

**TO
THE
NEW™**



Assessment -9

Application SERVER

Trainee Name : Arun Parmar

Mentor Name : Ravi Kumar

College: UPES

1. What is the difference between an Application Server and a Web Server?
 - Web server is useful or fitted for static content, whereas application server is fitted for dynamic content.
 - In web servers, multithreading is not supported, while in application server, multithreading is supported.
 - Application Server supports distributed transaction and EJB. While Web Server only supports Servlets and JSP.
 - Application Server can contain web server in them. most of App server e.g. JBoss or WAS has Servlet and JSP container.
 - Though its not limited to Application Server but they used to provide services like Connection pooling, Transaction management, messaging, clustering, load balancing and persistence. Now Apache tomcat also provides connection pooling.
 - In terms of logical difference between web server and application server. web server is supposed to provide http protocol level service while application server provides support to web service and expose business level service e.g. EJB.
 - Application server are more heavy than web server in terms of resource utilization.

2. What is Catalina?

Catalina is Tomcat's servlet container. Catalina implements Sun Microsystems' specifications for servlet and JavaServer Pages (JSP). In Tomcat, a Realm element represents a "database" of usernames, passwords, and roles (similar to Unix groups) assigned to those users. Different implementations of Realm allow Catalina to be integrated into environments where such authentication information is already being created and maintained, and then use that information to implement Container Managed Security as described in the Servlet Specification.

3. Describe tomcat directory structure.

The webapp directory is held in the coursework directory within the "courses" directory and is named after the coursework. If the coursework's name is D2, the coursework directory is "courses/D2Home" in the ExerTran installation directory, and the webapp directory is named "courses/D2Home/D2". The directory structure is as follows:-

jsp

This directory holds all the jsp files for the application, both system jsp files copied from the "src/jsp" directory (), but also question related jsp files from packages imported into the coursework or generated by the coursework lecturer (the chapter called Topic Packages). The directory structure of this directory is the same as that of the "src/jsp" directory, except that the "jsp/user/" directory also hold jsp files from the packages: with directories structured by package name.

images

This hold image files, primarily "gif", for particular questions. This directory is structured by package name.

html

This hold pure html files for particular questions. This directory is structured by package name.

WEB-INF

Tomcat hides the contents of this directory from users, and is the location where Java class files are stored as well as the Tomcat "web.xml" file which defines a number of parameters for the application in particular security information and the mapping of user requests, i.e. URIs, to servlets. The contents of this directory are as follows:-

web.xml

This is a key file for running a Tomcat application and defines various features of the application. This file is created automatically during a coursework build from a template file, "web.tmpl", held in "src/templates", with substitution of parameters defined in the coursework definition file (the chapter called Coursework Definition File. The contents of this file are described in .

classes

This holds the Java classes for the application, both system jsp files compiled from the "src/java" directory (), but also from the question related java files from packages imported into the coursework or generated by the coursework lecturer (the chapter called Topic Packages). The directory structure of this directory is the same as that of the "src/java" directory, except that the "java/user/" directory also hold java files from the packages: with directories structured by package name.

help

This holds the help for the application, both system help files compiled from the "src/help" directory (), but also from the question related help files from packages imported into the coursework or generated by the coursework lecturer (the chapter called Topic Packages). The structure of the help files is by question package, with the system help files in the directory "system".

logs

This holds the log files produced from the execution of a coursework. There are 3 basic log files, usually called applicationlog.text, activitylog.txt, and mysqlLog.txt. The first of these logs debug messages from the coursework; the second logs user activity, particularly question attempts and marks awarded; the third error messages from the mysql database, which is the default database. When the coursework is re-started, these log files are compressed when they are larger then a set limit, and versions created. It may be necessary to change the compression to occur while the system is running, if regular closure of the application does not occur. A further log file is also created when the connection to the marks database fails and marks cannot be stored in the database. The marks are then recorded in a log file, from which they are read stored to the database when the coursework system is restarted: on successful storage of these marks, the log file is removed.

lib

This contains jar files needed by Tomcat to run the @systemname; application. These are copied from "src/lib" on a coursework build.

images-xml

This hold Xml files that define an image to be rendered by the "synDrawings" package: these images can be modified at run-time in response to user input. This directory is structured by package name.

initParameter

This holds files holding name-value pairs for use by jsp and java packages so that the ExerTran application can be varied without editing the program sources.

xml

This holds Xml files with definitions of the database table columns. Changes to these files changes the structure of the database table on the next database write: note that this has only been tested for adding extra columns to a table not for removing columns. Note that the only database table that needs creating to start running a coursework is a "users" table, defined by "userd_def.xml" in the directory, with a manager user and password, all other tables can be created from the manager interface by saving to the database, even if there are no rows to save.

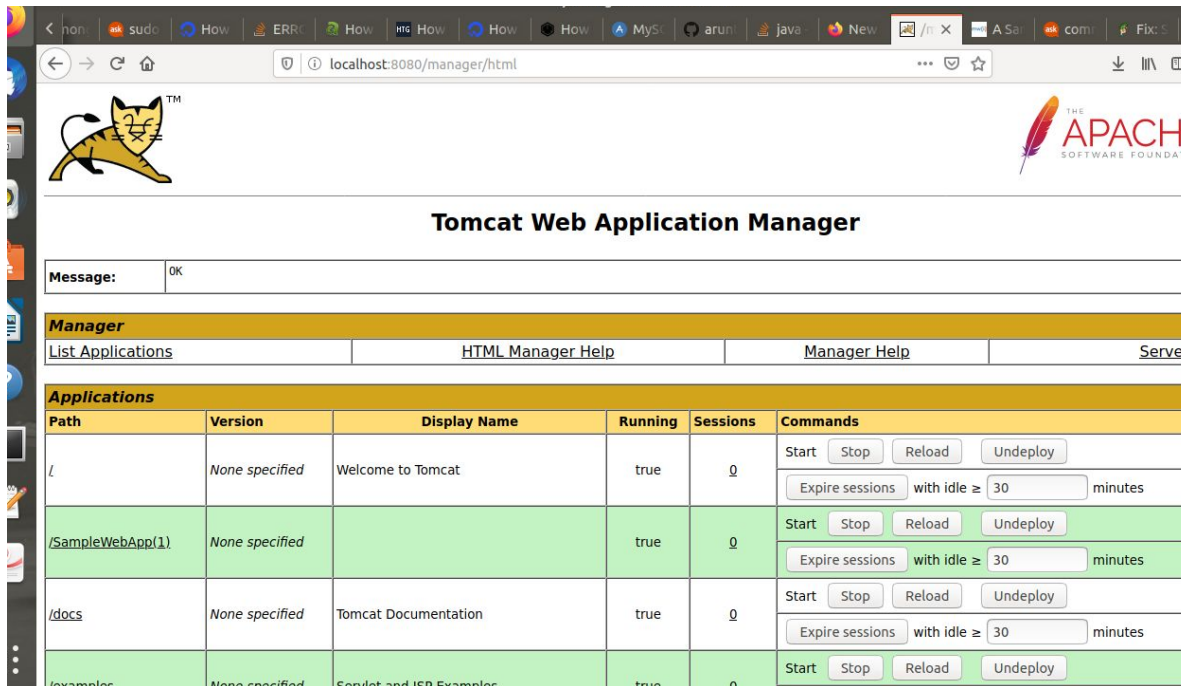
files

This is available for holding save files from the application, particularly csv files created from database tables.

4. Connect any sample.war to MySQL running on localhost.

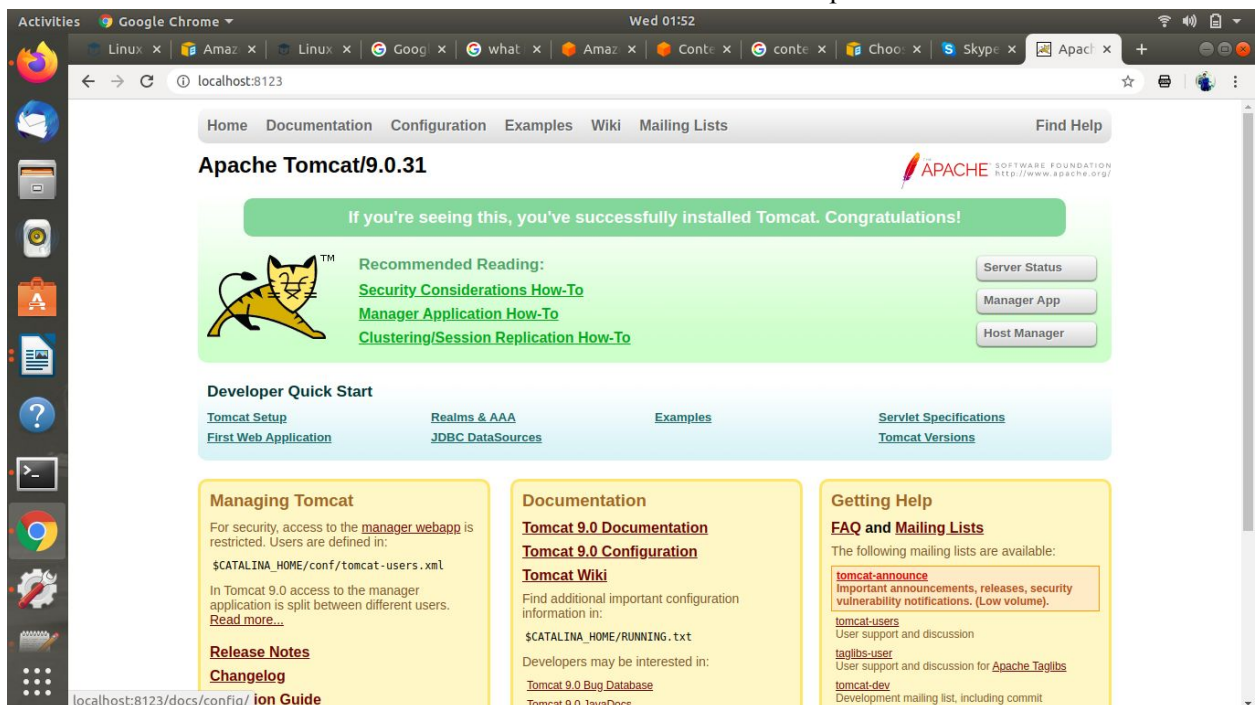
We just have to paste the sample war folder in webapps folder.

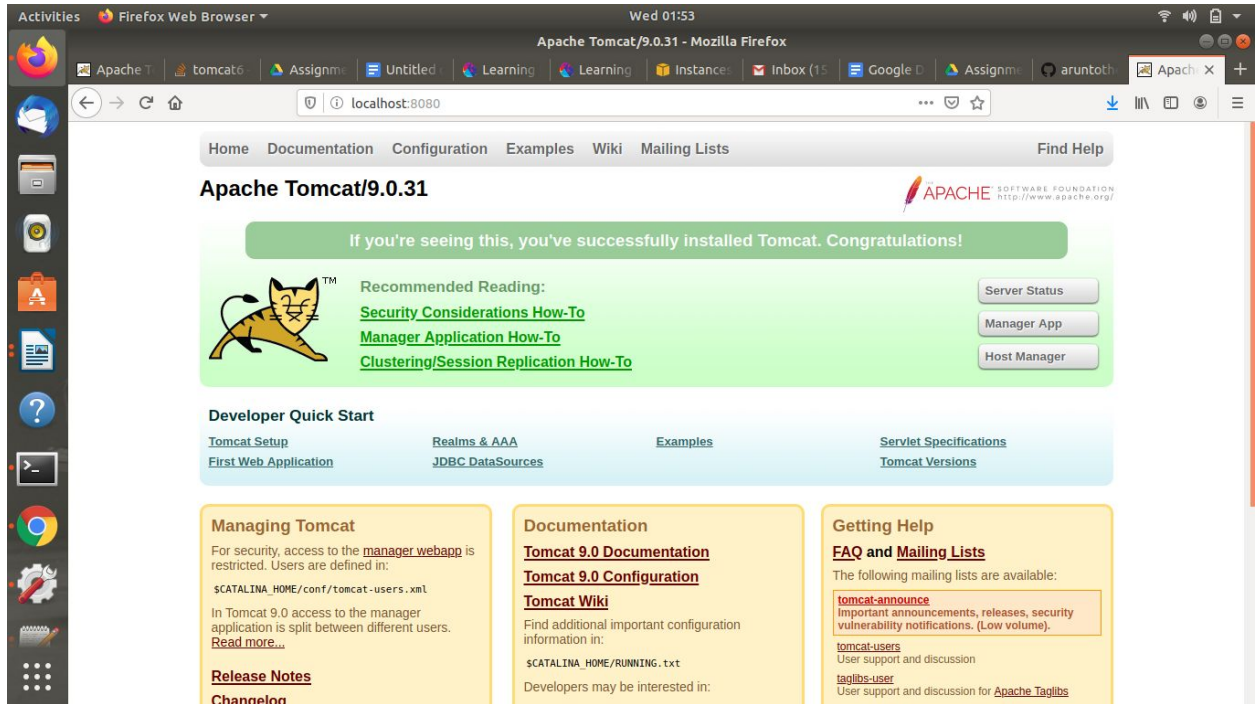
```
root@Arun-Parmar:/opt/tomcat/webapps# ^C
root@Arun-Parmar:/opt/tomcat/webapps# mv /home/arun/Downloads/SampleWebApp(1) /opt/tomcat/webapps/
bash: syntax error near unexpected token `('
root@Arun-Parmar:/opt/tomcat/webapps# mv /home/arun/Downloads/'SampleWebApp(1)'/ /opt/tomcat/webapps/
root@Arun-Parmar:/opt/tomcat/webapps#
```



5. Run multiple services on different ports with different connectors (AJP/HTTP) on same tomcat installation.

We create different service blocks for different services with different ports.





6. How to change default port in tomcat?
We can change it in server.xml file in conf folder.
7. Install tomcat and Use nginx as reverse proxy for tomcat application.

```

Activities  Terminal  Thu 13:27
root@Arun-Parmar: /opt/tomcat/conf

File Edit View Search Terminal Help
Warning: curl to output it to your terminal anyway, or consider --output
Warning: <FILE>" to save to a file.
arun@Arun-Parmar:tmp $ wget http://mirrors.estointernet.in/apache/tomcat/tomcat-9/v9.0.31/bin/apache-tomcat-9.0.31.tar.gz
--2020-02-20 10:37:40-- http://mirrors.estointernet.in/apache/tomcat/tomcat-9/v9.0.31/bin/apache-tomcat-9.0.31.tar.gz
Resolving mirrors.estointernet.in (mirrors.estointernet.in)... 103.97.84.254, 2403:8940:2::f
Connecting to mirrors.estointernet.in (mirrors.estointernet.in)|103.97.84.254|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11042076 (11M) [application/octet-stream]
Saving to: 'apache-tomcat-9.0.31.tar.gz'

apache-tomcat-9.0.31.tar.gz      100%[=====] 10.53M  1014KB/s  in 13s
2020-02-20 10:37:53 (855 KB/s) - 'apache-tomcat-9.0.31.tar.gz' saved [11042076/11042076]

```



```
apache-tomcat-9.0.31/bin/version.sh
arun@Arun-Parmar:tmp $ cd /opt/tomcat
arun@Arun-Parmar:tomcat $ sudo chgrp -R tomcat /opt/tomcat
arun@Arun-Parmar:tomcat $ sudo chmod -R g+r conf
arun@Arun-Parmar:tomcat $ sudo chmod g+x conf
arun@Arun-Parmar:tomcat $ sudo chown -R tomcat webapps/ work/ temp/ logs/
arun@Arun-Parmar:tomcat $ sudo update-java-alternatives -l
java-1.11.0-openjdk-amd64 1111 /usr/lib/jvm/java-1.11.0-openjdk-amd64
arun@Arun-Parmar:tomcat $ sudo nano /etc/systemd/system/tomcat.service
arun@Arun-Parmar:tomcat $ sudo vi /etc/systemd/system/tomcat.service
arun@Arun-Parmar:tomcat $ sudo systemctl daemon-reload
arun@Arun-Parmar:tomcat $ sudo systemctl start tomcat
arun@Arun-Parmar:tomcat $ sudo systemctl status tomcat
● tomcat.service - Apache Tomcat Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-02-20 10:42:14 IST; 8s ago
     Process: 28965 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)
    Main PID: 28972 (java)
       Tasks: 30 (limit: 4915)
      CGroup: /system.slice/tomcat.service
              └─28972 /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -Djava.util.logging.config.file=/opt/tomcat/co

Feb 20 10:42:14 Arun-Parmar systemd[1]: Starting Apache Tomcat Web Application Container...
Feb 20 10:42:14 Arun-Parmar startup.sh[28965]: Tomcat started.
Feb 20 10:42:14 Arun-Parmar systemd[1]: Started Apache Tomcat Web Application Container.

arun@Arun-Parmar:tomcat $ sudo ufw allow 8080
```

```
# Default server configuration#
server {
    listen 80;
    listen [::]:80;

    root /var/www/html/arun.com/html;
    index index.html index.htm index.nginx-debian.html;

    server_name arun.com www.arun.com;

    location / {
        proxy_pass http://localhost:8080;
    }
}

# pass PHP scripts to FastCGI server
#
```

Apache Tomcat/9.0.31



If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:

[Security Considerations How-To](#)

[Manager Application How-To](#)

[Clustering/Session Replication How-To](#)

Server Status

Manager App

Host Manager

Developer Quick Start

[Tomcat Setup](#)

[First Web Application](#)

[Realms & AAA](#)

[JDBC DataSources](#)

[Examples](#)

[Servlet Specifications](#)

[Tomcat Versions](#)

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in:

`$CATALINA_HOME/conf/tomcat-users.xml`

In Tomcat 9.0 access to the manager application is split between different users. [Read more...](#)

[Release Notes](#)

[Changelog](#)

[Migration Guide](#)

Documentation

[Tomcat 9.0 Documentation](#)

[Tomcat 9.0 Configuration](#)

[Tomcat Wiki](#)

Find additional important configuration information in:

`$CATALINA_HOME/RUNNING.txt`

Developers may be interested in:

[Tomcat 9.0 Bug Database](#)

Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

[tomcat-announce](#)
Important announcements, releases, security vulnerability notifications. (Low volume).

[tomcat-users](#)
User support and discussion

[taglibs-user](#)
User support and discussion for [Apache Taglibs](#)

[tomcat-dev](#)
Development mailing list, including commit