

Yugandhar

Open Master Data Management (MDM) Hub

Development Environment Setup Guide

Yugandhar Open MDM Hub Release - V1.0.0

Date – 27/12/2017

+++++

Copyright [2017] [Yugandhar Open MDM Hub]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

Contents

1. System Requirements	3
2. Software Download links	6
i. Eclipse	6
ii. Red Hat JBoss Enterprise Application Platform	6
iii. Apache maven.....	6
iv. Oracle Database	6
v. Java JDK	6
vi. Hibernate Tools.....	7
vii. Oracle SQL Developer	7
3. Setup Database	8
Table spaces –	8
Profile	8
USER	8
4. Setup Workspace	9
1. Setup up Network Connections	11
Eclipse network	11
Maven Settings	11
2. Set JDK Path	14
3. Configure RedHat JBOSS EAP	18
a. Add User to Access JBoss Management Console.....	24
b. Create JBoss Datasource and Active MQ Queues.....	26
a. Verify JBoss Configuration	28
• Verify XA Datasource	29
• Verify Active MQ Queues.....	30
4. Import Yugandhar Projects in the workspace.....	32
5. Deploy Yugandhar Boot Project To JBoss Server	35
6. TEST With SOAPUI.....	37

About Yugandhar Open MDM Hub Project

Master Data Management came a long way in last decade or so. There are currently more than 20 MDM solutions catering to various specializations of MDM like Customer Data Integration (CDI), Product Information Management (PIM), vendor and supplier management etc. However most of these solutions come with licensing costs amounting to thousands of dollar. To offer a completely free solution which would be made available through Apache 2.0 license, A Project is started in 2017 under the name 'Yugandhar Open MDM Project' to build Open Source MDM solutions catering to CDI, PIM and Data Governance Capabilities. Yugandhar in Sanskrit means Ever Lasting and the strongest of its time. Our vision is to build the strongest, Open Source, Multi Domain, Cross Industry and completely free MDM Solution.

We are happy to announce that the first release of the Yugandhar MDM Hub catering to CDI solution is built with Open source technologies like Spring and Hibernate etc, inbuilt data Model, 400+ ready to use services and having incredible Out of the Box capabilities is currently being distributed. We aim to make the current CDI offering the strongest and Planning to bring Data Stewardship and PIM solutions in upcoming years.

About this document

This document covers the system requirements for Yugandhar Open MDM Hub.

1. System Requirements

Below are the System Requirements for setting up Development Environment

1. OS – Windows 7 enterprise edition, Service Pack 1 or later
2. 8GB RAM and 100 GB Storage
3. Eclipse Java EE IDE for Web Developers. Neon.3 Release (4.6.3) or later
4. JBoss EAP 7.1 full runtime or later
5. apache-maven-3.5.0 or later
6. Java jdk 1.8 (jdk1.8.0_121) or later
7. Spring Boot 1.5.6.RELEASE
8. Hibernate Tools
9. Oracle Database 11g Express Edition Release 11.2.0.2.0
10. Oracle SQL Developer
11. SOAPUI or any other tool to test REST services

Note - Internet connectivity is needed to setup workspace. If internet is not available because of any reason then all the software and Maven jars needs to be manually downloaded which might be tedious task.

IMPORTANT NOTICE - Please read the licensing terms before downloading the software/s. Yugandhar team would not be responsible for licensing violations if any.

2. Software Download links

i. Eclipse

Eclipse Neon 3 download page - Eclipse IDE for Java EE Developers

Version - Neon.3 Release (4.6.3) or later

Download page - <http://www.eclipse.org/downloads/packages/release/Neon/3>

ii. Red Hat JBoss Enterprise Application Platform

Red Hat JBoss Enterprise Application Platform

Version - JBoss EAP 7.1 full runtime or later

Download page - <https://developers.redhat.com/products/eap/download/>

iii. Apache maven

Apache Maven comes integrated with Eclipse Neon but if you want to install standalone maven then you may download it from below path

Version - apache-maven-3.5.0 or later

Download Page - <https://maven.apache.org/download.cgi>

iv. Oracle Database

Download Oracle from Oracle download site

Version - Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production OR Oracle Database 11g Express Edition Release 11.2.0.2.0

Note - Oracle Database 11g Express Edition is specifically meant for testing purpose so won't be useful on production.

Download link - <http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>

v. Java JDK

Download Java jdk 1.8 (jdk1.8.0_121) or later from below link

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

vi. Hibernate Tools

You may choose to download the Hibernate Tools (Now renamed as JBoss Tools) from the link below or install to install the plugin from eclipse market place using eclipse Menu → Help > Eclipse Marketplace... option.

<http://tools.jboss.org/downloads/jbosstools/oxygen/4.5.1.Final.html#marketplace>

vii. Oracle SQL Developer

Download SQL developer to connect to database

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

3. Setup Database

Install the Oracle 11g XE database using the instructions mentioned below.

https://docs.oracle.com/cd/E11882_01/nav/portal_11.htm

Also Install the Oracle SQL Developer. The step by step installation instructions for installing the Oracle 11g XE database and Oracle SQL Developer is out of scope of this document.

By Default Yugandhar Open MDM Hub uses the schema MDM_OWNER. If you wish to change the user name (schema name) then you need to modify all the scripts to change the schema name.

Download the scripts from github repository resources\dbsetupscripts location and execute the below scripts in sequence

Github repository link - <https://github.com/yugandharproject/YugandharMDMHub>

Download the sqls from resources\dbsetupscripts path of the YugandharMDMHub

1. CreateTablespaces.sql – Needs DBA access
2. FullSchema.sql – Can be executed with MDM_OWNER user access
3. CreateSequence.sql – Can be executed with MDM_OWNER user access
4. LoadTableDataWrapper.sql – Can be executed with MDM_OWNER user access
5. DataImport Folder – download all the scripts present in the folder

Verify the logs and check that all the steps are executed correctly and REF_xxx as well as CONFIG_xxx tables are loaded with sample data. In Summary, below mentioned objects to the TABLES, Sequence and INDEXES are created in database

Table spaces –

MDM_DATATS – used for data and reference tables

MDM_INDXTS – used for Indexes

Profile –

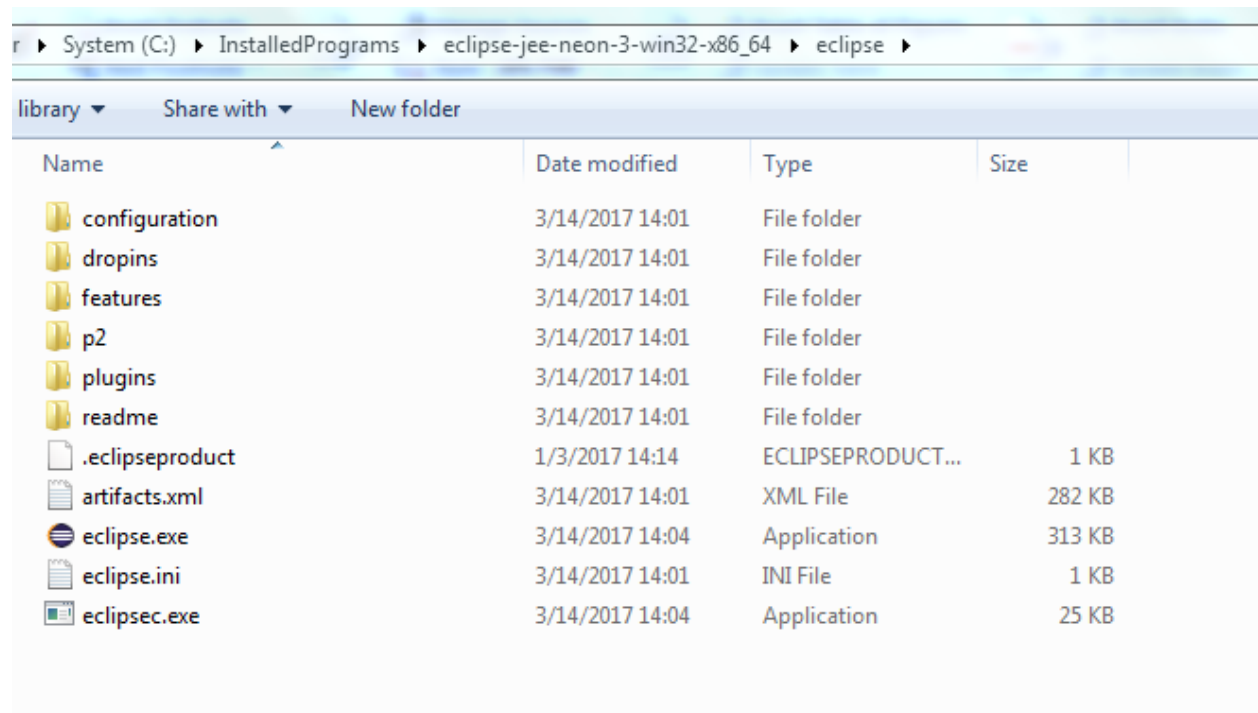
MDM_PROFILE - Used to create MDM_OWNER user












USER –

MDM_OWNER – Default user Schema used by Yugandhar Open MDM Hub.

4. Setup Workspace

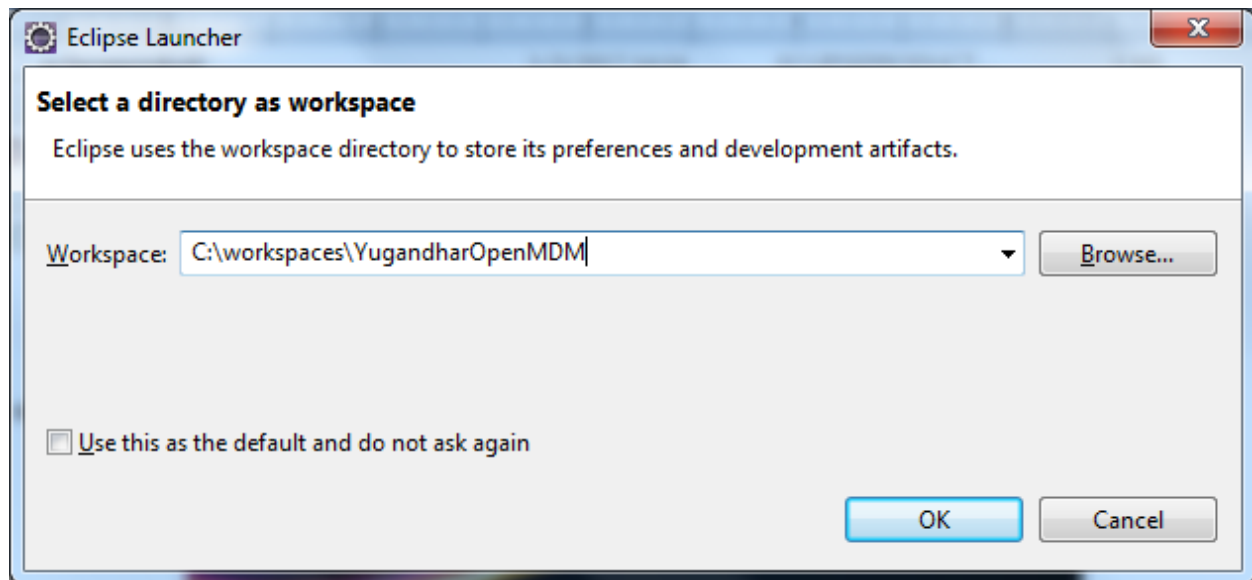
Extract the downloaded eclipse archive in any of the folder of your choice.



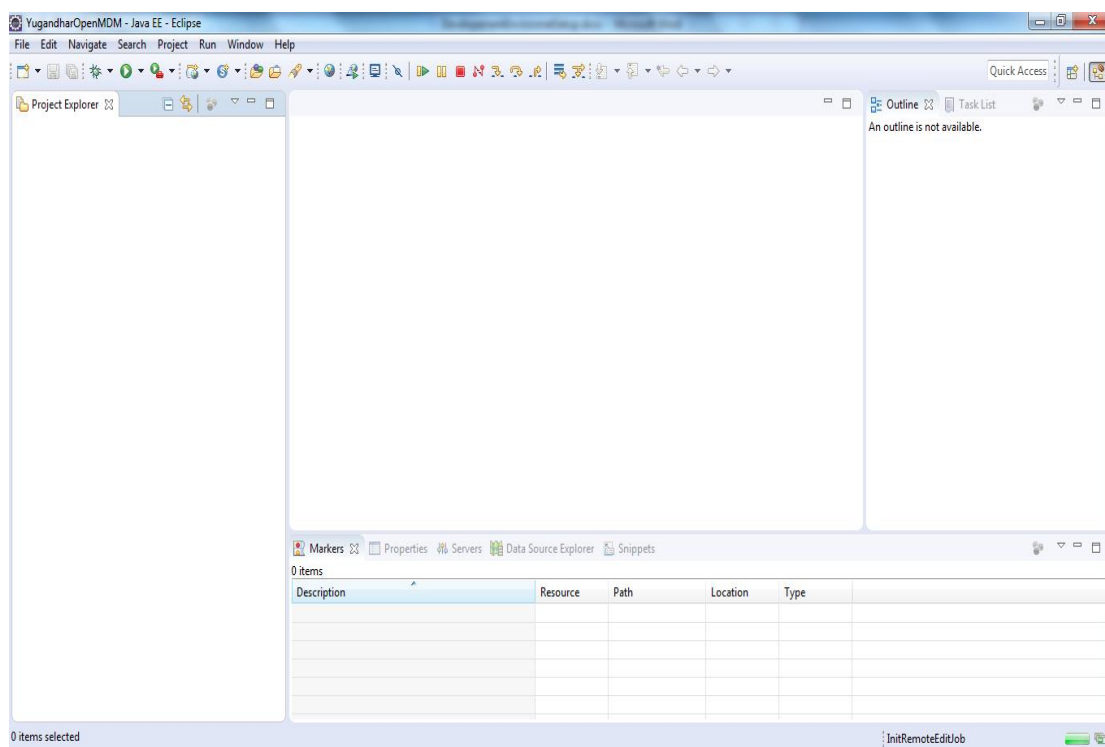
library ▾ Share with ▾ New folder				
Name	Date modified	Type	Size	
 configuration	3/14/2017 14:01	File folder		
 dropins	3/14/2017 14:01	File folder		
 features	3/14/2017 14:01	File folder		
 p2	3/14/2017 14:01	File folder		
 plugins	3/14/2017 14:01	File folder		
 readme	3/14/2017 14:01	File folder		
 .eclipseproduct	1/3/2017 14:14	ECLIPSEPRODUCT...	1 KB	
 artifacts.xml	3/14/2017 14:01	XML File	282 KB	
 eclipse.exe	3/14/2017 14:04	Application	313 KB	
 eclipse.ini	3/14/2017 14:01	INI File	1 KB	
 eclipsec.exe	3/14/2017 14:04	Application	25 KB	

The extracted folder would have above files, click on eclipse.exe to start the eclipse IDE.

The next window would ask you for the workspace location, provide the location as per your choice, for this documentation purpose we are creating workspace in C:\workspaces\YugandharOpenMDM folder.



Your workspace is ready to be configured now.



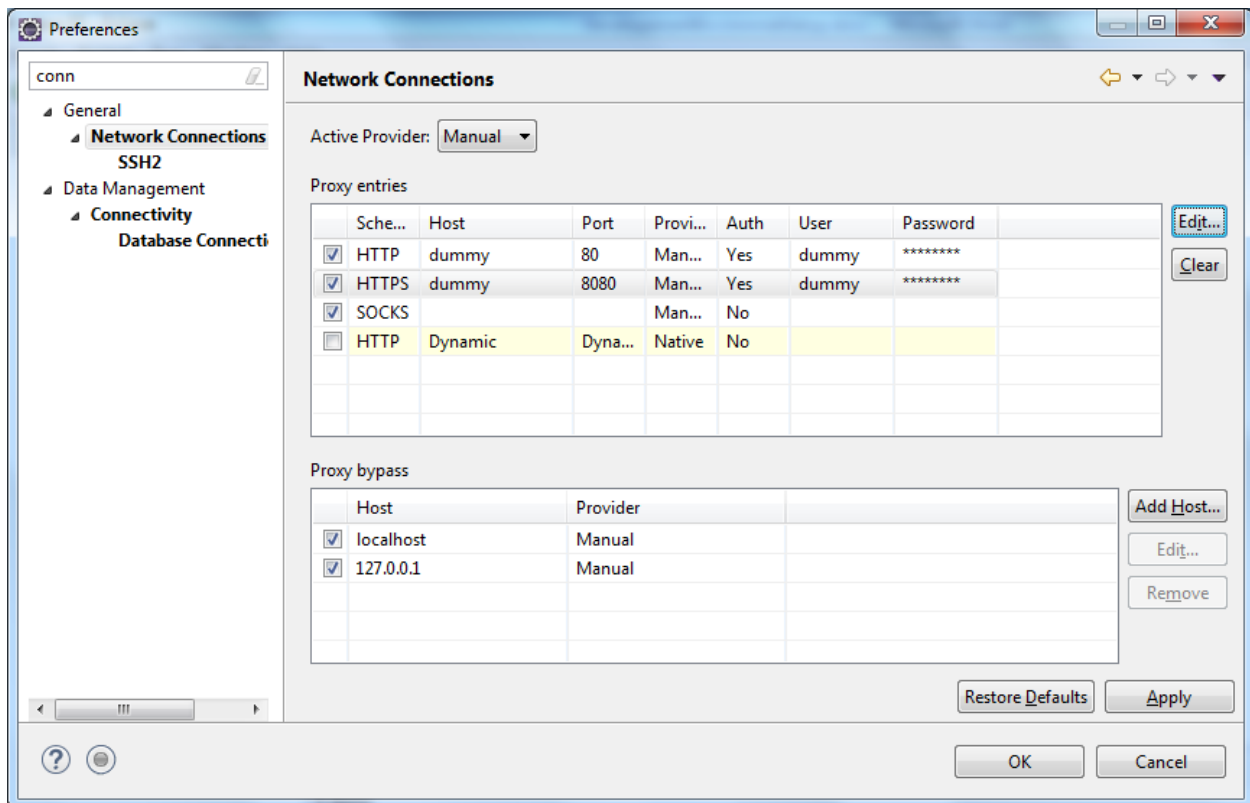
1. Setup up Network Connections

If you are behind firewall then you need to perform below steps. Those who have direct internet connection (without proxy) can skip the eclipse Network and Maven Settings step.

Eclipse network

Go to Windows -- > preferences → General → Network Connections

Change the Active Provider to manual and set the proxy host, port and user credentials as per your firewall settings. The below settings are dummy so should not be copied as-is. Click on apply and click OK.



Maven Settings

Create a simple text file named MavenSettings.xml in workspace having below content.

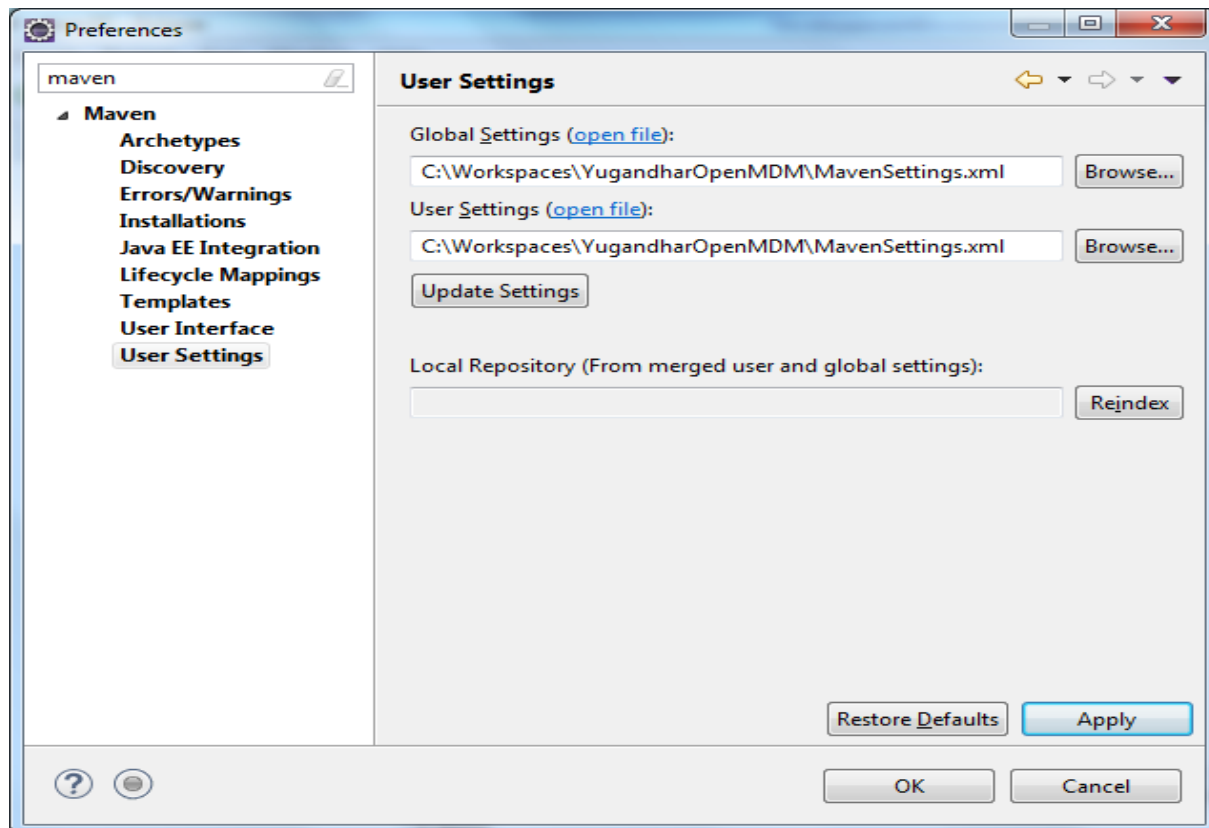
You may also download sample MavenSettings.xml from git hub resources/mavensettings folder.

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
                    http://maven.apache.org/xsd/settings-1.0.0.xsd">
<localRepository/>
<interactiveMode/>
<usePluginRegistry/>
<offline/>
<pluginGroups/>
<servers/>
<mirrors/>
<proxies>
  <proxy>
    <id>myproxy</id>
    <active>true</active>
    <protocol>http</protocol>
    <host>dummy</host>
    <port>8080</port>
    <username>dummy</username>
    <password>dummy</password>
    <nonProxyHosts>localhost,127.0.0.1</nonProxyHosts>
  </proxy>
</proxies>
<profiles/>
<activeProfiles/>
</settings>
```

Note – Make sure to change the host, port, username and password as per your firewall settings.

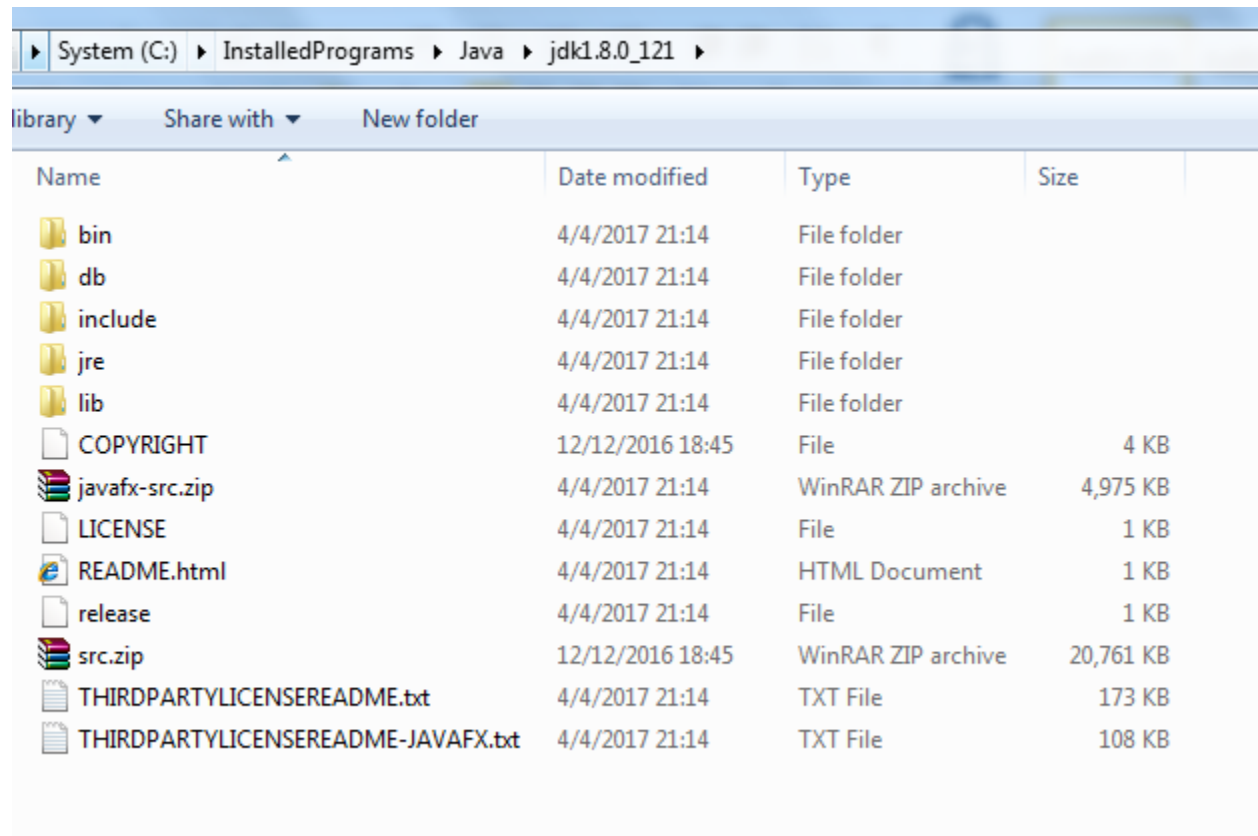
Now go to Eclipse Menu → windows → preferences → maven → User Settings → provide path of the file MavenSettings.xml in the local and global settings as shown in screenshot below



2. Set JDK Path

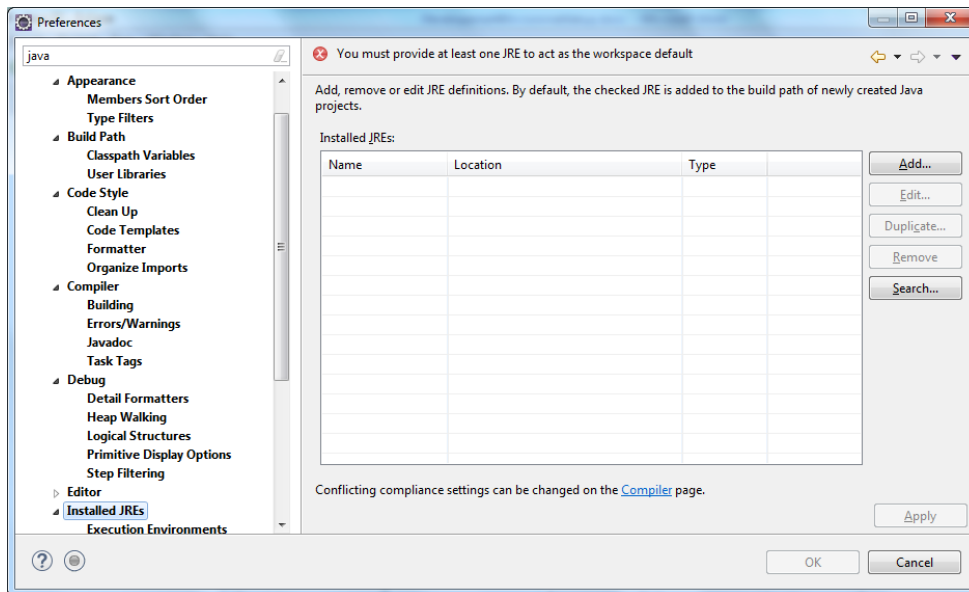
Extract the JDK in a folder of your choice, for the document purpose the jdk directory is

C:\InstalledPrograms\Java\jdk1.8.0_121. The directory should look something like below

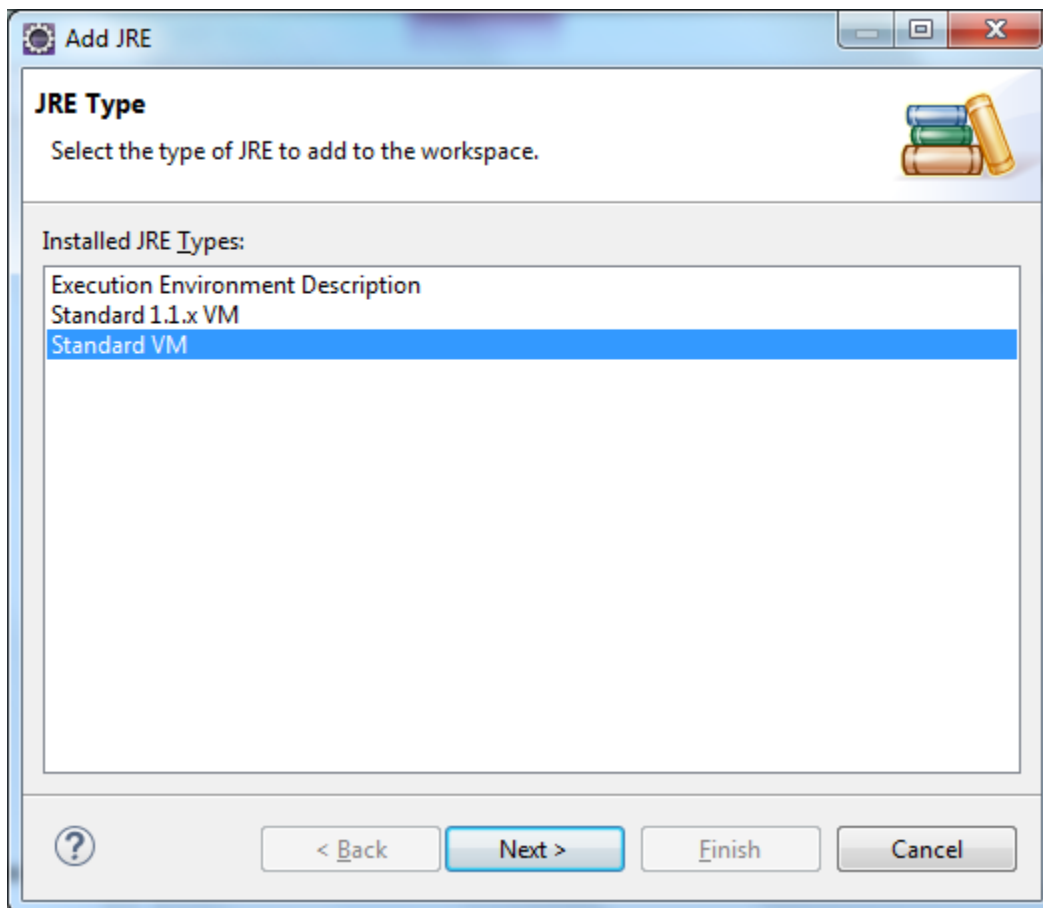


System (C:) > InstalledPrograms > Java > jdk1.8.0_121				
library ▾ Share with ▾ New folder				
Name	Date modified	Type	Size	
bin	4/4/2017 21:14	File folder		
db	4/4/2017 21:14	File folder		
include	4/4/2017 21:14	File folder		
jre	4/4/2017 21:14	File folder		
lib	4/4/2017 21:14	File folder		
COPYRIGHT	12/12/2016 18:45	File	4 KB	
javafx-src.zip	4/4/2017 21:14	WinRAR ZIP archive	4,975 KB	
LICENSE	4/4/2017 21:14	File	1 KB	
README.html	4/4/2017 21:14	HTML Document	1 KB	
release	4/4/2017 21:14	File	1 KB	
src.zip	12/12/2016 18:45	WinRAR ZIP archive	20,761 KB	
THIRDPARTYLICENSEREADME.txt	4/4/2017 21:14	TXT File	173 KB	
THIRDPARTYLICENSEREADME-JAVAFX.txt	4/4/2017 21:14	TXT File	108 KB	

Go to eclipse Menu → Windows → Preferences → Java → Installed JREs

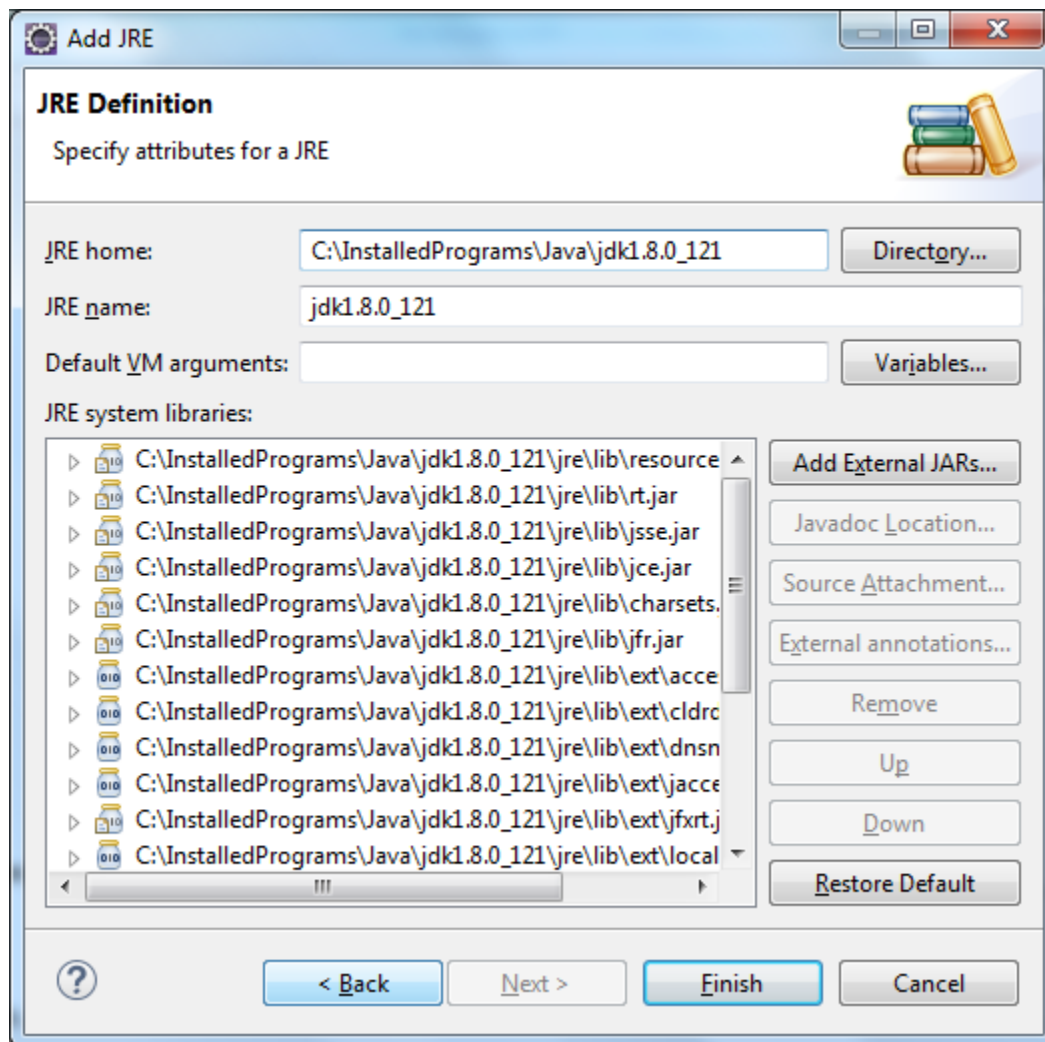


Click Add

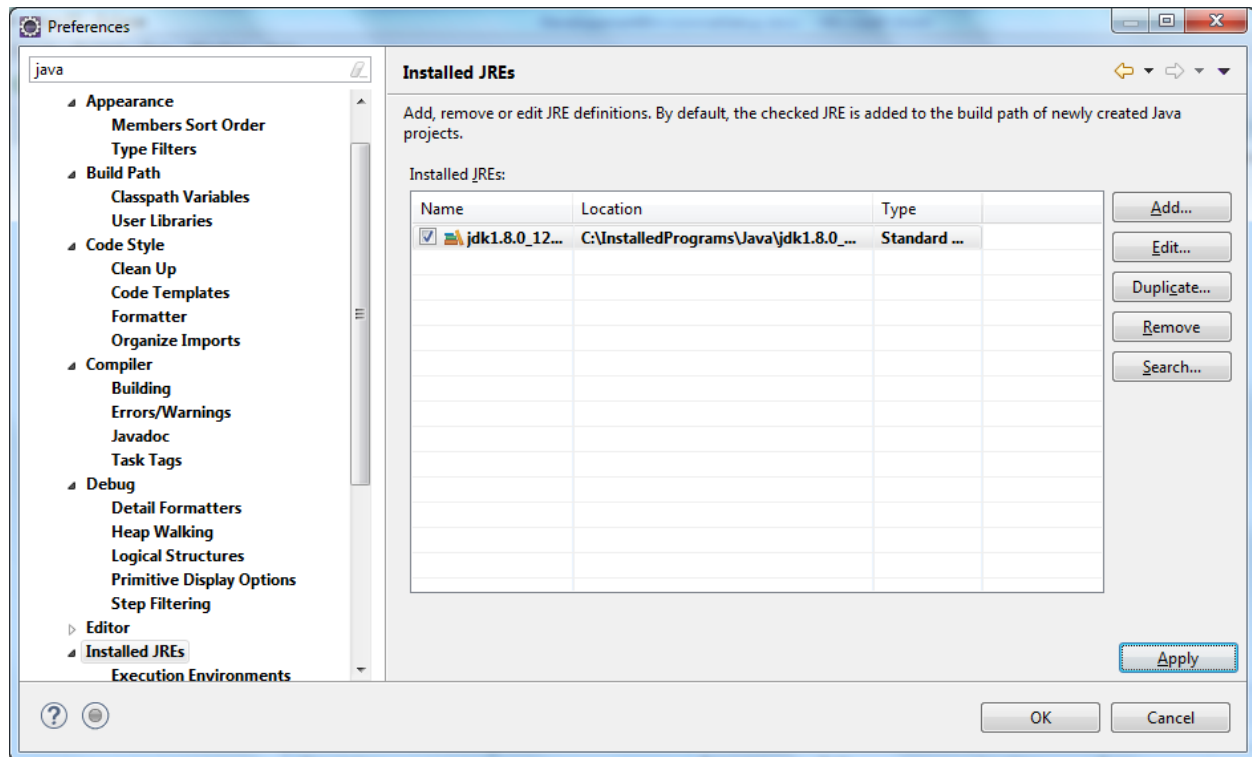


Click Next

Provide the JDK folder name where JDK is extracted



Click finish

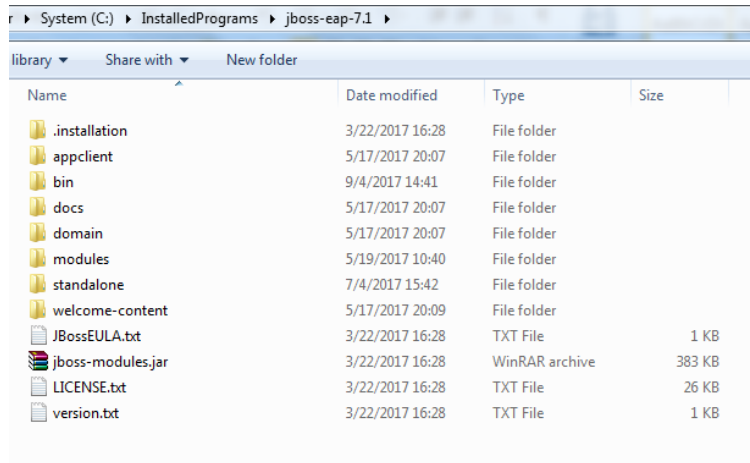


Check the textbox against recently added JDK, click Apply and OK.

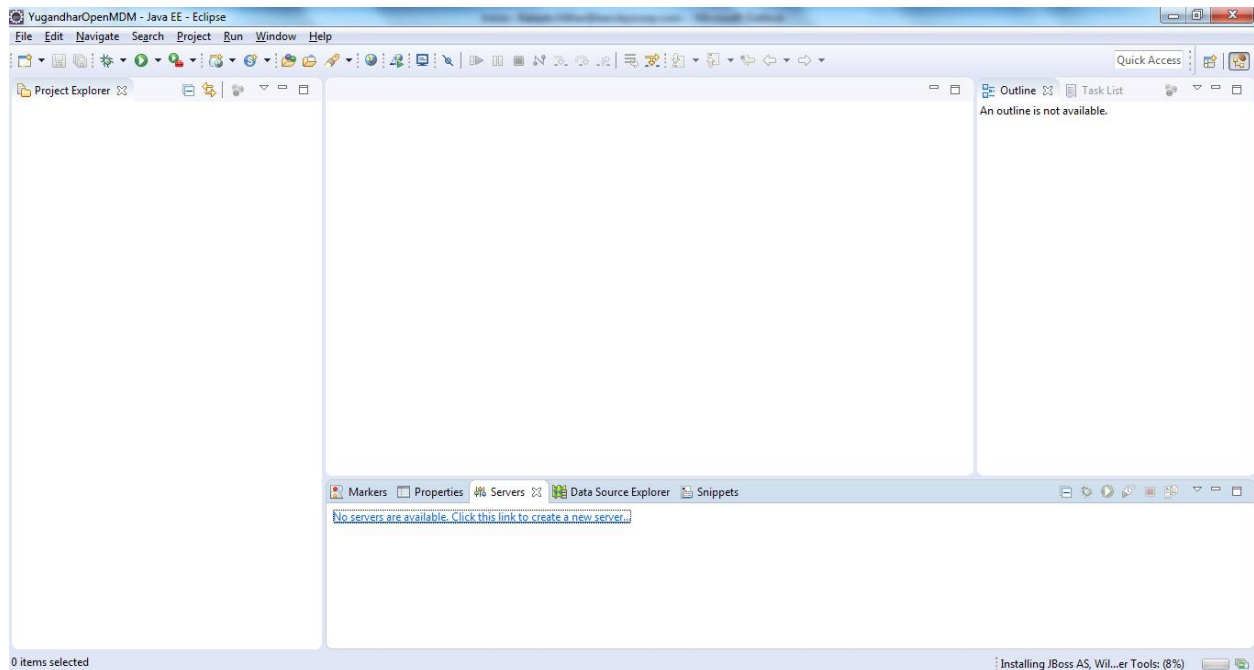
3. Configure RedHat JBOSS EAP

Extract the downloaded RedHat Server to a directory of your choice like

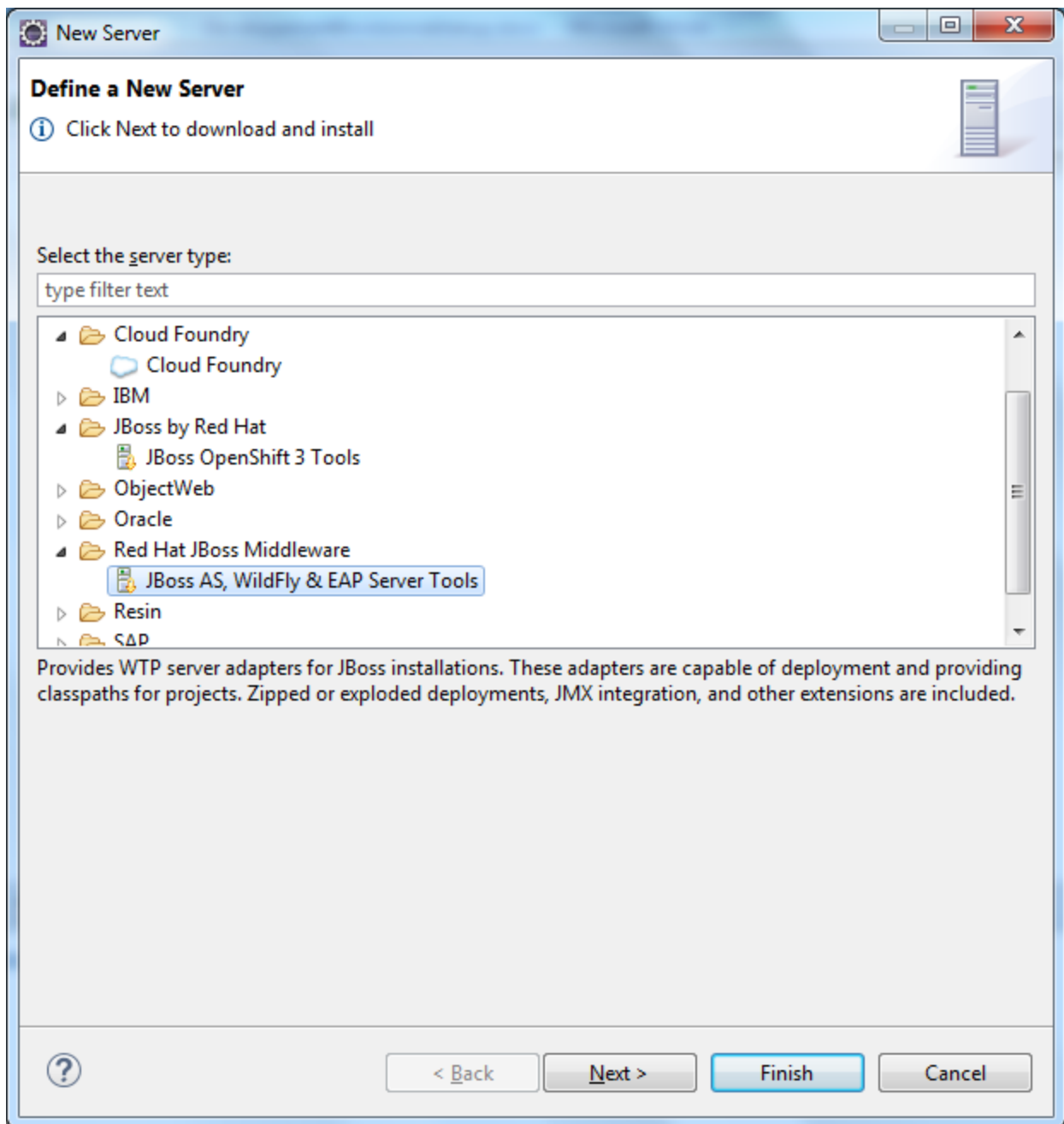
C:\InstalledPrograms\jboss-eap-7.1



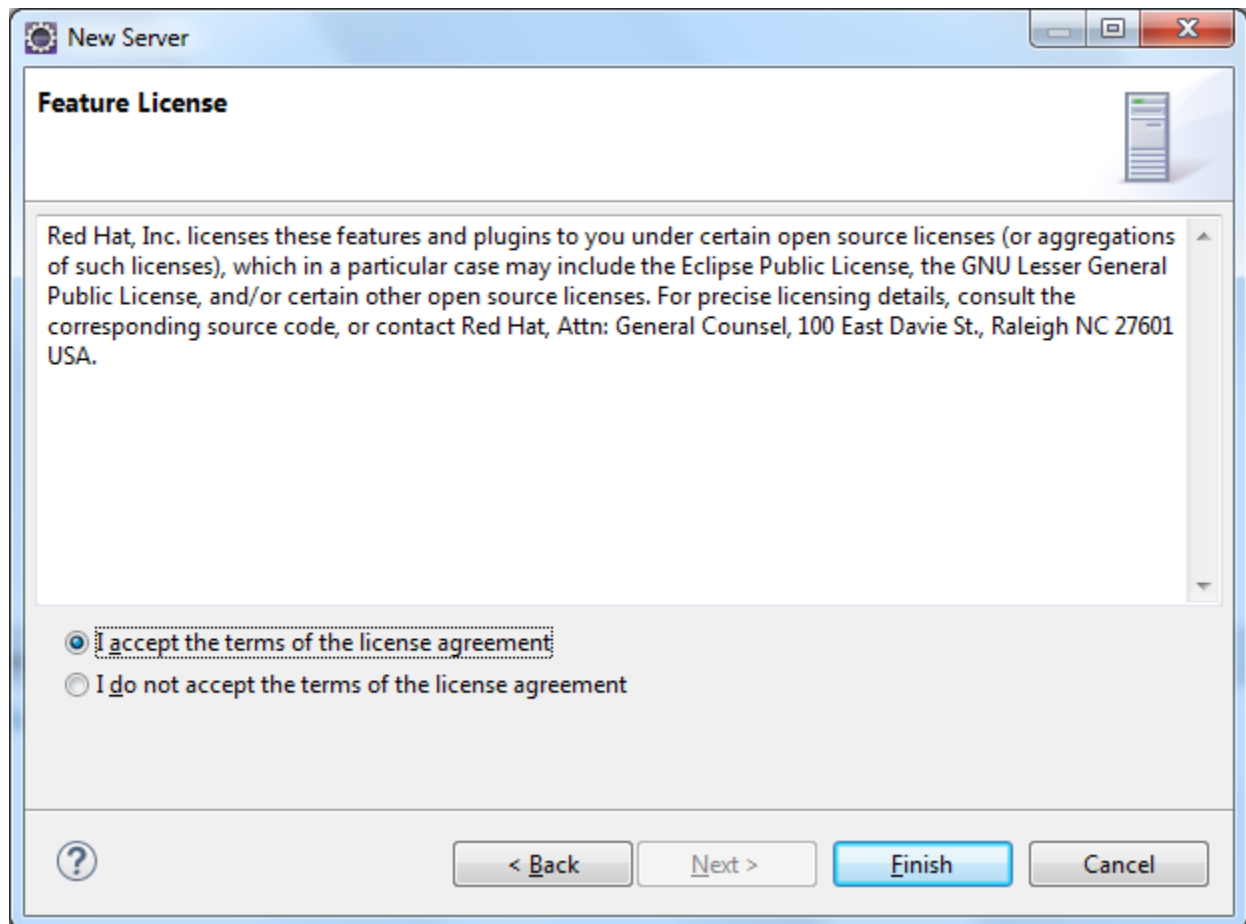
Go to eclipse 'Servers' tab and click on the link to create new Server.



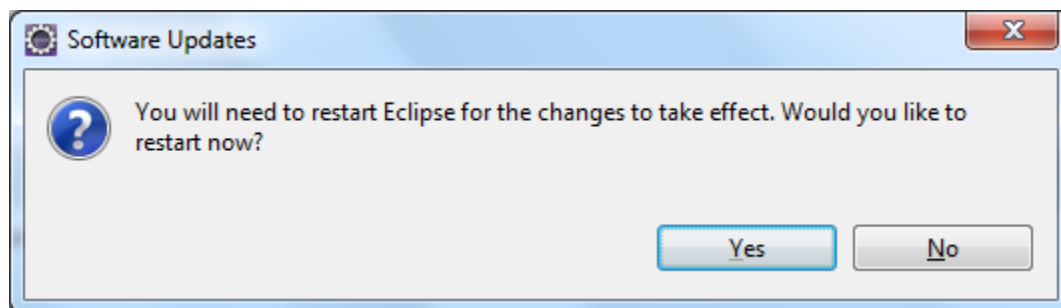
Click the Red Hat Jboss Middleware option as shown in below screenshot



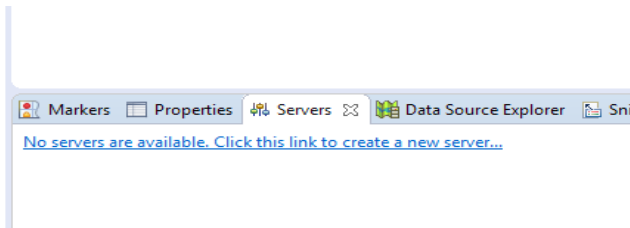
Click Next. It may take some time to process after which will ask to accept the licensing terms. Accept the license and click Finish.



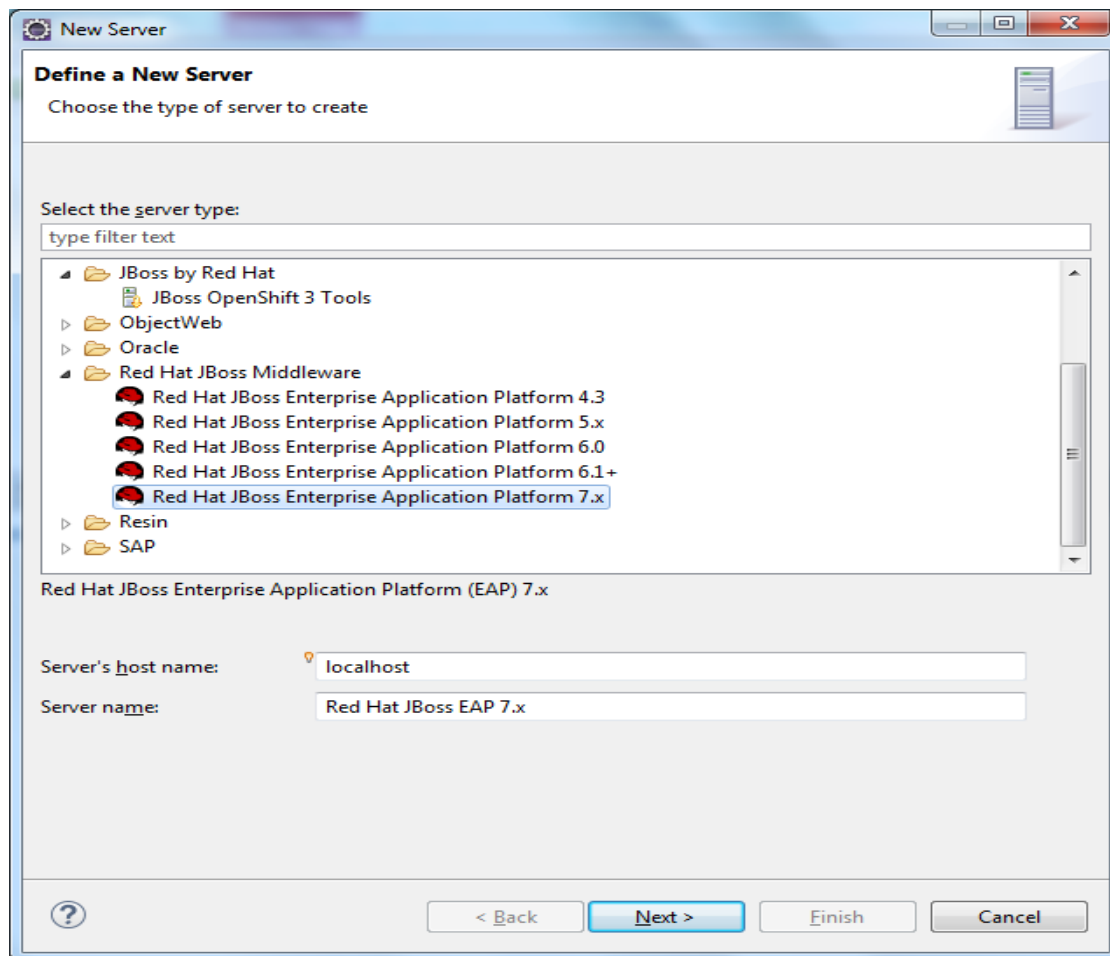
Eclipse will automatically download and install the JBoss Tools which may take some time based on the speed of your internet connection. The eclipse IDE needs to be restarted after this step so that JBoss EAP tools are available in eclipse.



Click on the Server Tab and Create new server link



You would now see Red Hat JBoss Enterprise Application Platform 7.x as an option now. Click on the same and click Next



Click Next

New Server

Create a new Server Adapter

Red Hat JBoss Enterprise Application Platform (EAP) 7.x

A Server Adapter manages starting and stopping instances of your server. It manages command line arguments and keeps track of which modules have been deployed.

The server is:

- ☒ Local
- ☐ Remote

Controlled by:

- ☒ Filesystem and shell operations
- ☐ Management Operations

☐ Server lifecycle is externally managed.

The selected profile requires a runtime.

☒ Assign a runtime to this server

Create new runtime (next page) ▼

Runtime Details

JRE:

Home Directory:

Base Directory:

Configuration File:

? < Back Next > Finish Cancel

Provide the directory where we have extracted Jboss archive. Also provide the JRE we added in previous steps. Also choose the Configuration file as standalone-full-ha.xml.

New Server

JBoss Runtime
Red Hat JBoss Enterprise Application Platform (EAP) 7.x

A JBoss Server runtime references a JBoss installation directory. It can be used to set up classpaths for projects which depend on this runtime, as well as by a "server" which will be able to start and stop instances of JBoss.

Name
JBoss EAP 7.x Runtime

Home Directory
C:\InstalledPrograms\jboss-eap-7.1 [Download and install runtime...](#) [Browse...](#)

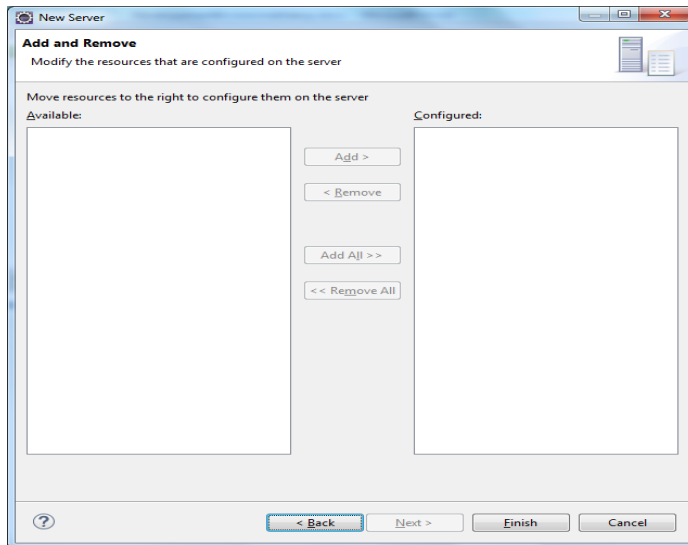
Runtime JRE
☐ Execution Environment: JavaSE-1.8 [Environments...](#)
☒ Alternate JRE: jdk1.8.0_121 [Installed JREs...](#)

Server base directory: standalone [Browse...](#)

Configuration file: standalone-full-ha.xml [Browse...](#)

[?<](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

Click Next and Finish



You would see the added server in the Servers Tab now. You may click on the start button and check that server is started without errors. Check the logs in Console View and after verification stop the server.

a. Add User to Access JBoss Management Console

Go to directory where jboss is installed e.g. C:\InstalledPrograms\jboss-eap-7.1.0\bin

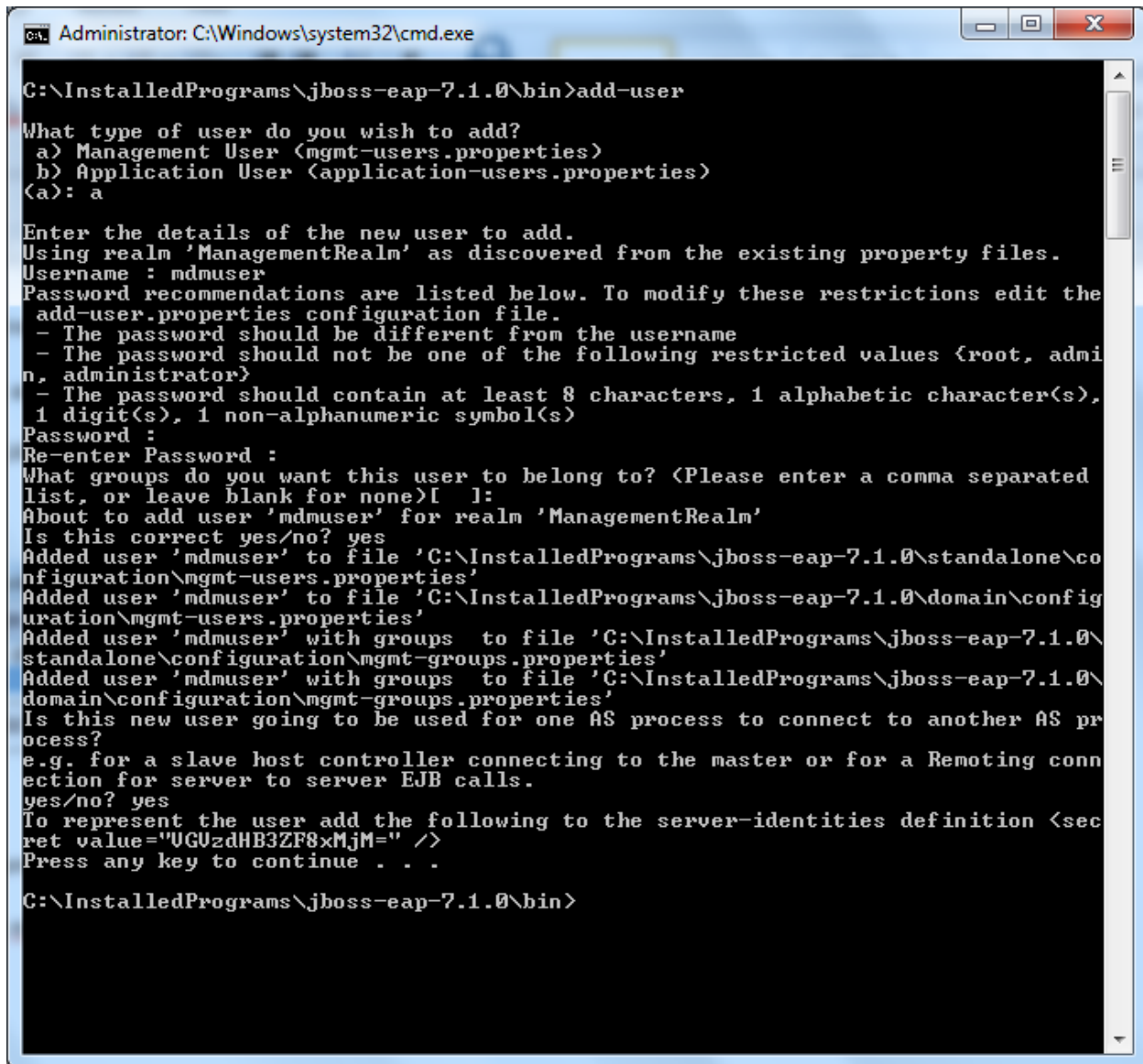
Edit the add-user.bat file to set java home, add the jdk home path where you extracted the JDK

e.g. set JAVA_HOME=C:\InstalledPrograms\Java\jdk1.8.0_121

```
add-user.bat - Notepad
File Edit Format View Help
@echo off
set JAVA_HOME=C:\InstalledPrograms\Java\jdk1.8.0_121
rem -----
rem Add User script for windows
rem -----
```

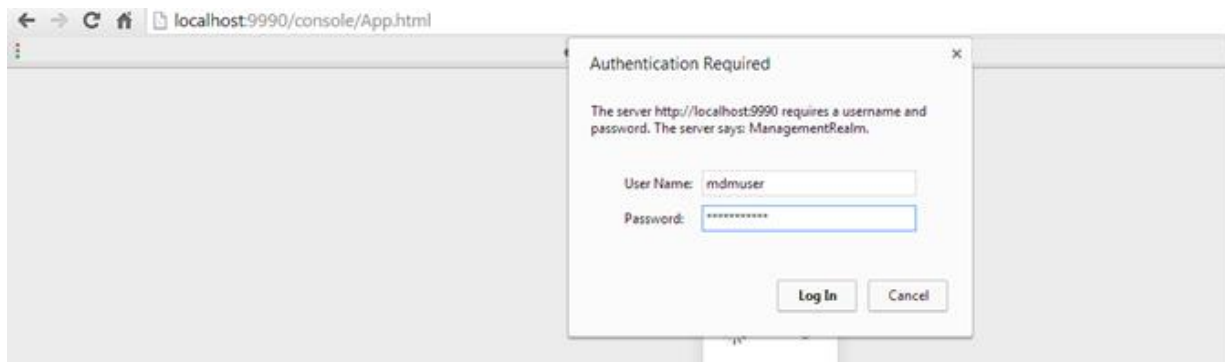
Go to command prompt and add the user by running add-user.bat file

Provide mdmuser / Testpwd_123 as username and password (Password may be different of your choice)



```
C:\Windows\system32\cmd.exe
C:\InstalledPrograms\jboss-eap-7.1.0\bin>add-user
What type of user do you wish to add?
a) Management User <mgmt-users.properties>
b) Application User <application-users.properties>
(a): a
Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : mdmuser
Password recommendations are listed below. To modify these restrictions edit the
add-user.properties configuration file.
- The password should be different from the username
- The password should not be one of the following restricted values {root, admin, administrator}
- The password should contain at least 8 characters, 1 alphabetic character(s), 1 digit(s), 1 non-alphanumeric symbol(s)
Password :
Re-enter Password :
What groups do you want this user to belong to? <Please enter a comma separated list, or leave blank for none>[]
About to add user 'mdmuser' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'mdmuser' to file 'C:\InstalledPrograms\jboss-eap-7.1.0\standalone\configuration\mgmt-users.properties'
Added user 'mdmuser' to file 'C:\InstalledPrograms\jboss-eap-7.1.0\domain\configuration\mgmt-users.properties'
Added user 'mdmuser' with groups to file 'C:\InstalledPrograms\jboss-eap-7.1.0\standalone\configuration\mgmt-groups.properties'
Added user 'mdmuser' with groups to file 'C:\InstalledPrograms\jboss-eap-7.1.0\domain\configuration\mgmt-groups.properties'
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.
yes/no? yes
To represent the user add the following to the server-identities definition <secret value="UGVzdHB3ZF8xMjM=" />
Press any key to continue . . .
C:\InstalledPrograms\jboss-eap-7.1.0\bin>
```

Type the url <http://localhost:9990/> to login to console and provide username and password



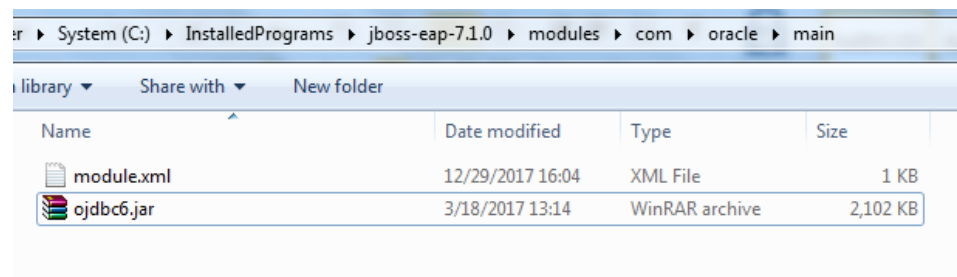
b. Create JBoss Datasource and Active MQ Queues

Yugandhar Open MDM Hub needs JBoss based datasource to connected to Oracle database. The JNDI name of the datasource is java:/XAOracleDS. To create Oracle data source in JBoss, You need to manually make some changes on the file system. Also make sure that ojdbc6.jar is downloaded on your system from oracle distribution site.

You may take help from the jboss sample configuration file provided in the <github Repository/resources\jbossconfig folder

1. Create a new module for oracle driver.

- i. Create a folder hierarchy with path `$JBOSS_HOME/modules`.
- ii. `$JBOSS_HOME/modules/com/oracle/main`
- iii. Copy `ojdbc6.jar` to `$JBOSS_HOME/modules/com/oracle/main` folder



- iv. Create `module.xml` file.
- v. Add the following content:

```
<?xml version="1.0" ?>
<module xmlns="urn:jboss:module:1.1" name="com.oracle">
  <properties>
    <property name="jboss.api" value="unsupported" />
  </properties>
  <resources>
    <resource-root path="ojdbc6.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api" />
    <module name="javax.transaction.api" />
    <module name="javax.servlet.api" optional="true" />
  </dependencies>
</module>
```

2. Configure data source

in `$JBOSS_HOME/standalone/configuration/standalone.xml`

Add the following configuration with in `<datasources>`

Note – make sure to change userid/password and other details as may be relevant

```
<xa-datasource jndi-name="java:/XAOracleDS" pool-name="XAOracleDS" enabled="true" use-ccm="false">
  <xa-datasource-property name="URL">
    jdbc:oracle:thin:@localhost:1521:xe
  </xa-datasource-property>
  <driver>oracle</driver>
  <xa-pool>
    <is-same-rm-override>false</is-same-rm-override>
    <no-tx-separate-pools>true</no-tx-separate-pools>
  </xa-pool>
  <security>
    <user-name>MDM_OWNER</user-name>
    <password>mdm_owner</password>
  </security>
  <validation>
    <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker"/>
    <background-validation>true</background-validation>
    <stale-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChecker"/>
    <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter"/>
  </validation>
</xa-datasource>
```

3. Add the drivers in drivers section

```
<driver name="oracle" module="com.oracle">
    <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-
class>
</driver>
```

4. Disable default JPA in jboss 7.1 standalone-full-ha.xml by removing below entry altogether

```
<subsystem xmlns="urn:jboss:domain:jpa:1.1">
    <jpa default-datasource="" default-extended-persistence-inheritance="DEEP"/>
</subsystem>
```

5. Create ActiveMQ server and JMS Queues

Add the below entries in Active MQ subsystem inside below tag

```
<subsystem xmlns="urn:jboss:domain:messaging-activemq:2.0">
```

```
<server name="Yug">
    <security enabled="false"/>
    <management address="jms.queue.activemq.management1"/>
    <statistics enabled="true"/>
    <security-setting name="#">
        <role name="guest" send="true" consume="true" manage="true"/>
    </security-setting>
    <address-setting name="#" dead-letter-address="jms.queue.DLQ" expiry-address="jms.queue.ExpiryQueue"/>
    <remote-connector name="yugConnectorRemote" socket-binding="http"/>
    <in-vm-connector name="yugConnectorInvm" server-id="0"/>
    <jms-queue name="YUG.DEFAULT.RESPONSE"
entries="java:jboss/com/yugandhar/default/responseQueue"/>
    <jms-queue name="YUG.DEFAULT.REQUEST" entries="java:jboss/com/yugandhar/default/requestQueue"/>
    <pooled-connection-factory name="YugandharDefaultPooledConnectionFactory"
entries="java:jboss/com/yugandhar/DefaultPooledConnectionFactory" connectors="yugConnectorInvm" statistics-
enabled="true"/>
</server>
```

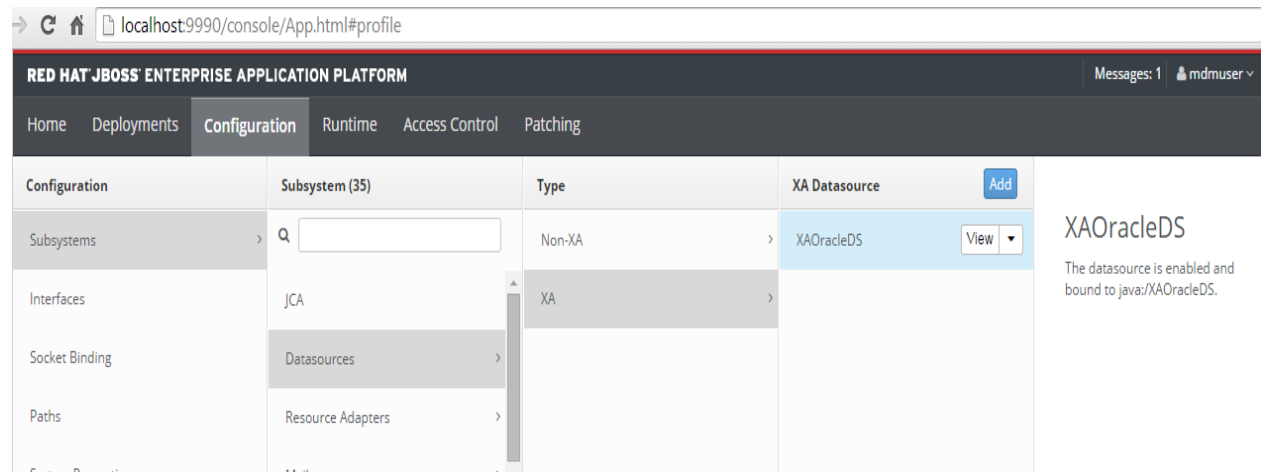
a. Verify JBoss Configuration

We have created XA datasource and Active MQ Queues on the JBoss servers. The same needs to be verified

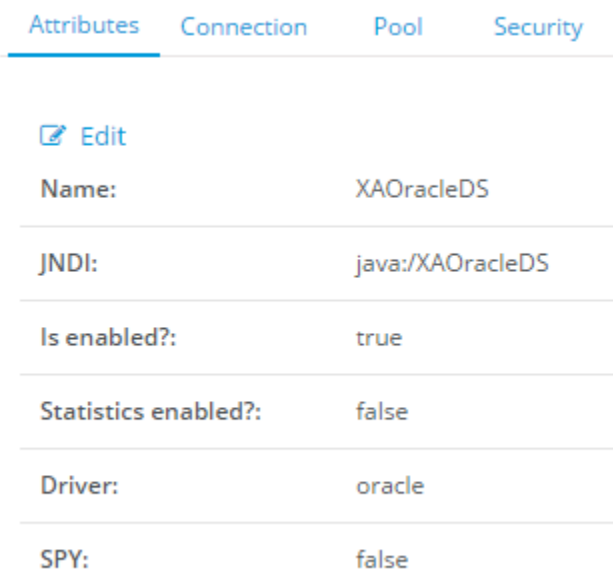
- **Verify XA Datasource**

Login to JBoss Management Server using web browser and <http://localhost:9990/> web url (if you have changed the port then use the same).

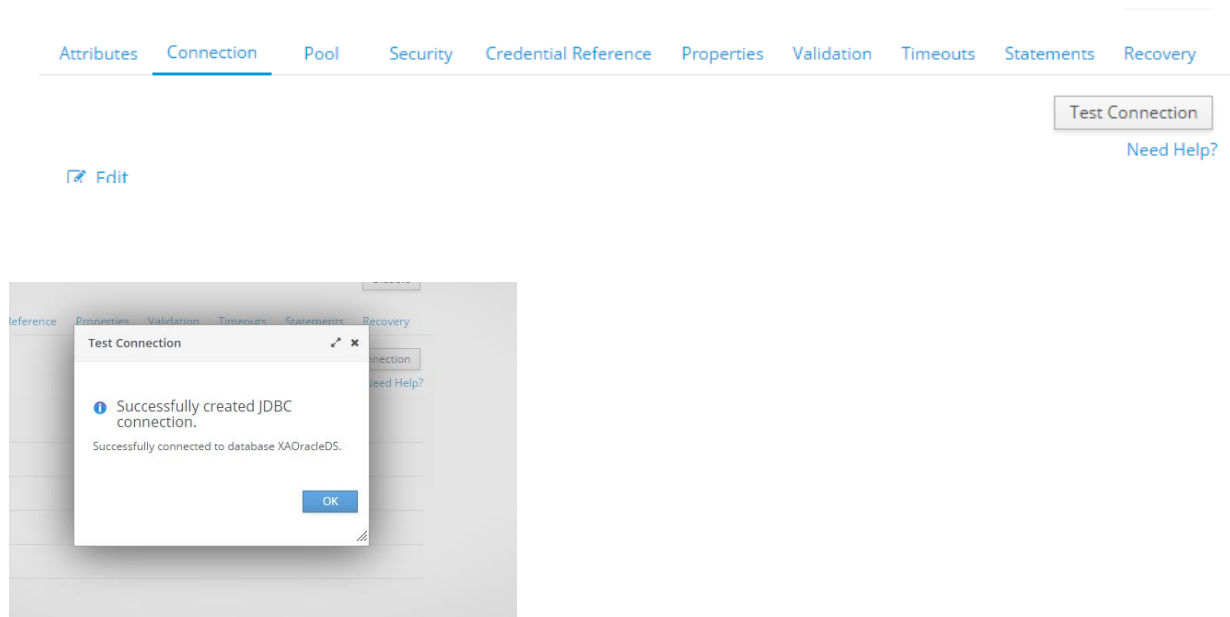
In the Configuration → Datasources → XA → XAOracleDS → 'View' Button



Click on the 'View' button and verify that the datasource is enabled



In the Connection tab, click on the Test Connection button and verify that Datasource connection is successful.



- **Verify Active MQ Queues**

Go to Configuration → Messaging –Active MQ → Messaging Provider → Yug → 'Queues/Topics'

Home	Deployments	Configuration	Runtime	Access Control	Patching
Configuration		Subsystem (35)	Settings	Messaging Provider Add	
Subsystems >		<div>Q <input type="text"/></div>	Messaging Provider >	Yug <div>Queues/Topics ▾</div>	
Interfaces		Messaging - ActiveMQ >	JMS Bridge	default	
Socket Binding		Security >			

Verify that you see the default Request and Response Queues

Core Queues

JMS Queues/Topics

Security Settings

Address Settings

Diverts

JMS Endpoints: Provider Yug

Configuration for JMS queues and topics.

Queues Topics

Add

Remove

Name	JNDI
YUG.DEFAULT.RESPONSE	[java:jboss/com/yugandhar/default/responseQueue]
YUG.DEFAULT.REQUEST	[java:jboss/com/yugandhar/default/requestQueue]

« < 1-2 of 2 > »

[Edit](#)

[Need Help?](#)

Name: YUG.DEFAULT.RESPONSE

JNDI Names: java:jboss/com/yugandhar/default/responseQueue

Durable?: true

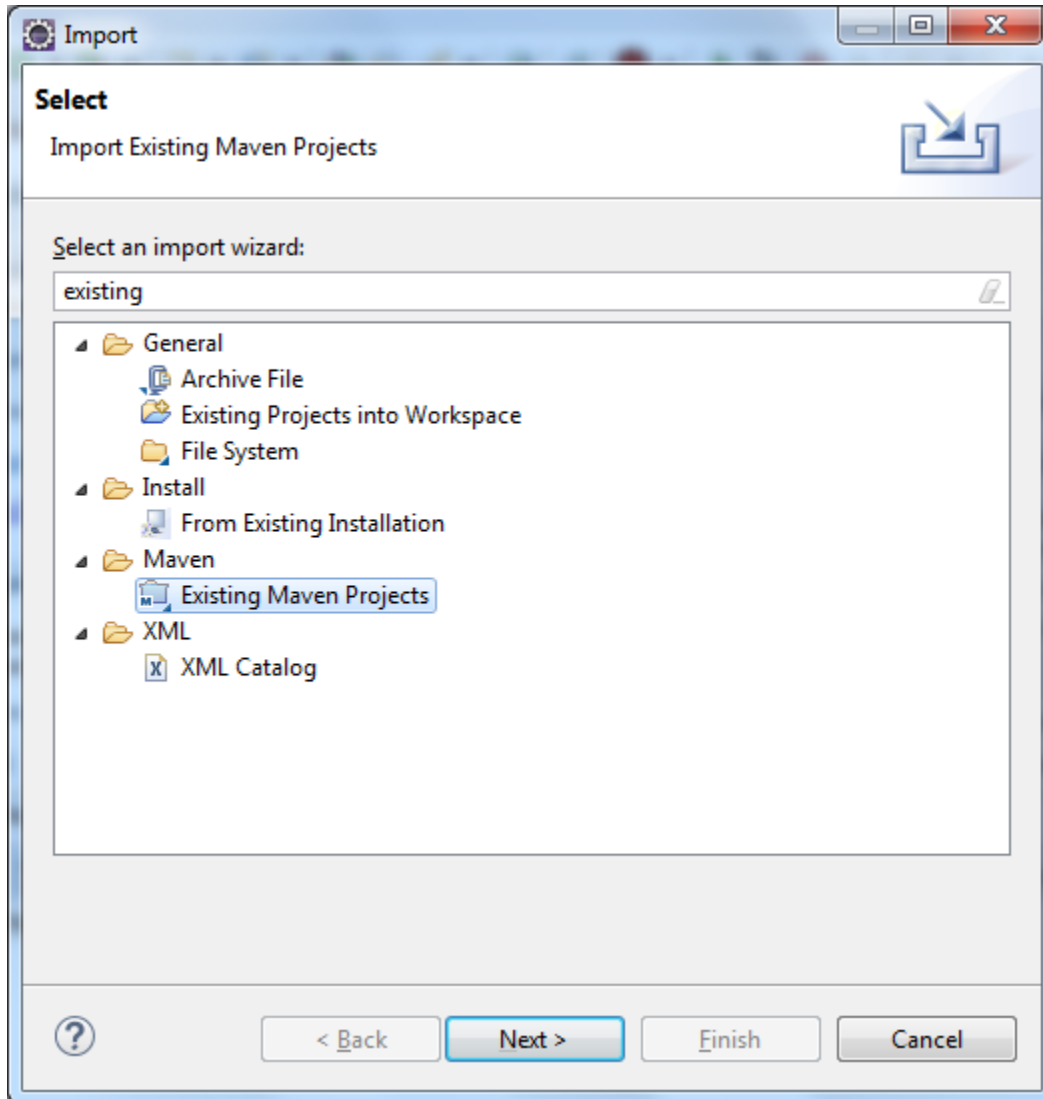
Selector:

This Concludes the JBoss Configuration.

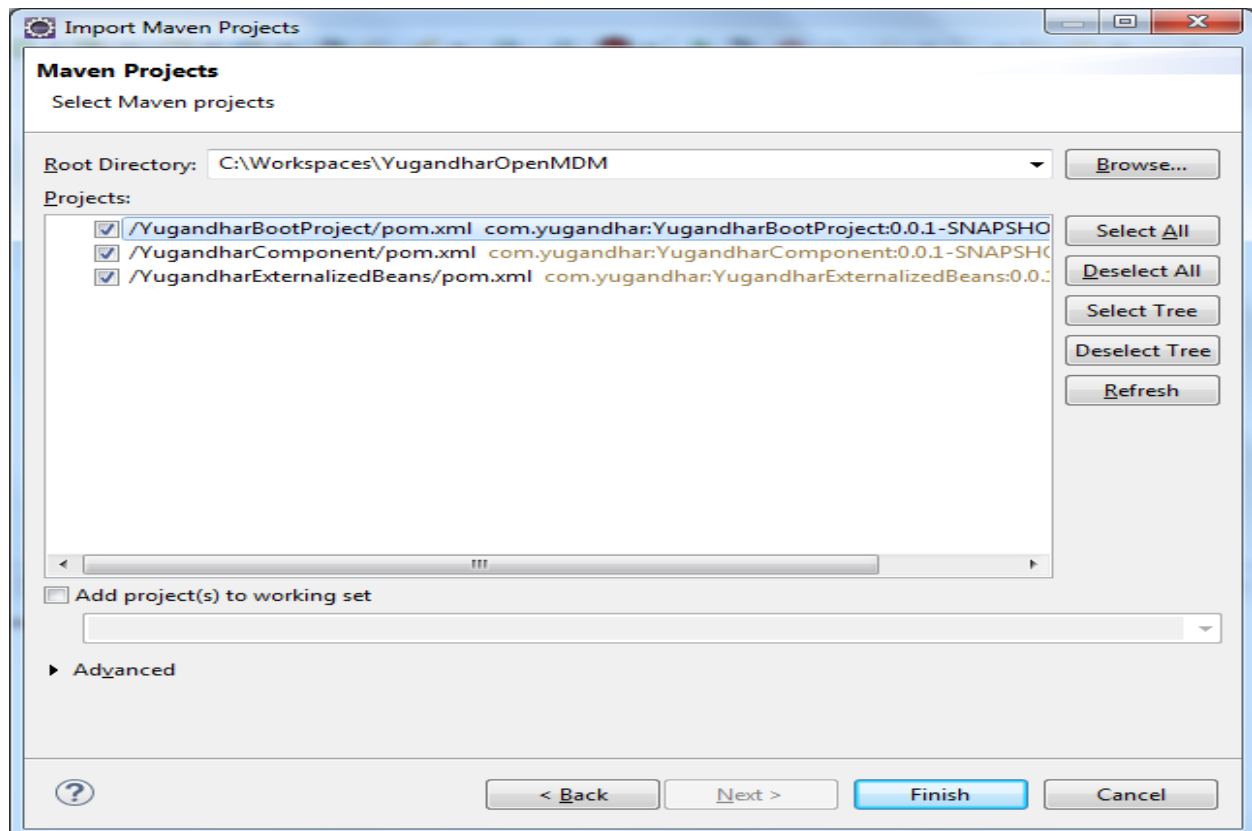
4. Import Yugandhar Projects in the workspace

Download the Yugandhar Projects from the github repository, extract it in the workspace directory (e.g. C:\Workspaces\YugandharOpenMDM) and import in Workspace

Go to File → Import Menu → Existing Maven Projects →



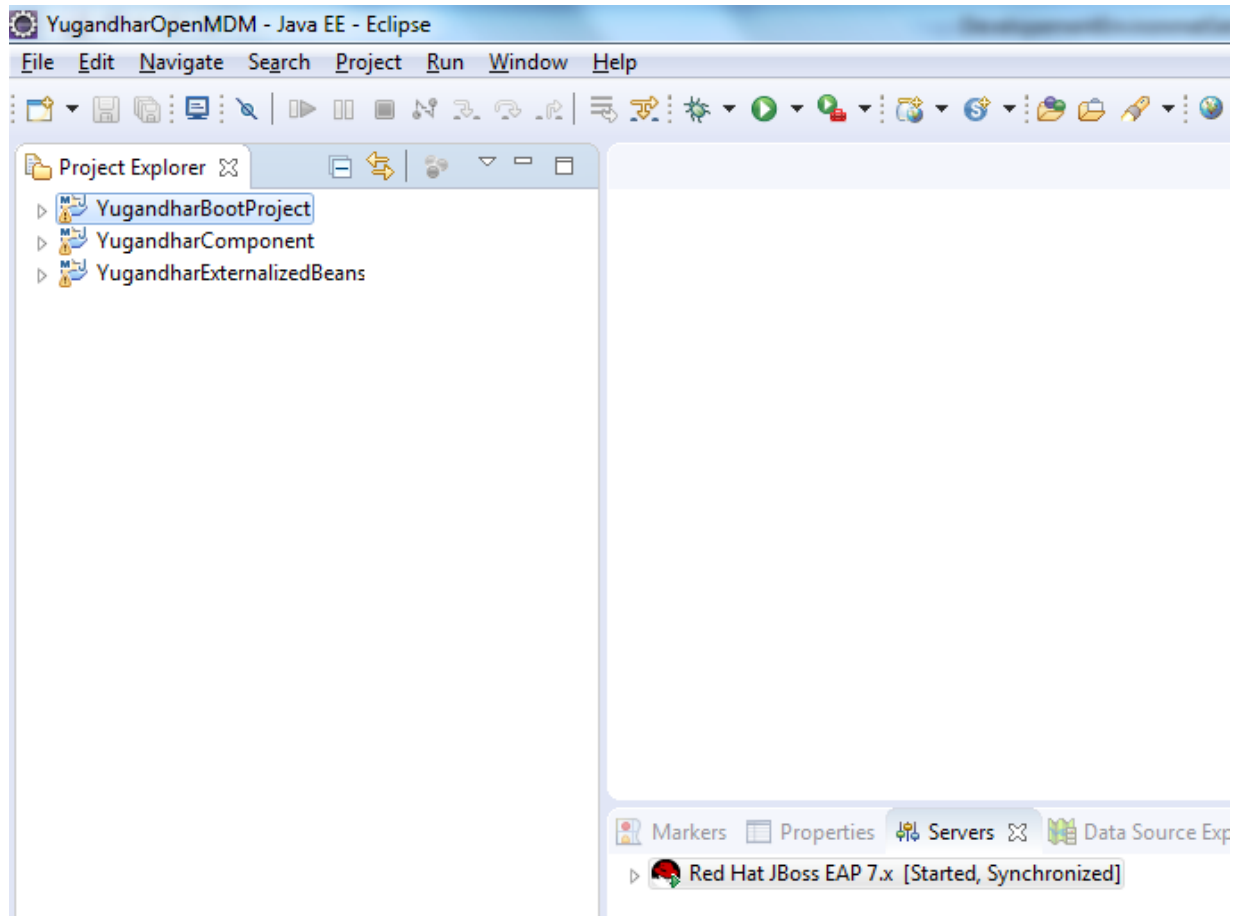
Click on Next Button



Select the three projects and click Finish.

It may take some time as eclipse will download the maven jars automatically and build the project. You may track the progress in 'Progress' tab.

Make Sure that your workspace is error free

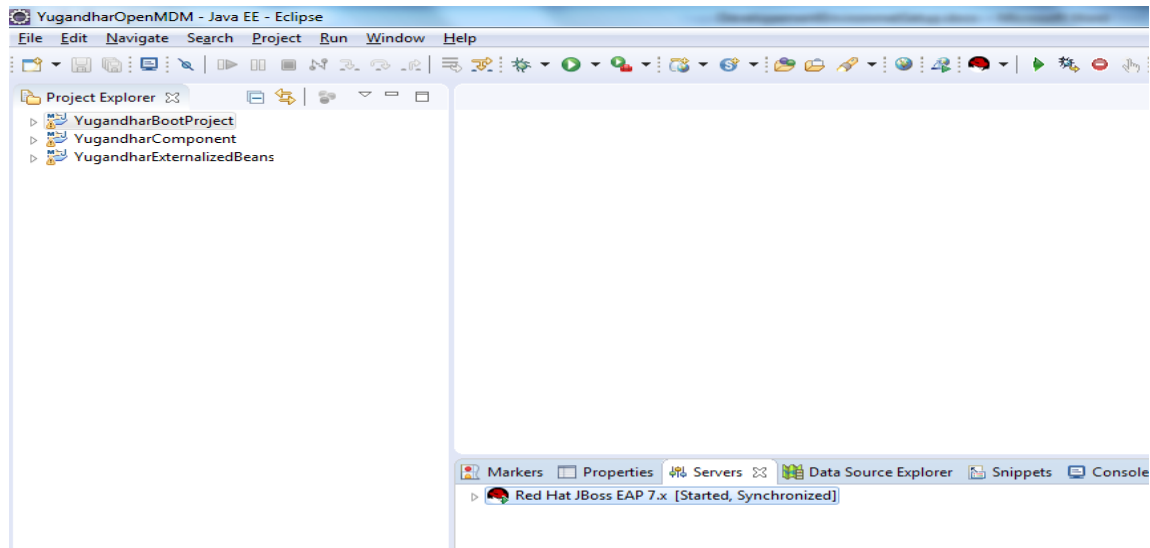


You may choose to change the log file location in yugandhar_logback.xml

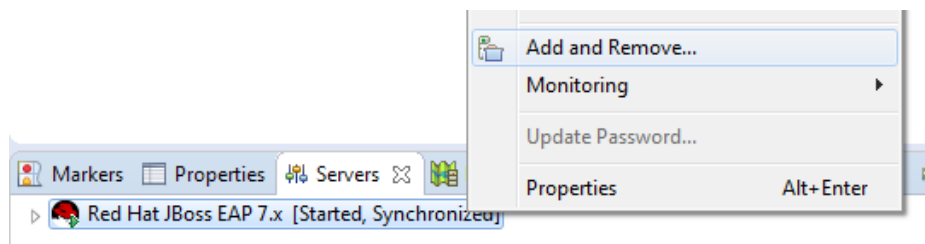
```
<appender name="TIME_AND_SIZE_BASED_APPENDER" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>C:/yugandhar/logs/YugandharCommon.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <fileNamePattern>C:/yugandhar/logs/YugandharCommon.%d{yyyy-MM-dd}.%i.log</fileNamePattern>
    <timeBasedFileNamingAndTriggeringPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP">
```

5. Deploy Yugandhar Boot Project To JBoss Server

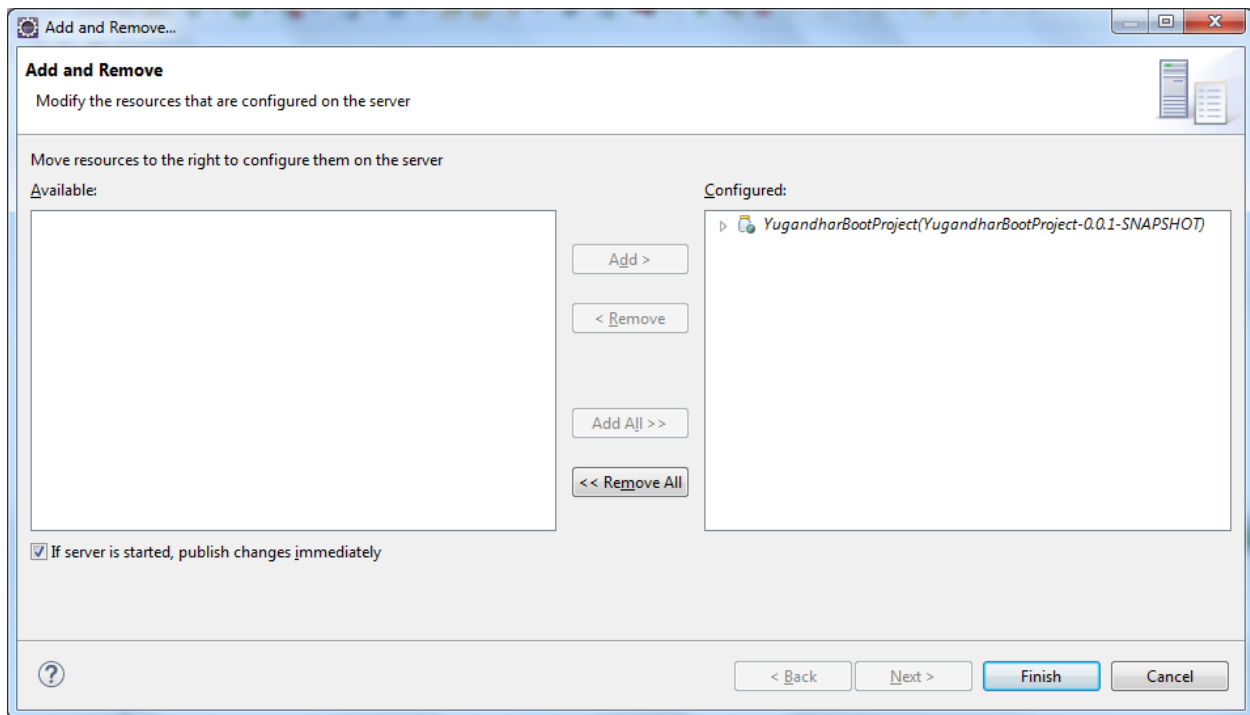
Make sure your jboss server is started. If not, start the server.



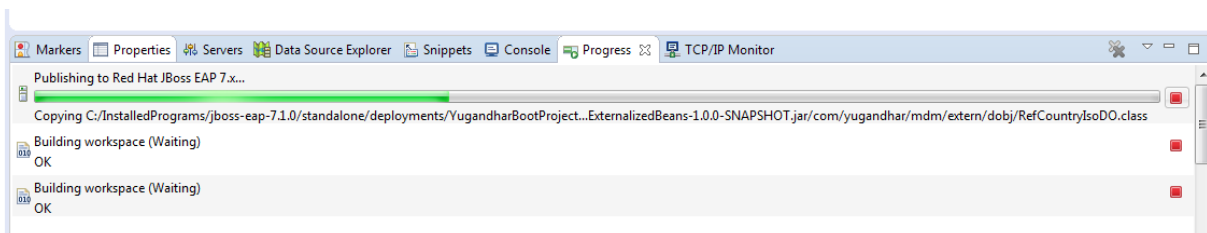
Right click and select 'Add and Remove...'



Select Yugandhar Boot Project and click Finish



Let the project gets published to the server and watch the progress in the progress bar



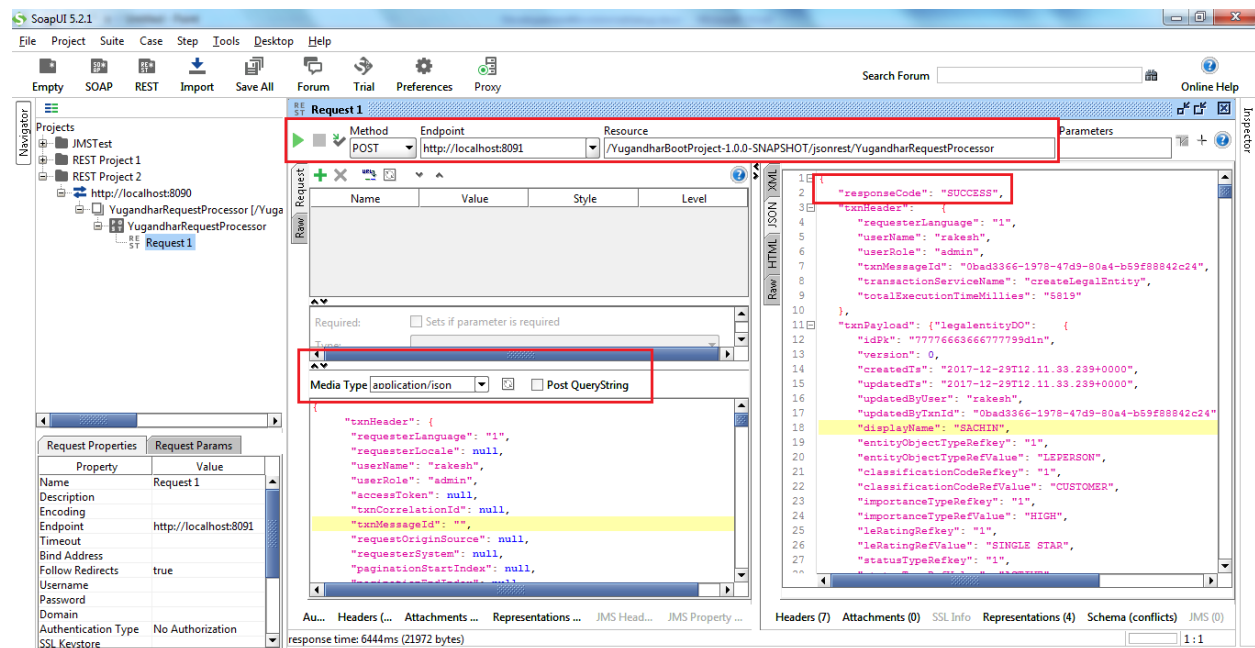
The rest url is as below.

<http://localhost:8091/YugandharBootProject-1.0.0-SNAPSHOT/jsonrest/YugandharRequestProcessor>

use the port 8090 if you are using default port else use the one which you have used for jboss.



Create a new test case and execute it with attached json message



This certifies your workspace.

