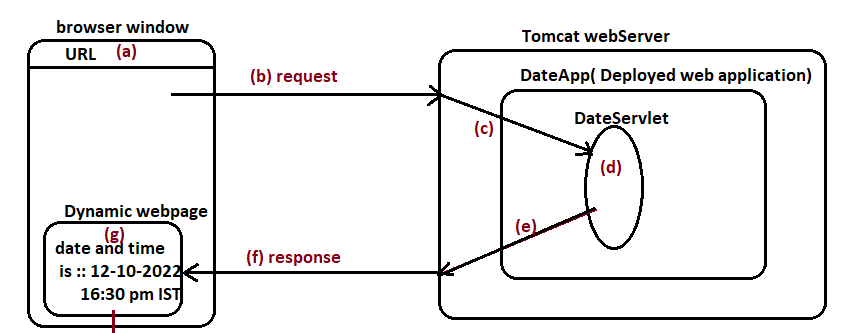
**1. Procedure to develop the web-app having servlet component as Dynamic web component.**

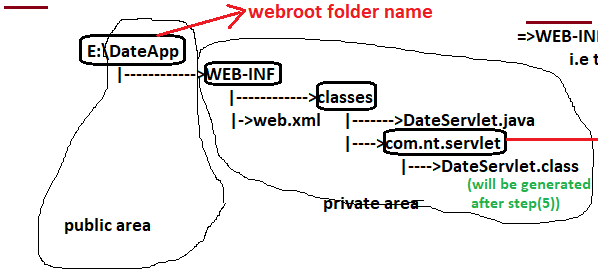
**2. Important points.**

**1.Procedure to develop the web-app having servlet component as Dynamic web component**

**DateApp**  is web application having the DateServlet component as dynamic web component.

DateServlet component generates the dynamic web page having the System date and time as content.

**Step:1** Create a Deployment directory structure.

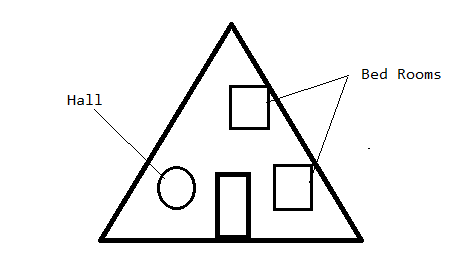


1. WEB-INF and its sub-folders are called private area of web application. The content of private-area can only be accessed by Underlying web server or web container. The developer keep the servlet source code and byte code in

WEB-INF/classes to protect them from the hackers.

1. The out-side folders of WEB-INF is called public area of web application. The content of public-area can be accessed by anyone without taking underlying web server or web container permission. The .html files,image files, audio files, video files , jsp files and …etc are placed in public area of web application because these file should go to the browser. Therefore there is no meaning of protecting source code by keeping in private area.

Example:-



House is web application.

Bed Rooms are WEB\_INF and its sub folders.

Hall is Out side folders of WEB\_INF.

1. Web.xml file name and location is fixed. This file is also called as “deployment descriptor”. This file is also “configuration file” of web application Because it contains various instruction to container and server about various web components of web application.

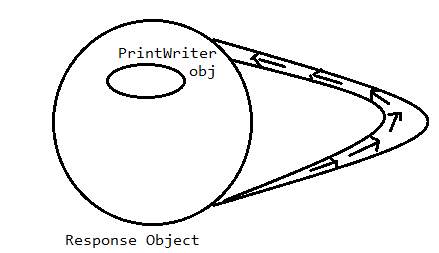
Which URL based request should be linked/mapped to which servlet component in private area will be configured by writing xml entries in web.xml file.

This file is read and used by servlet container automatically when the moment web application deployed in web server.

**Step2:**- Develop the servlet component in WEB-INF/classes folder.

We must take servlet component class as public to visible to servlet container to perform the loading and instantiation .

**Step2.1**. Response Object



The response object Stream points to itself.

**Responseobject . getWriter()**  method gives the printWriter Stream object reference.

**Step2.2.** Browser will display the received content of response as plain text on browser (i.e) html tags will not be recognized and html tags will be considered as plain text.

Responseobject.setContentType(“text/html”/

“text/plain”/

“Application/Vnd-ms-excel”/

“Application/ms-word”/

“text/xml);

This method gives instruction to browser from servlet component through response object, web container and web server to receive the content of response as specified and display response in specified format.

MIME (Multipurpose internet Mail extension)types are

text/html,

text/plain,

Application/Vnd-ms-excel,

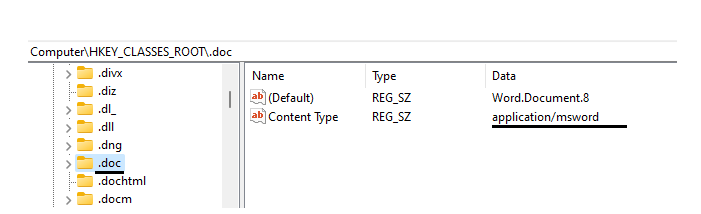
Application/ms-word,

“text/xml..etc

Note:- If no content type is set in servlet component , browser uses Default MIME type is text/html.

**From window OS, we get MIME types in following way:**

Windows+r 🡪 Type regedit 🡪 HKEY\_CLASSES\_ROOT 🡪 select extension and get MIME Type.



**Step 2.3.** Write the output/result of servlet component class to response object using println(-)method.

package com.nt.servlet;

import jakarta.servlet.\*;

import java.util.\*;

import java.io.\*;

public class DateServlet extends GenericServlet{

public void service(ServletRequest req,ServletResponse res)throws ServletException,IOException{

PrintWriter pw=res.getWriter();

res.setContentType("text/html");

Date d=new Date();

pw.println("<b> Today Date & Time is:"+d+"</h2>");

pw.close();

}

}

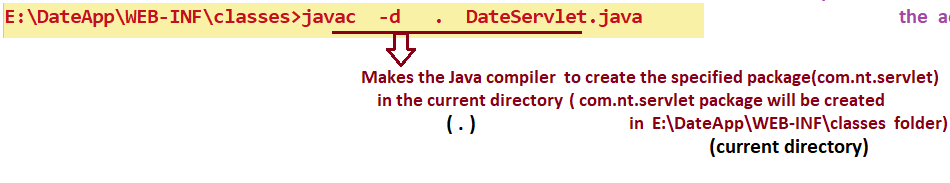
**Step 3.**



The servlet-api.jar is not part of JDK.

This api is in Java based web server.

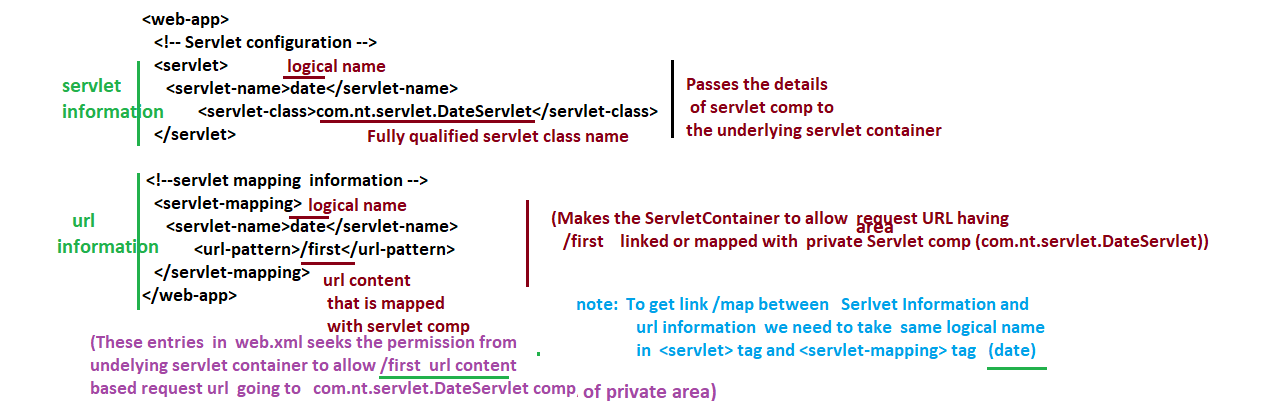
**Step 4.** Compile the servlet component class.



**Step5.** Create the web.xml file in WEB-INF folder.

**Servlet config means**:- Passing the information about servlet component class to servlet container.

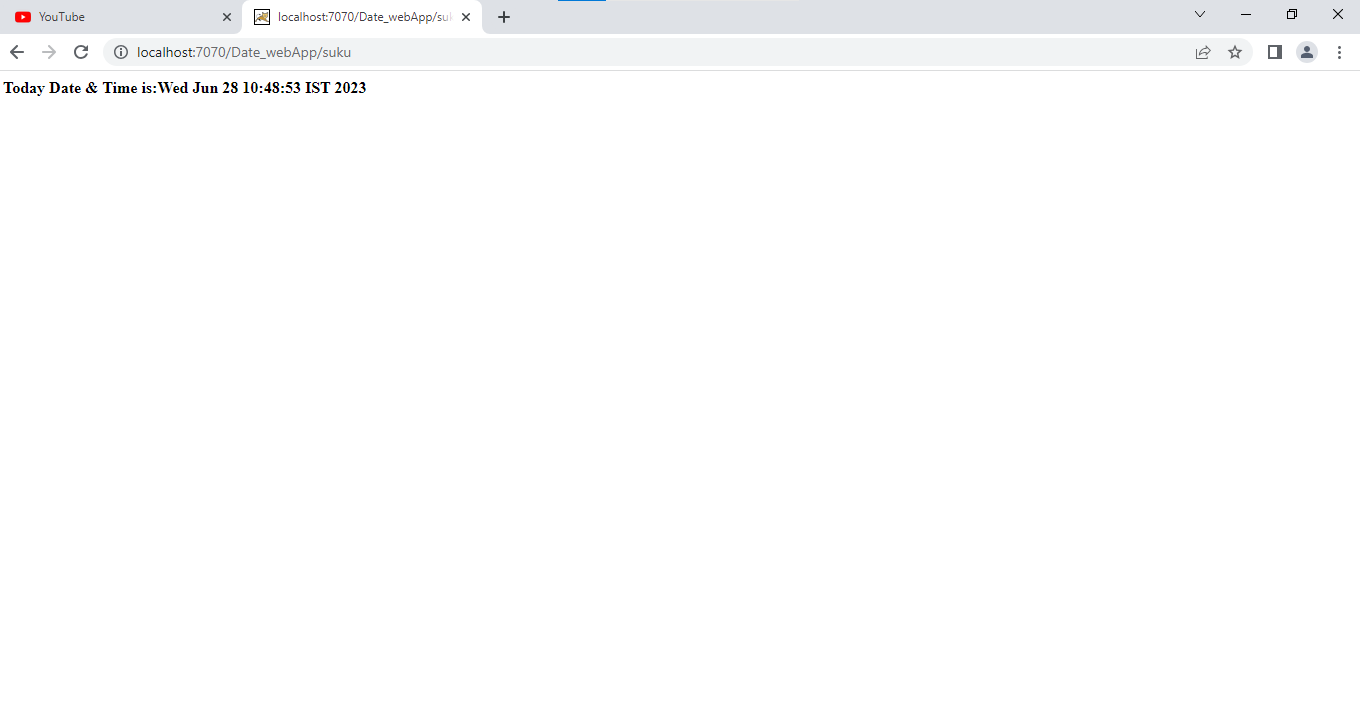
**Servlet Mapping Means**:- Linking the Browser generated URL request with servlet component in private area of web application.



**Step6: start the Tomcat web server.**

**Step7: Deploy the web application in web server.**

**Step8: Test the Deployed web application.**



**2. Important Points:**

**Note1:-**After deploying the web application , if we want to modify the source code of servlet applet then those changes reflects in browser output only after recompilation of servlet component class and **reloading** of the web application.

Reloading web application means stopping the web application and starting the web application.

**Step1:-** make sure the Tomcat server is started and web application is deployed.

**Step2:-** After doing the modification in servlet component of deployed web application , recompile the servlet component.

pw.println("<h2 style='color:red'>Today Date & Time is:"+d+"</h2>");

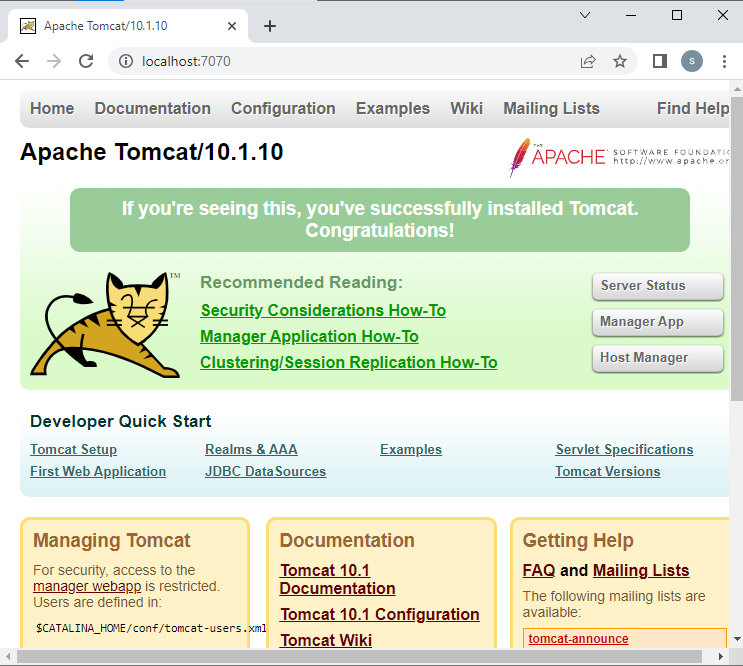
C:\Tomcat 10.1\webapps\Date\_webApp\WEB-INF\classes>set classpath=.;C:\jdk-19\bin;C:\Tomcat 10.1\lib\servlet-api.jar;

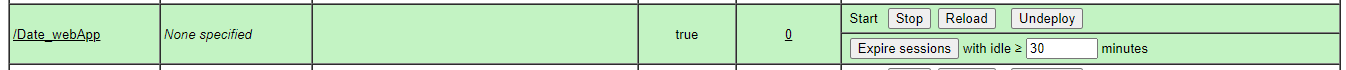
C:\Tomcat 10.1\webapps\Date\_webApp\WEB-INF\classes>javac -d . DateServlet.java

C:\Tomcat 10.1\webapps\Date\_webApp\WEB-INF\classes>

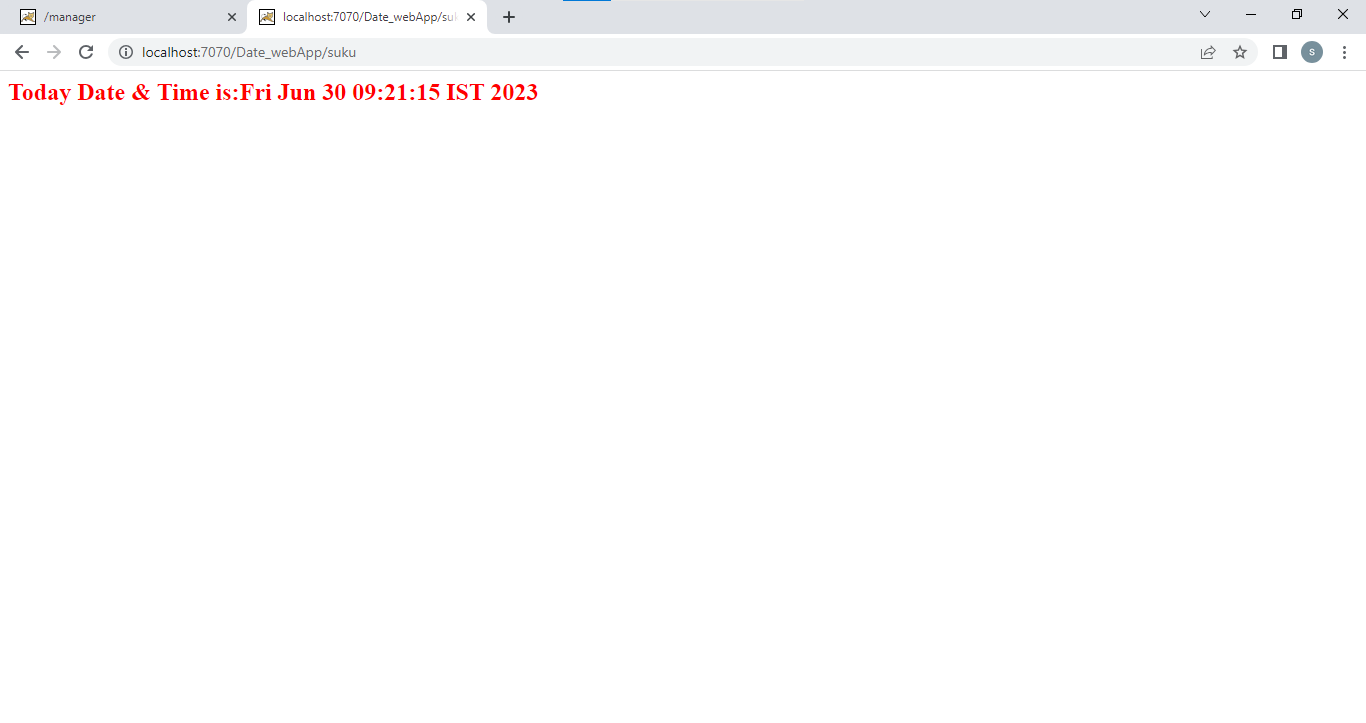
**Step3:-** Reload the web application.

<http://localhost:7070> 🡪 ManagerApp 🡪 GoTo corresponding web app 🡪 click the Reload button.





**Step4: Tes**t the Changes.



**Note2:-** The modification in web.xml and in static web components will be recognized automatically by underlying web container or web server automatically. Implicitly, the web application is reloaded.

**Note3:- 500 Error raises the following reason.**

If servlet container fails to create object for servlet class then 500 error raises.

1. If servlet component is not taken as “public”.
2. If we pass wrong class name in <servlet-name> tag in web.xml file.
3. If servlet class has non-zero param constructor.

**Note4:- 400 Error raises the following reason**.

1. If we place the wrong the URL in address bar of browser window.
2. In web.xml ,If logical name given in <servlet-name> of <servlet> tage is not match with name given in <servlet-name> of <servlet-mapping> tag.

**Note5:- in** servlet class, System.out.println(-) method writes the message on console window of Tomcat server.

**Note6:-** The servlet component class is not standard alone application because that class does not has main(-) method. It can not be executed from command prompt.