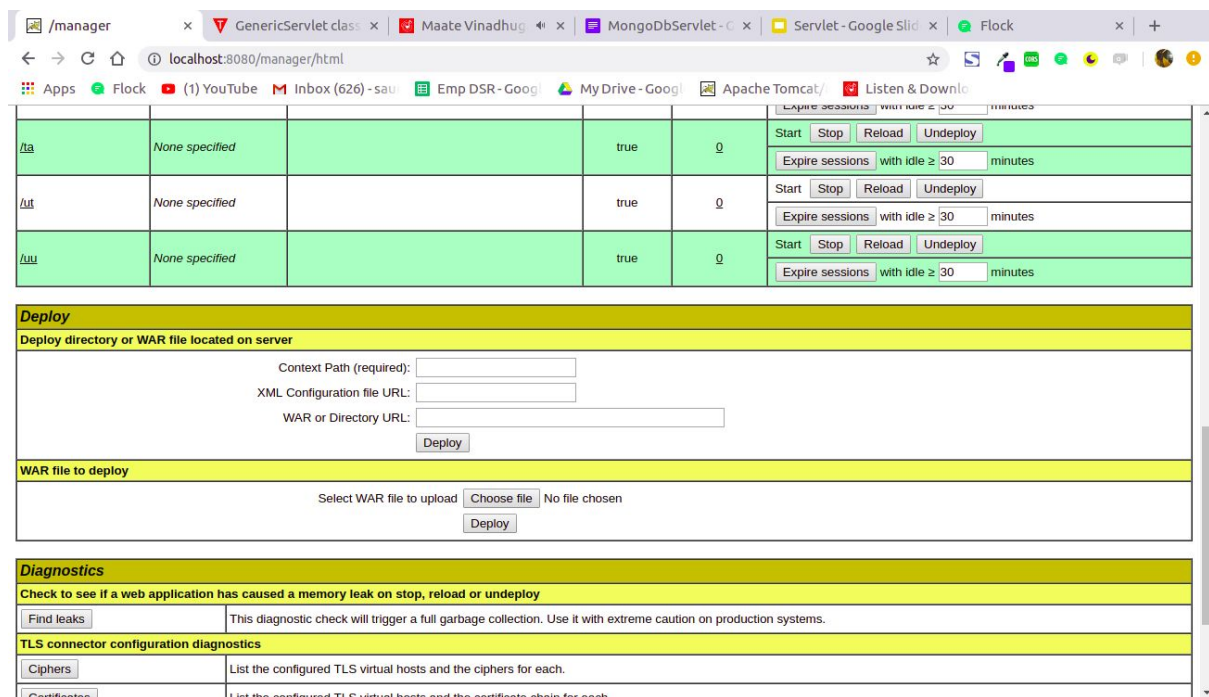
```
$ sudo systemctl start tomcat
```

Story 3- To run our project

Task 1- Deploy your war file to Tomcat server goto the browser and type
<http://localhost:8080> in your address bar after that this page will open



Task 2- Goto Manager App after next page open goto below of page



Task 3- Choose File-- Here you choose your war file and after that click on deploy key

Browser tabs: /manager, GenericServlet class, Dil Mein Ho Tum, MongoDBServlet - C, Servlet - Google Slid, Flock

Address bar: localhost:8080/manager/html

Message: OK

Manager					
List Applications	HTML Manager Help		Manager Help		Server Status
Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/aa	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/complete	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/demoservlet	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/filter	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Task 4 - Here demo your war file click on this war file and your project will start

Browser tabs: CBSE Exerci, CBSE Exerci, localhost:80, load on start, MongoDBSe, MongoDBSe, Flock, Webmail Loc

Address bar: localhost:8080/demo/

Message: OK

Hello Servlet

Story 4 – Html in Servlet

Task 1 -You can defined html in servlet as like below

In Servlet html written inside the out.println()

```
PrintWriter out=res.getWriter();
out.println("<html><body
style='background-color:darkorange;height:200px';>");
out.println("<head><h1 align='center'
style='margin-top:30px;'><u>User LogIn</u></h1></head>");
out.println("<table width=30%
style='margin-top:100px;height:200px' align='center'
border=1>");
out.println("<form action='verify'><tr><td>Enter
Username</td><td><input type='text' name='name'></td></tr>");
out.println("<tr><td>Enter Password</td><td><input
type='password' name='pass'></td></tr></table>");
out.println("<input type='submit'
style='color:blue;margin-left:580px;margin-top:30px;width:90p
x;height:40px;' value='LogIn'></form>");
}
```

1.

