# Yugandhar Open MDM Hub for JEE Container Web Server (JEEC)

## Development Environment Setup Guide

Yugandhar Open MDM Hub - JEEC Release - V1.0.0
Date – 11/06/2018

# Contents

## About Yugandhar Open MDM Hub

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.

## System Requirements

Below are the System Requirements for setting up Development Environment

1. OS – Windows 7 enterprise addition, Service Pack 1 or later
2. 8GB RAM and 100 GB Storage
3. Eclipse Java EE IDE for Web Developers. Oxygen.3a Release (4.7.3a) or later
4. JBoss EAP 7.1.0 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 2.0.2.RELEASE
8. Hibernate Tools
9. Databases
    a. Oracle Database 11g Release 11.2.0.2.0 or later
    b. Oracle 12c
    c. MariaDB v10.3.x
10. Oracle SQL Developer/HeidiSQL
11. SOAPUI /postman 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 would be tedious task.

IMPORTANT NOTICE - Please read the licensing terms of all the above listed software before using for commercial as well as non-commercial purpose. Yugandhar team would not be responsible for licensing violations if any.

## Software Download links

### Eclipse

Eclipse Java EE IDE for Web Developers.

**Version** - Oxygen.3a Release (4.7.3a) or later

**Download page** - http://www.eclipse.org/downloads/

### 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/

## 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

## Database

Download the database as per your choice

### Oracle

Download Oracle from Oracle download site
**Version** - Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit OR Oracle Database 12c
**Download link** - https://www.oracle.com/database

### MariaDB

Version: MariaDB v10.3 or later
https://mariadb.org/

## 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

## Hibernate Tools

You may choose to download the Hibernate Tools (Now renamed as JBoss Tools) from the link below. 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

## Oracle SQL Developer

Download SQL developer to connect to database

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

## Heidi SQL

Heidi SQL comes packaged along with Maria DB installable; you may use the same to explore maria db.

## Database drivers

- Oralce JDBC drivers: download ojdbc14.jar from below link
http://www.oracle.com/technetwork/apps-tech/jdbc-10201-088211.html
- Oralce JDBC drivers: Oracle 11g and 12c driver for JDK7 and JDK8 ojdbc7.jar
http://www.oracle.com/technetwork/database/features/jdbc/jdbc-drivers-12c-download-1958347.html
- **MariaDB drivers:** Download mariadb-java-client-2.2.3.jar from below location https://mariadb.com/downloads/connector
https://downloads.mariadb.com/Connectors/java/connector-java-2.2.3/mariadb-java-client-2.2.3.jar

# Setup Database

Install either Oracle or mariaDB database as per requirement. Oracle 11g/12c/MariaDB 5.5.3 are supported. The step by step installation instructions for installing the Oracle 11g, 12c database and Oracle SQL Developer is out of scope of this document.

By Default Yugandhar Open MDM Hub uses the schema YUG_OWNER. If different user name (schema name) is needed then modify all the scripts with required schema name.

## Oracle setup:

### Oracle Install

Install the Oracle database using the instructions mentioned below.

**Oracle 11g**: https://docs.oracle.com/cd/E11882_01/nav/portal_11.htm

**Oracle 12c:**
http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/12c/r1/Windows_DB_Install_OBE/Installing_Oracle_Db12c_Windows.html

### Yugandhar MDM HUB schema setup for oracle

Github repository link - https://github.com/yugandharproject/yugandhar-open-mdmhub

Download the scripts from '<gitrepo>/resources/yugmdm-dbsetupscripts-oracle'

1. CreateTablespaces.sql – create YUG_OWNER user as the tables. Needs DBA access
2. FullSchema.sql – creates the full schema. Can be executed with YUG_OWNER user access
3. CreateSequence.sql – Can be executed with YUG_OWNER user access
4. LoadTableDataWrapper.sql – Can be executed with YUG_OWNER user access

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 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 YUG_OWNER user

***User Schema:***

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

## MariaDB setup:

### MariaDB Install

Install MariaDB as per instructions mentioned in below links

https://mariadb.com/kb/en/library/getting-installing-and-upgrading-mariadb/

https://mariadb.com/products/get-started

### Yugandhar MDM HUB schema setup for MariaDB

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

Github repository link - https://github.com/yugandharproject/yugandhar-open-mdmhub

Download the sqls from '<gitrepo>/resources/yugmdm-dbsetupscripts-mariadb'

- "1.yugmdm_mariadb_createuser-and-database.sql" – login through root or DBA

Verify the logs and check that all the steps are executed correctly. Also verify that 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 HUB_DATATS – used for data tables as well as reference tables

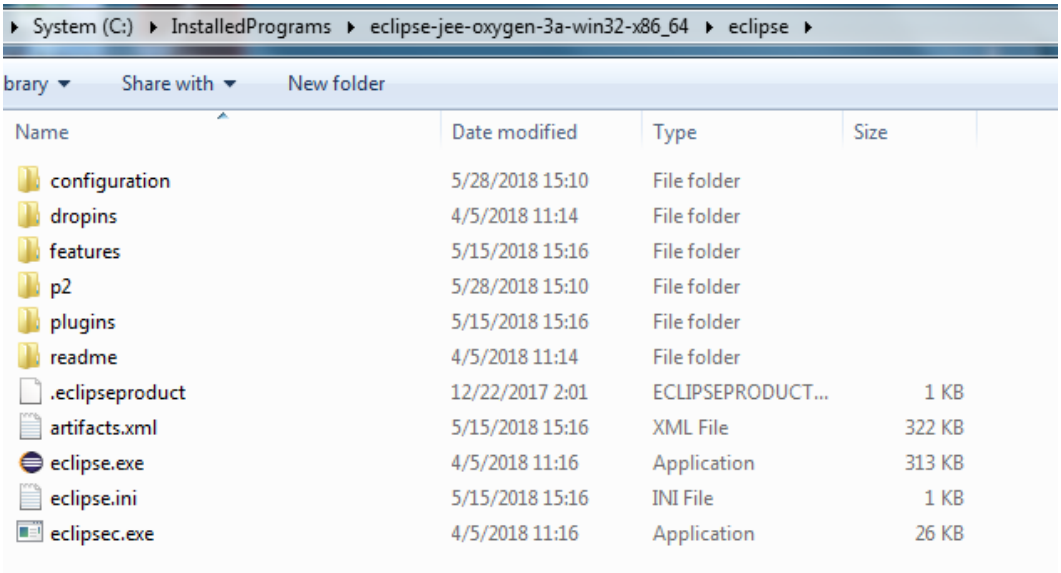
MDM HUB_INDXTS – used for Indexes

***DB User YUG_OWNER:*** *Default user used by Yugandhar Open MDM Hub to connect to database.*

***Database*****:** *YUG_OWNER database is created.*

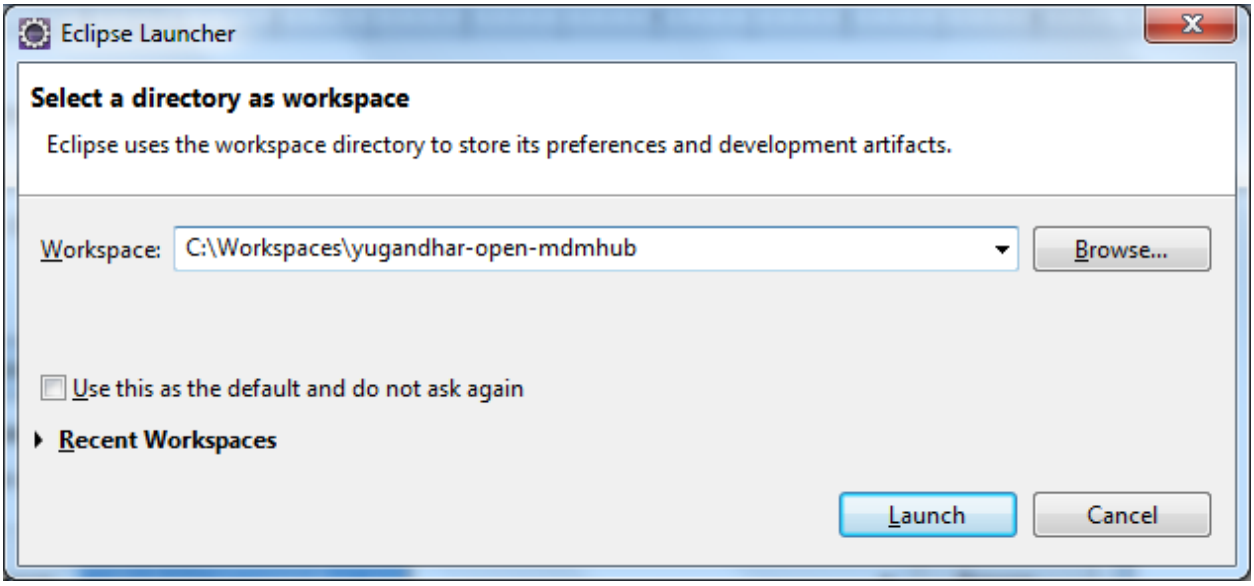
## Setup Workspace

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

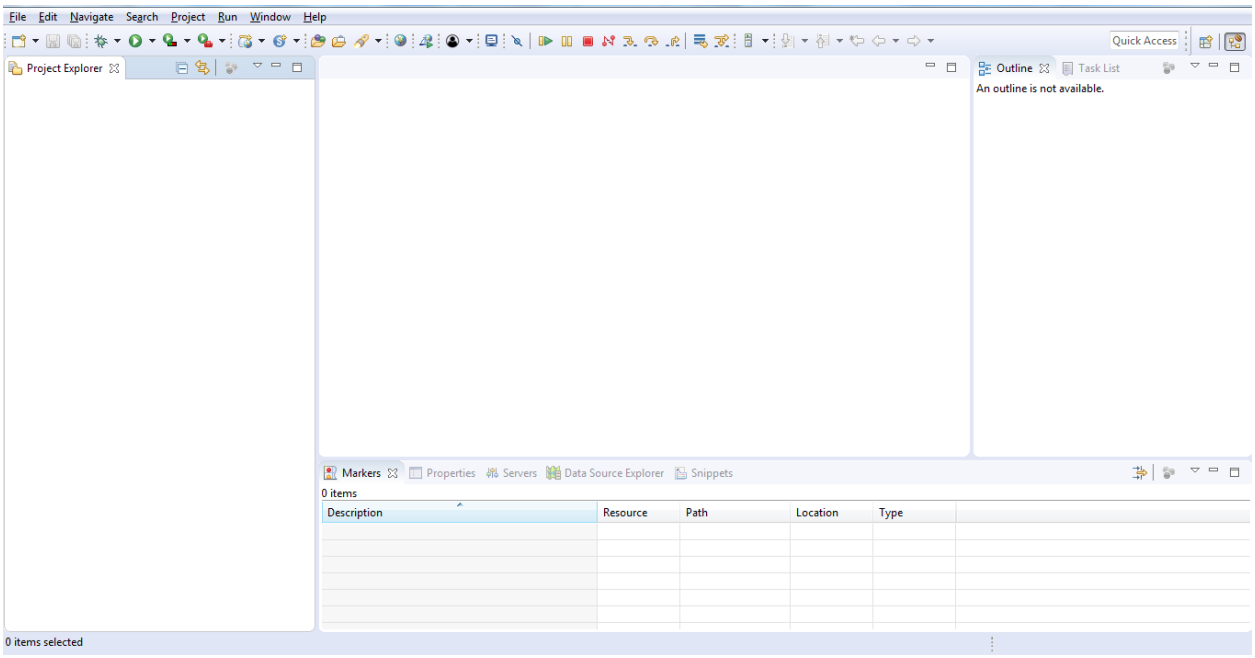



The extracted folder would have above files, click on eclipse.exe to start the eclipse IDE.

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\yugandhar-MDM Hub folder.




Your workspace is ready to be configured now.



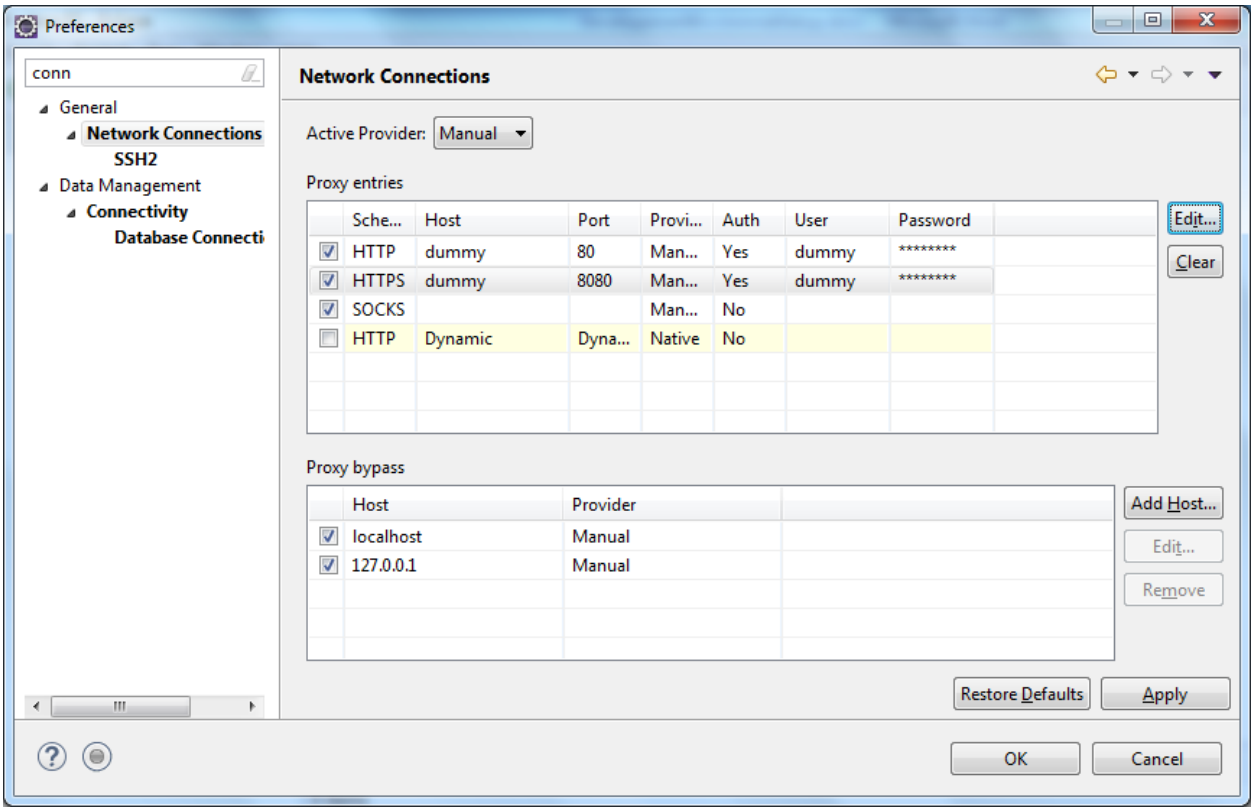
## Setup up Network Connections

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.

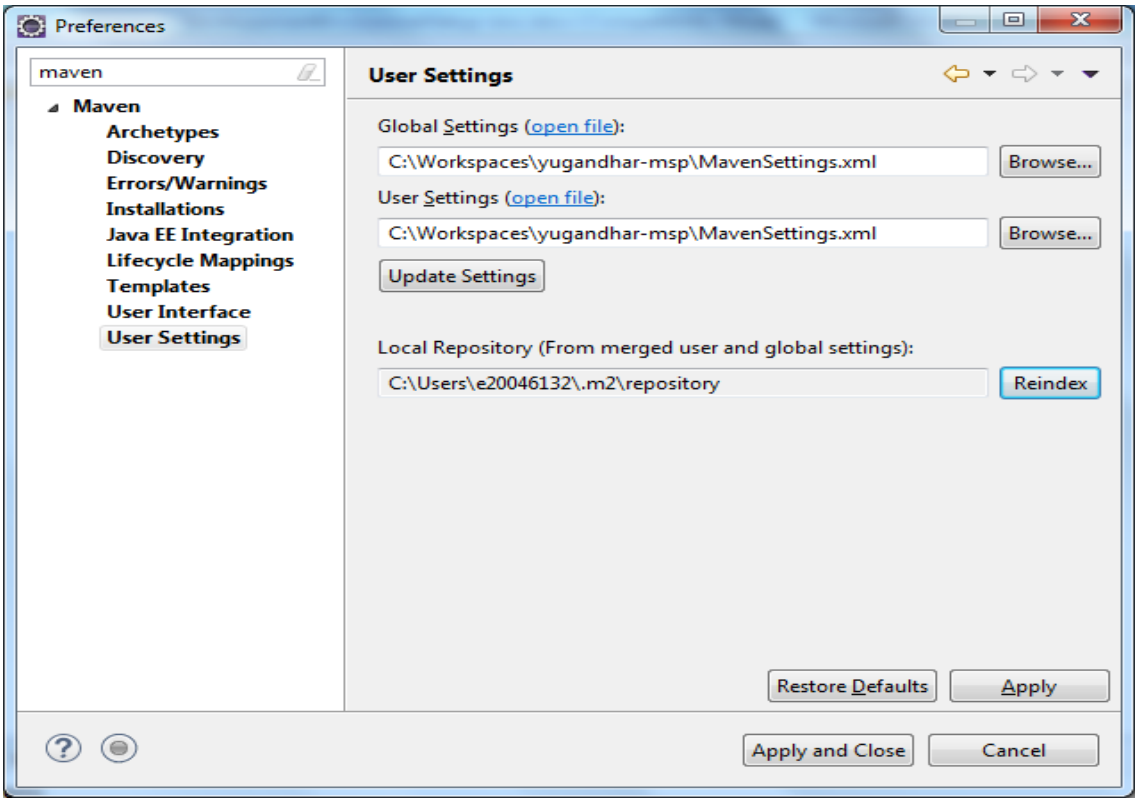
You may also download sample MavenSettings.xml from git hub resources/mavensettings folder.

```
<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>C:\Users\<UserName>\.m2\repository</localRepository>
  <interactiveMode/>
  <usePluginRegistry/>
  <offline/>
  <pluginGroups/>
  <servers/>
  <mirrors/>
  <proxies>
    <proxy>
      <id>myproxy</id>
      <active>true</active>
      <protocol>http</protocol>
      <host>dummy</host>
      <port> dummy </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. Also change the localRepository to users folder.

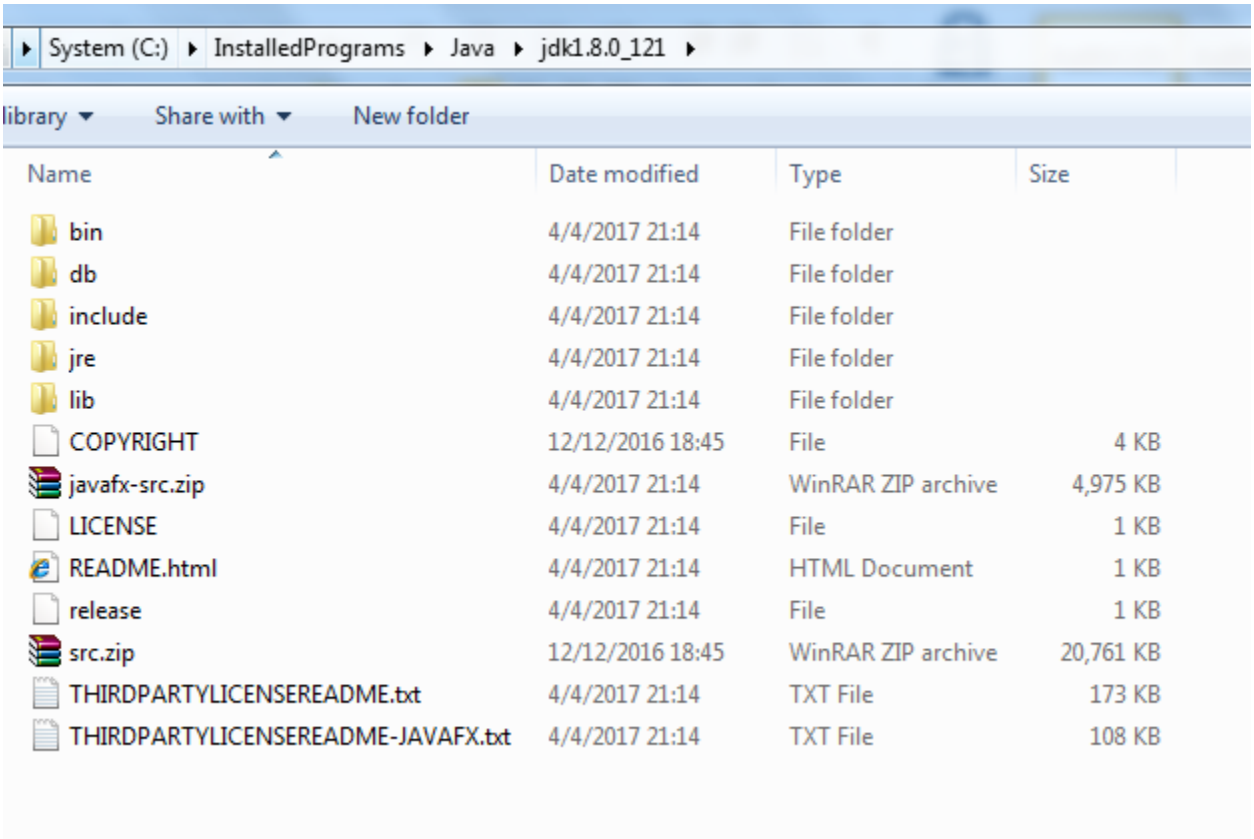
Now go to Eclipse Menu → windows → preferences → maven → User Settings → provide path of the file MavenSettings.xml in the local and global settings as shown in screenshot below



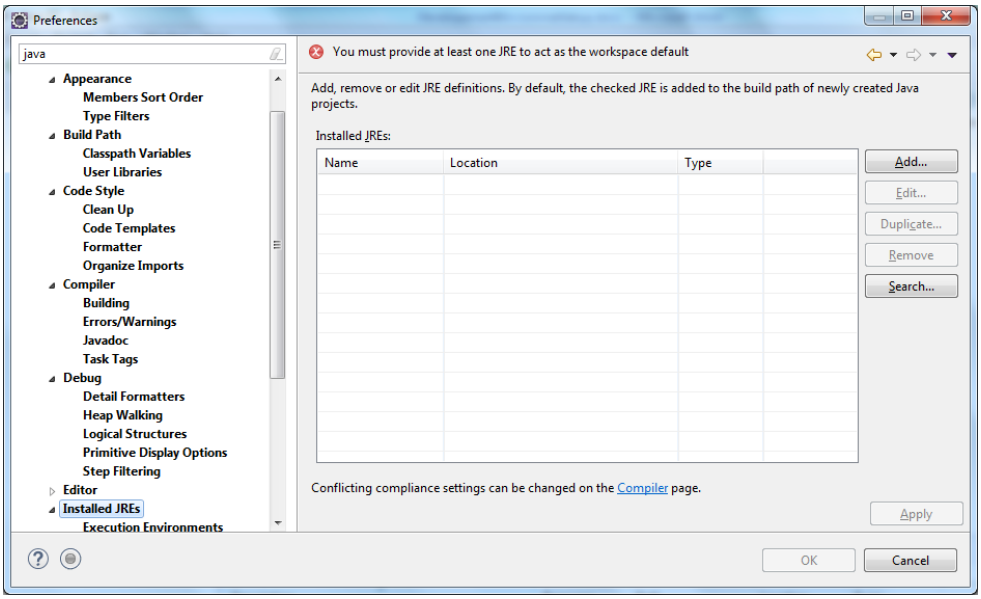
## Set JDK Path

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

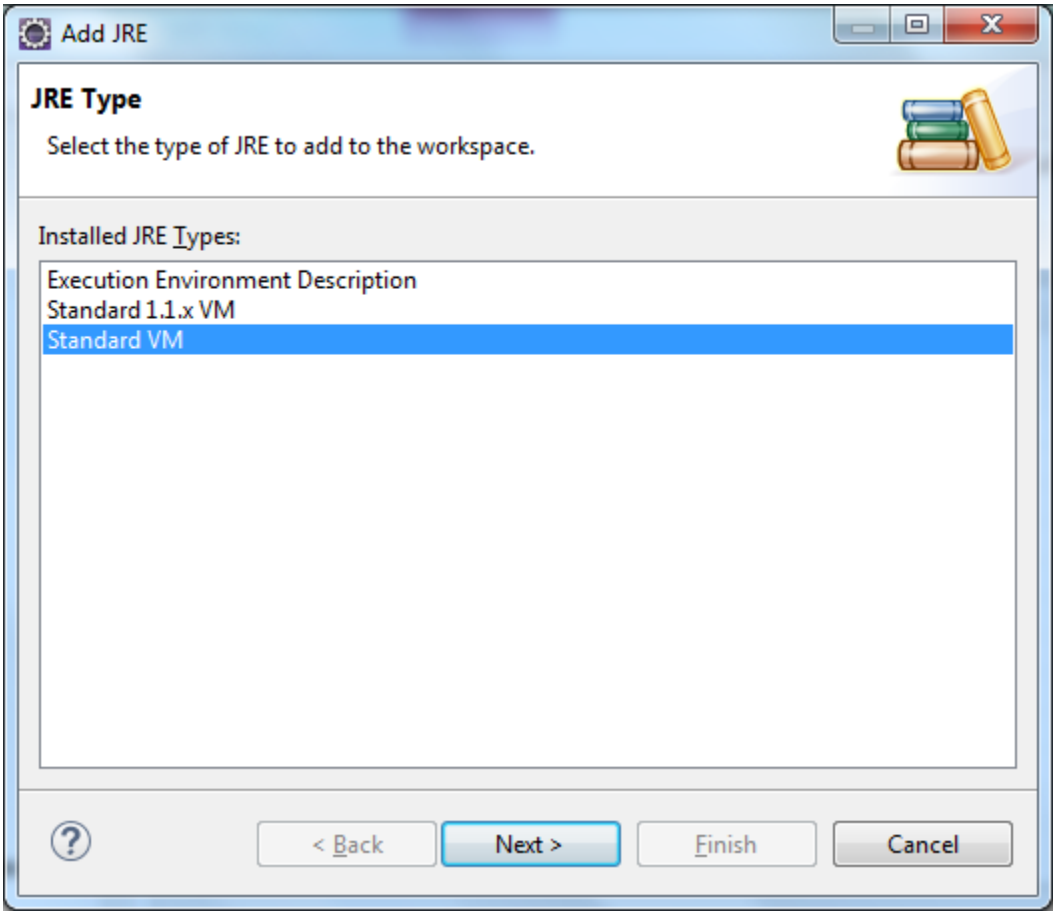
C:\InstalledPrograms\Java\jdk1.8.0_121. The directory should look something like below




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

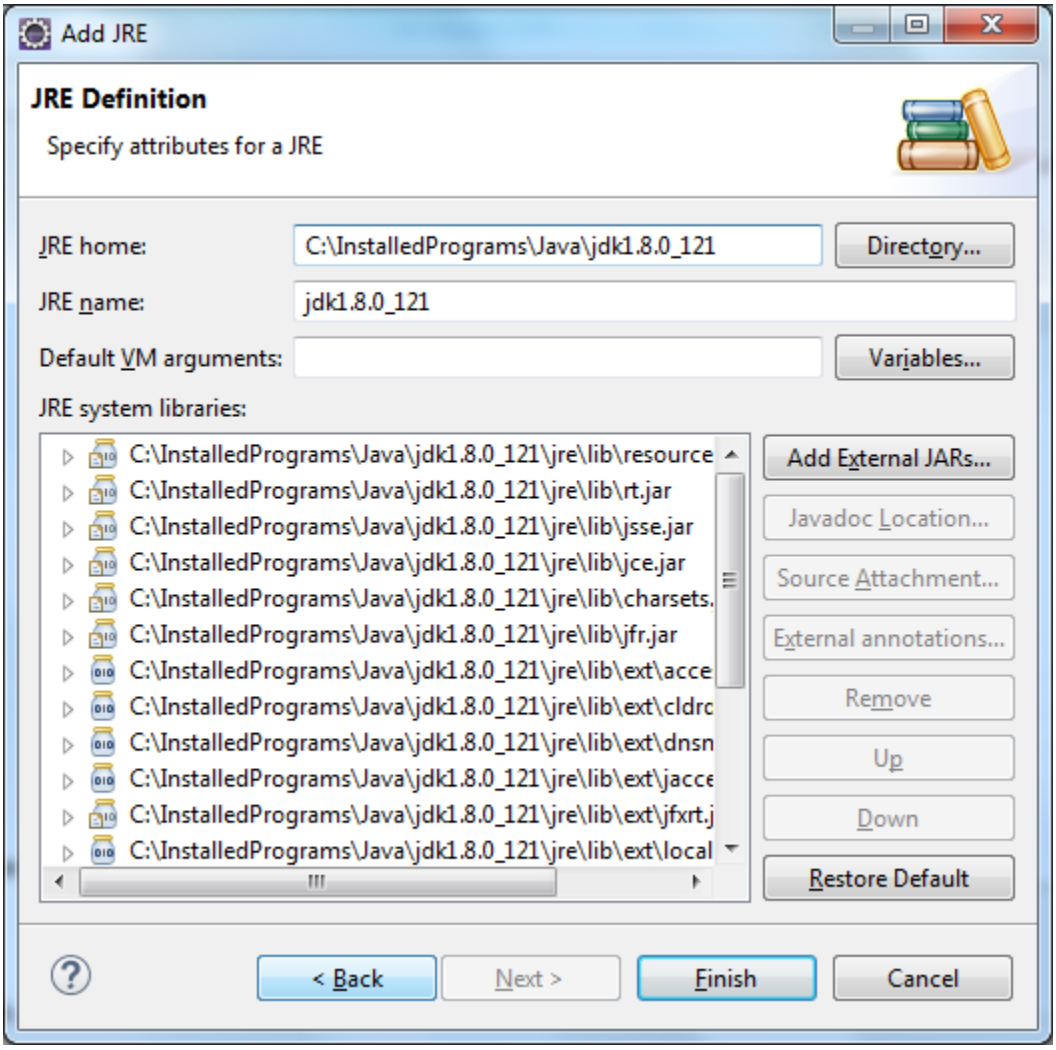



Click Add

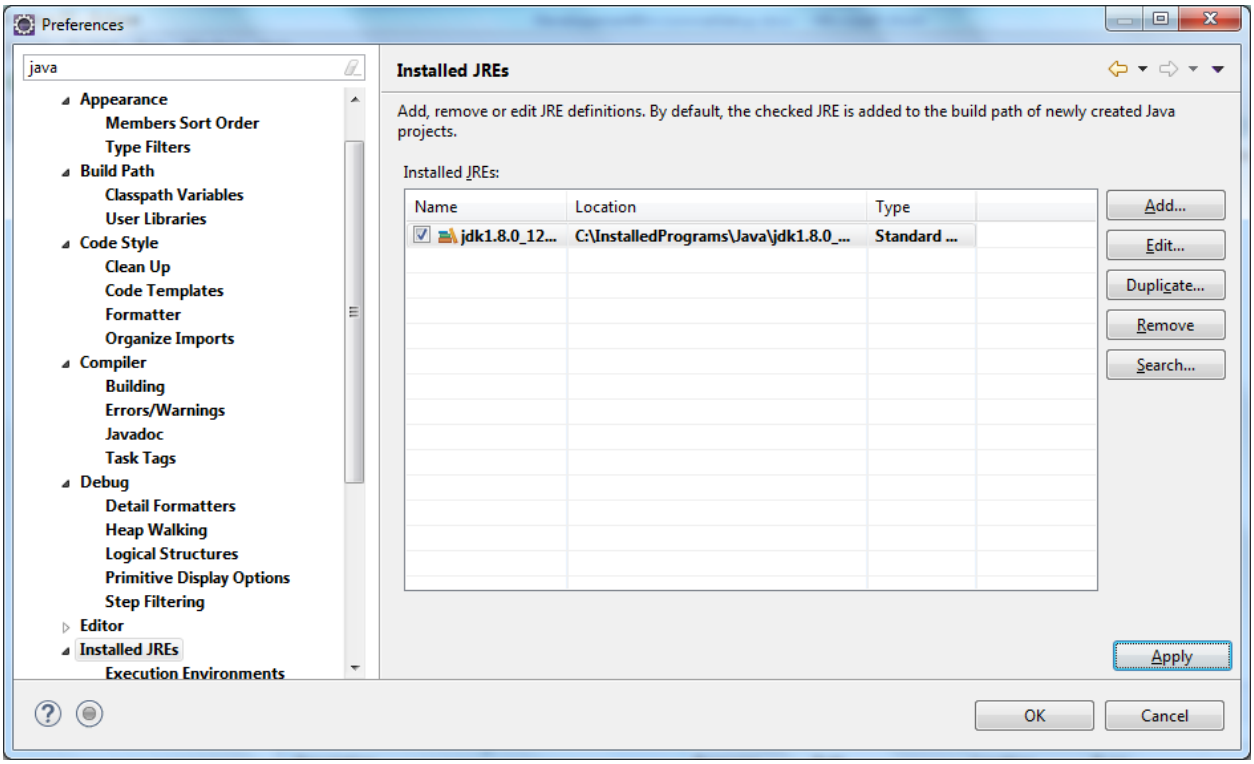



Click Next

Provide the JDK folder name where JDK is extracted




Click finish

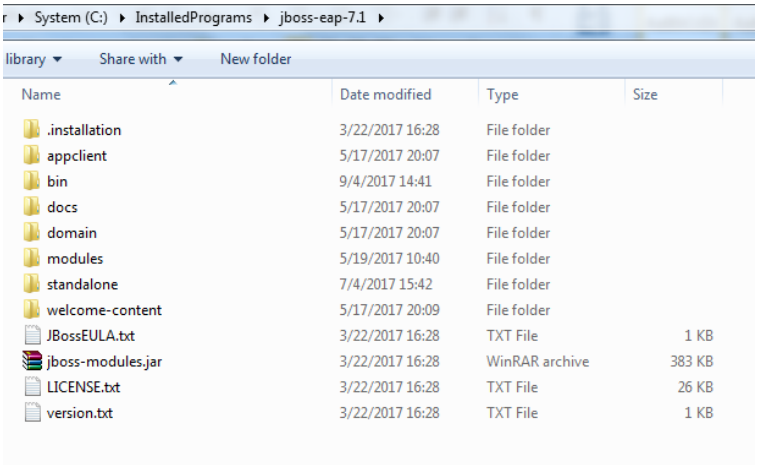



Check the textbox against recently added JDK, click Apply and OK.

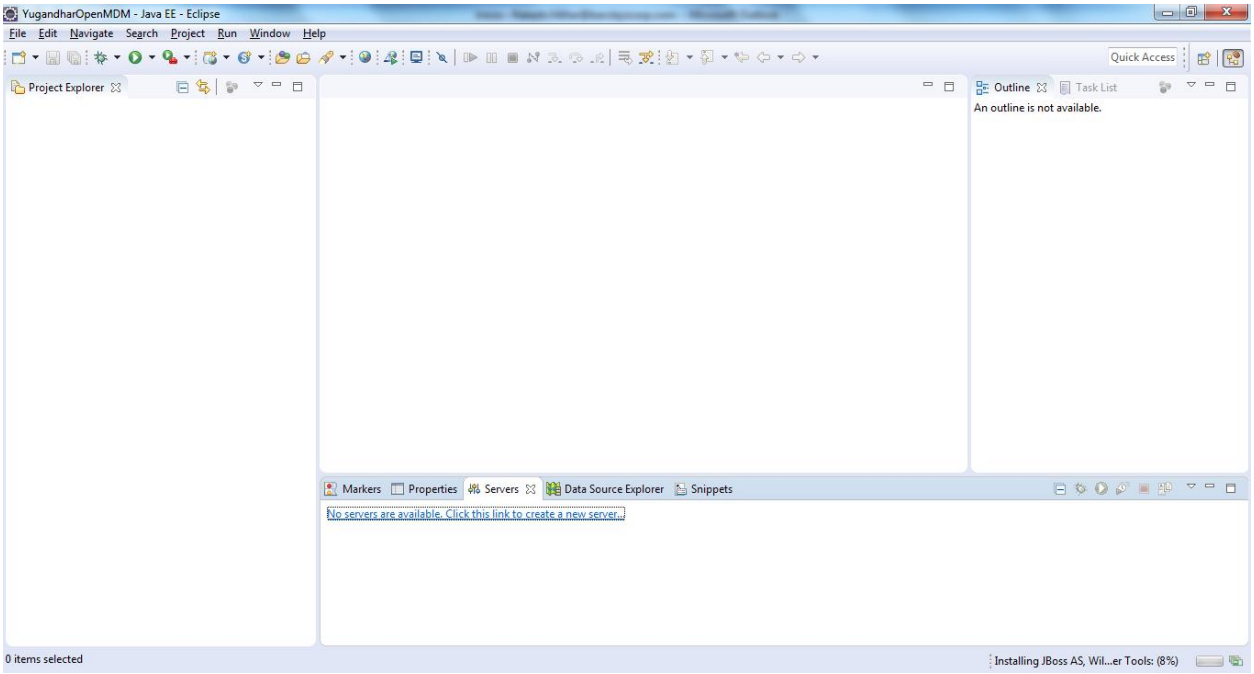
## Configure RedHat JBOSS EAP

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

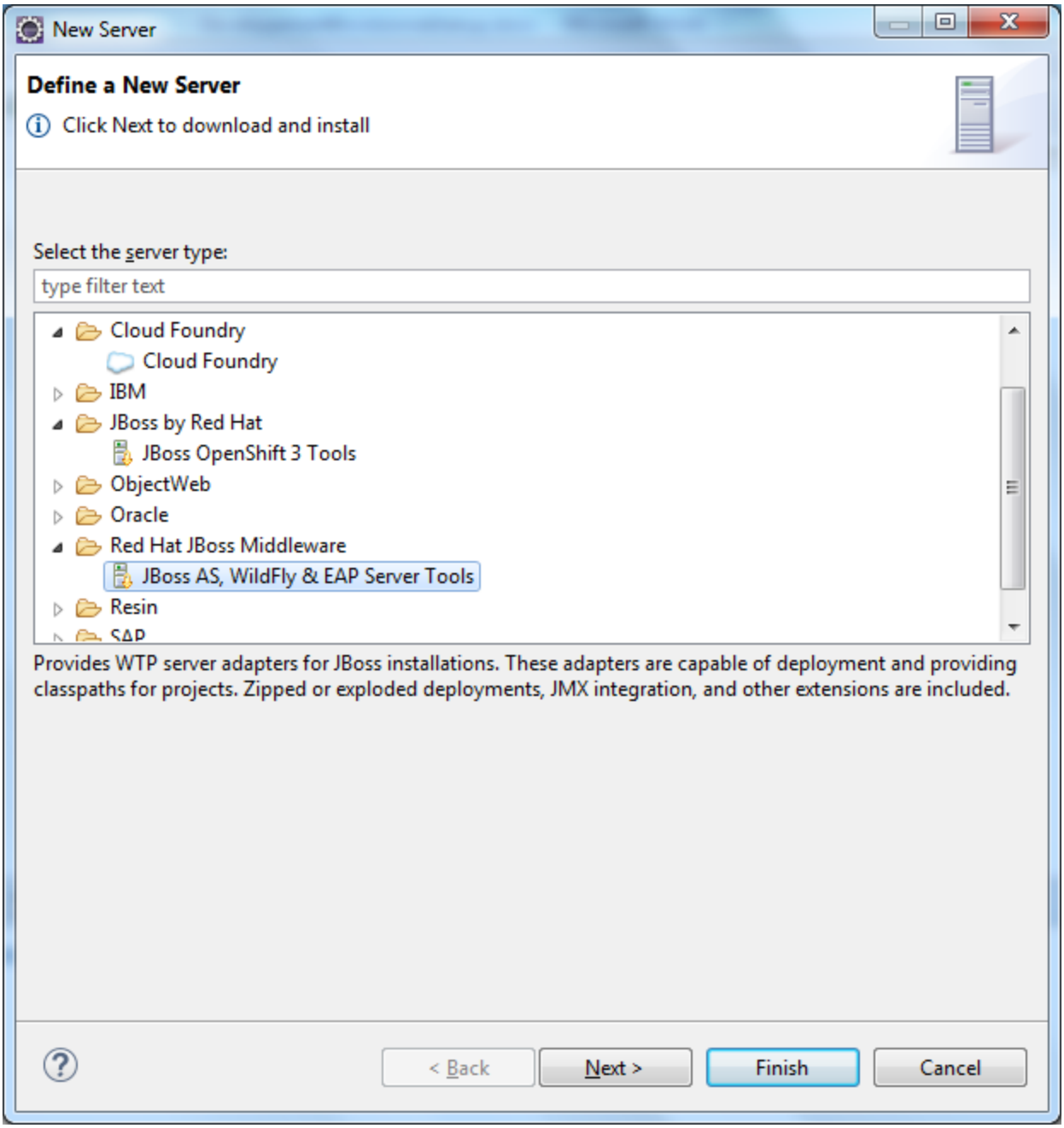
C:\InstalledPrograms\jboss-eap-7.1



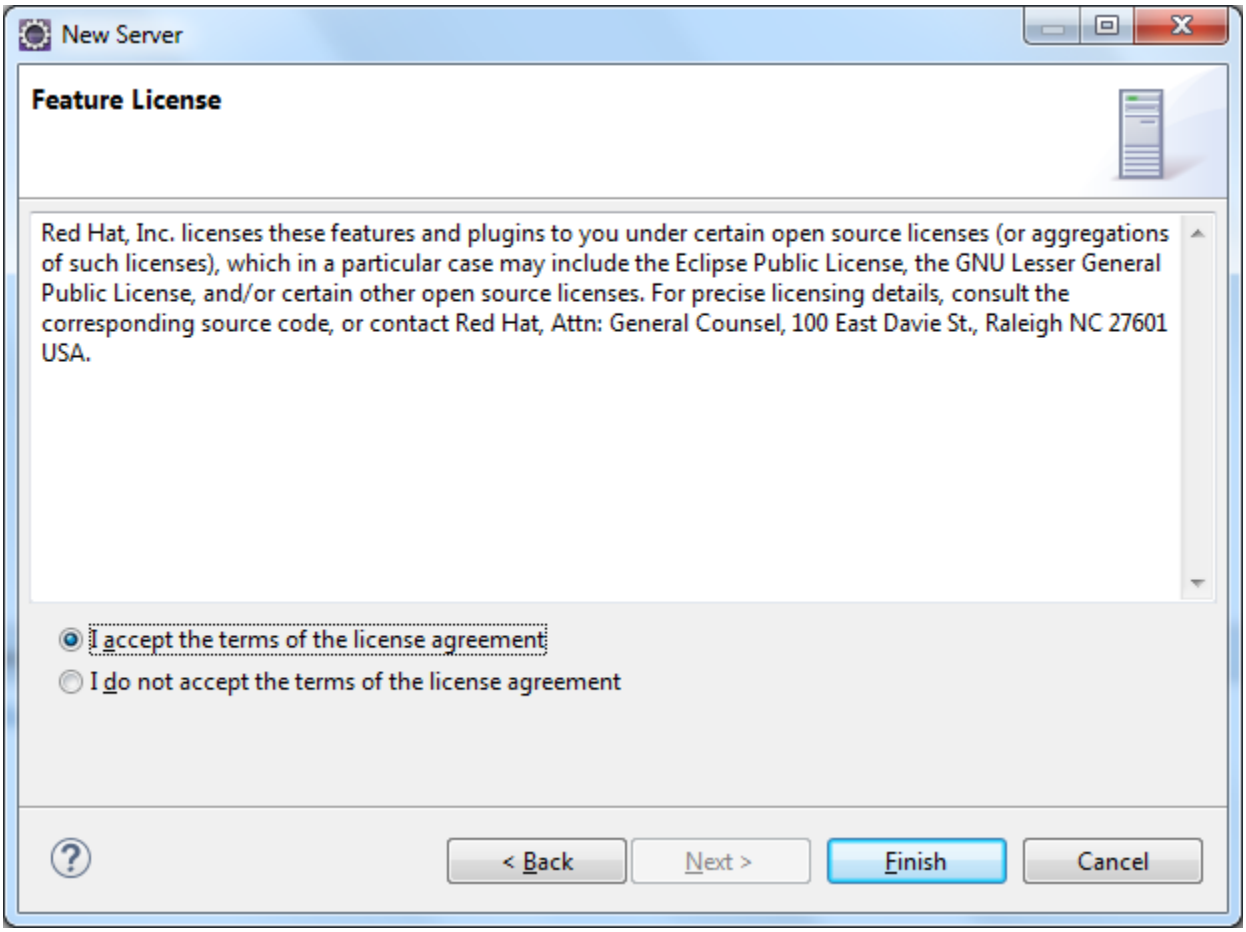

Go to eclipse 'Servers' tab and click on the link to create new Server.



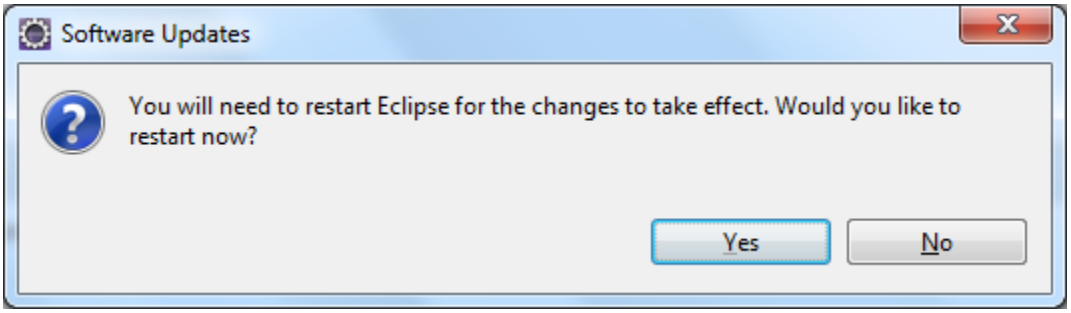

Click the Red Hat Jboss Middleware option as shown in below screenshot



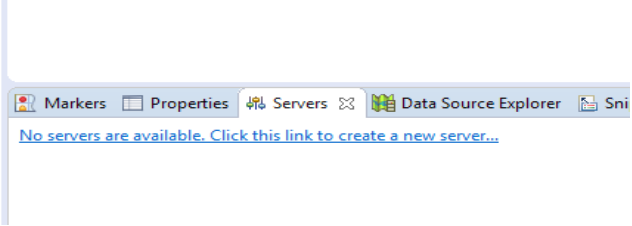
Click Next. It may take some time to process after which will ask to accept the licensing terms. Accept the license and click Finish.



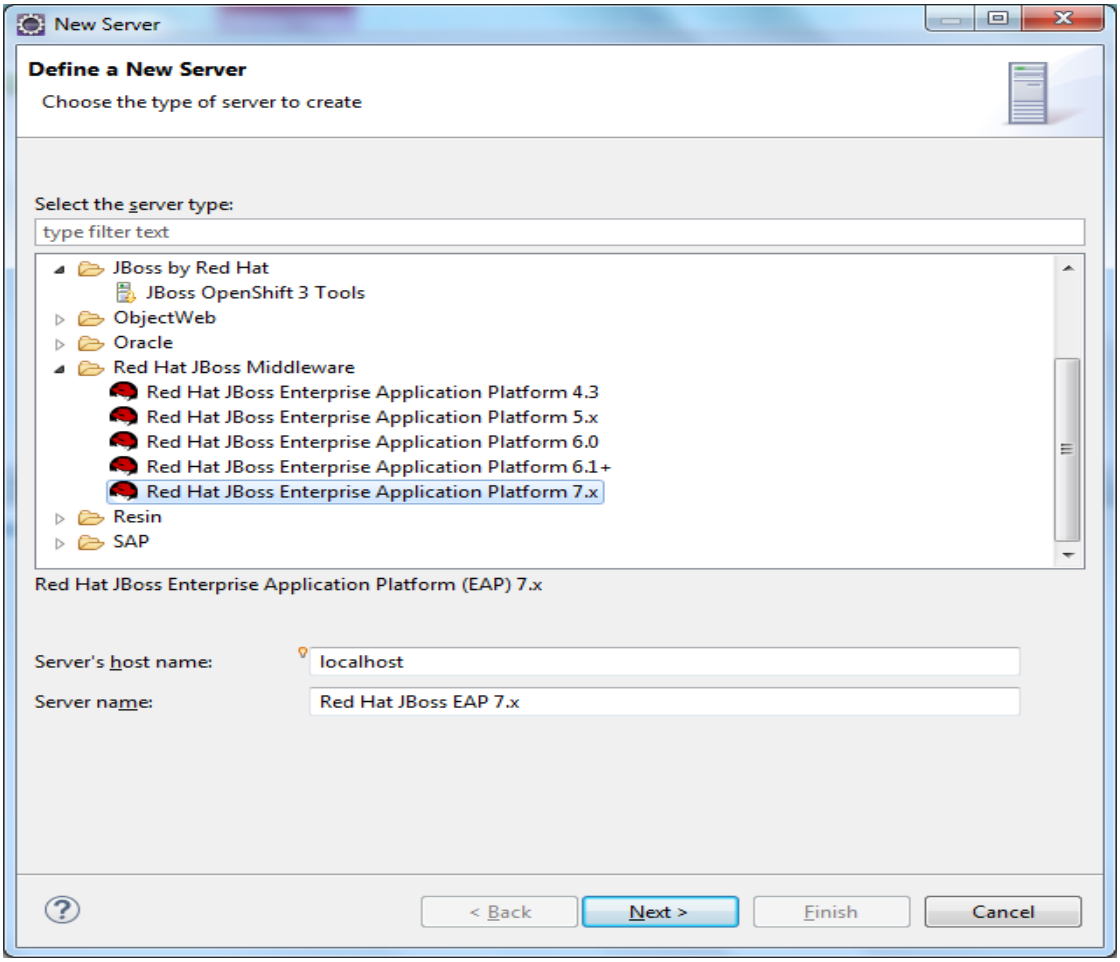

Eclipse will automatically download and install the JBoss Tools which may take some time based on the speed of your internet connection. The eclipse IDE needs to be restarted after this step so that JBoss EAP tools are available in eclipse.



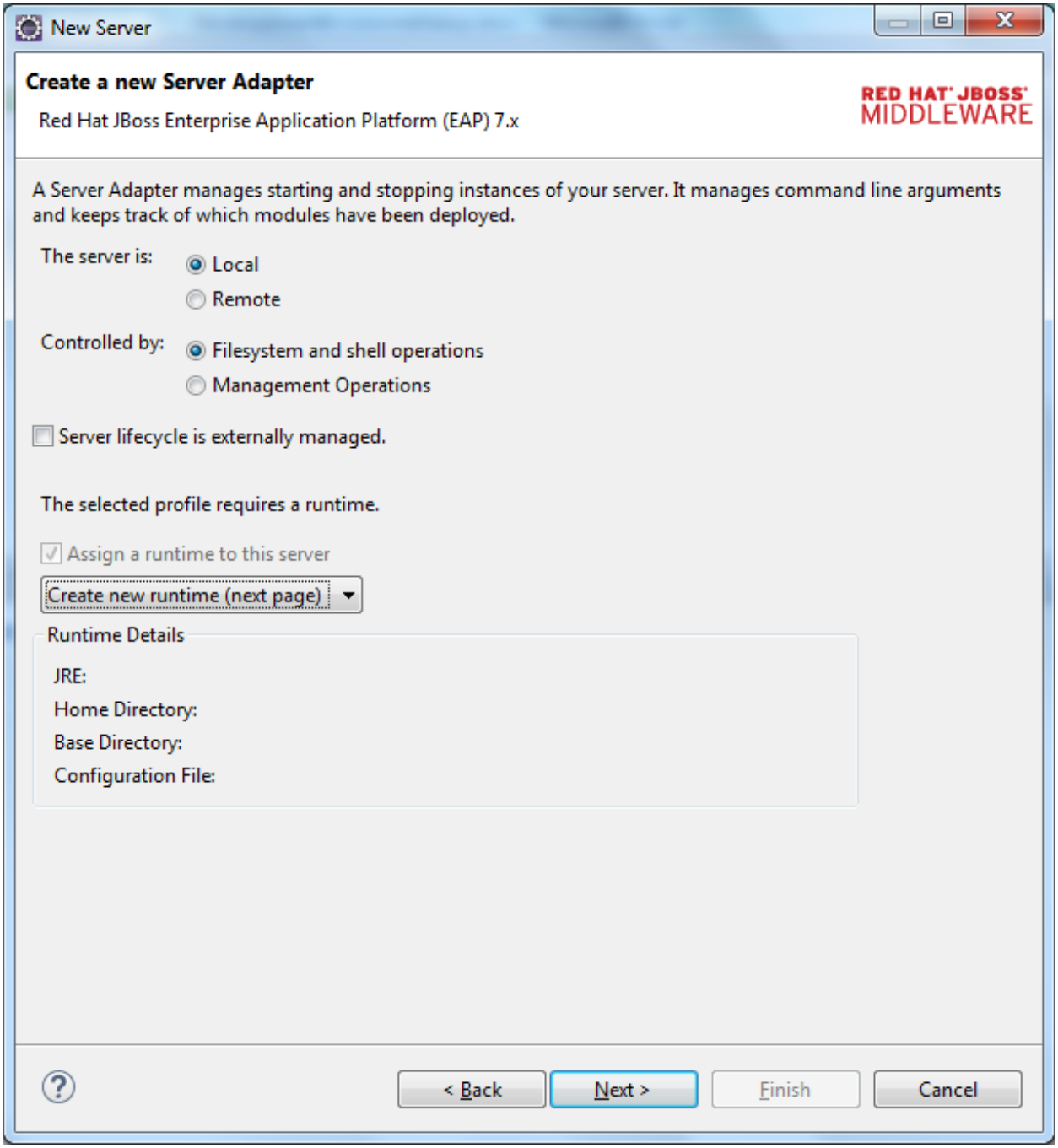

Click on the Server Tab and Create new server link



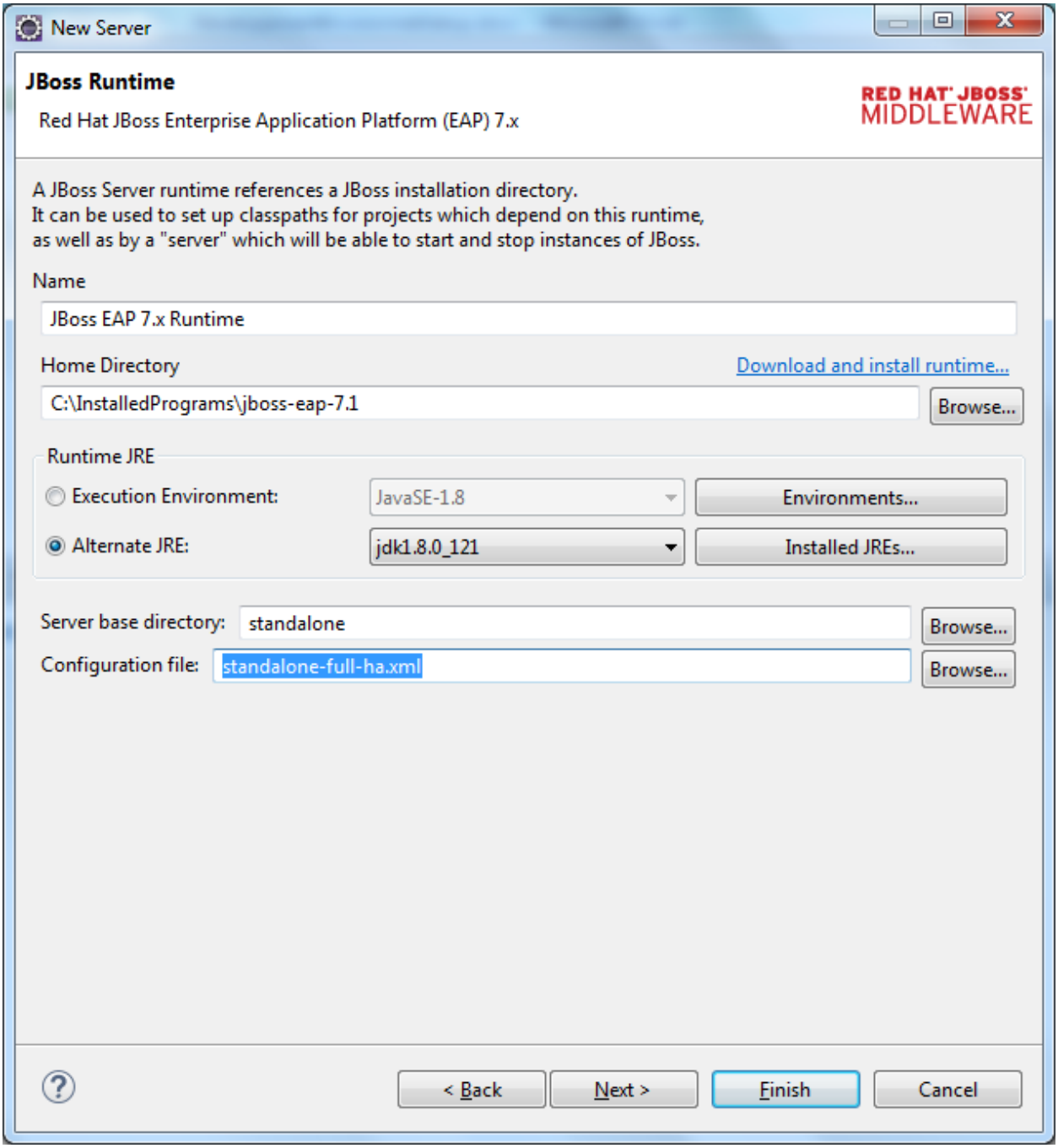

You would now see Red Hat JBoss Enterprise Application Platform 7.x as an option now. Click on the same and click Next



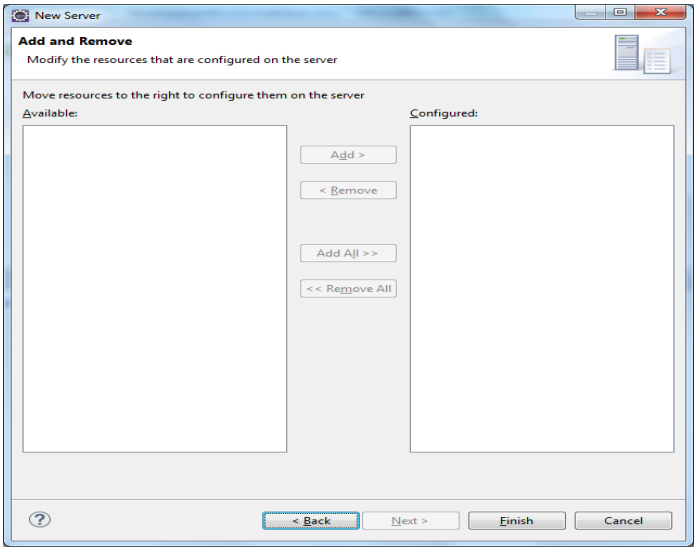
Click Next



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.



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. Set the JBoss server port

Modify the below property to set the desired port, for yugandhar msp default port is set to 8091

<socket-binding name="http" port="${jboss.http.port:8091}"/>

### b. 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)

Administrator: C:\Windows\system32\cmd.exe

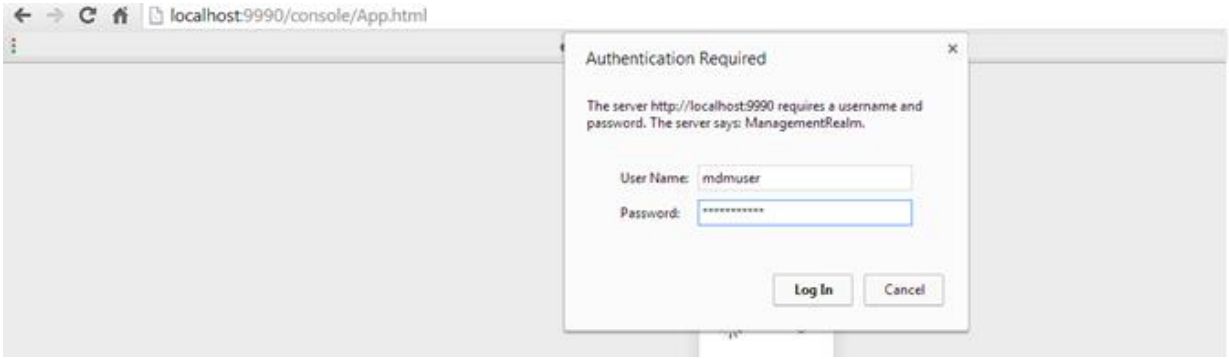
```
C:\InstalledPrograms\jboss-eap-7.1.0\bin>add-user

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, admi
n, 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\co
nfiguration\mgmt-users.properties'
Added user 'mdmuser' to file 'C:\InstalledPrograms\jboss-eap-7.1.0\domain\config
uration\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 pr
ocess?
e.g. for a slave host controller connecting to the master or for a Remoting conn
ection for server to server EJB calls.
yes/no? yes
To represent the user add the following to the server-identities definition <sec
ret value="VGVzdHB3ZF8xMjM=" />
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




### c. Create JBoss Datasource and Active MQ Queues

MDM Hub needs JBoss based datasource to connect to the database. The JNDI name of the datasource needs to be provided in application.properties file. We need to create either of the below Data source and uncomment the related data source in the properties

```
#spring.datasource.jndi-name=java:/YUGMDM_XAOracle11gDS
#spring.datasource.jndi-name=java:/YUGMDM_XAOracle12cDS
#spring.datasource.jndi-name=java:/YUGMDM_XAMariaDBDS
```

***Configure Oracle drivers***

Note- Either configure Oracle or MariaDB, no need to configure both databases.

To create Oracle data source in JBoss, You need to manually make some changes on the file system. Also make sure that ojdbc7.jar is downloaded on your system from oracle distribution site.

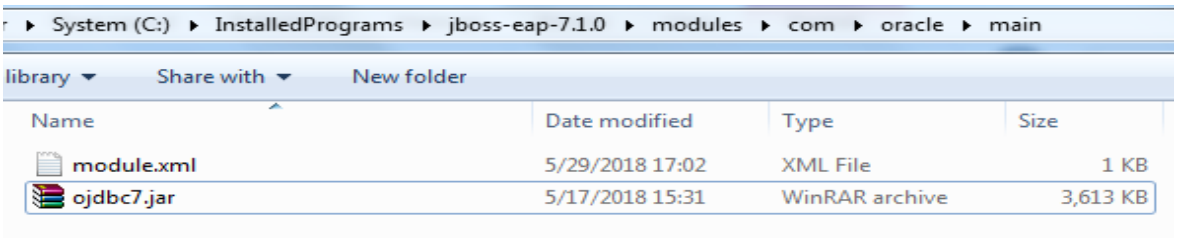
You may take help from the jboss sample configuration file provided in the <github Repository/resources\jbossconfig folder

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 ojdbc7.jar to $JBOSS_HOME/modules/com/oracle/main folder




iv. Create module.xml file.

v. Add the following content:

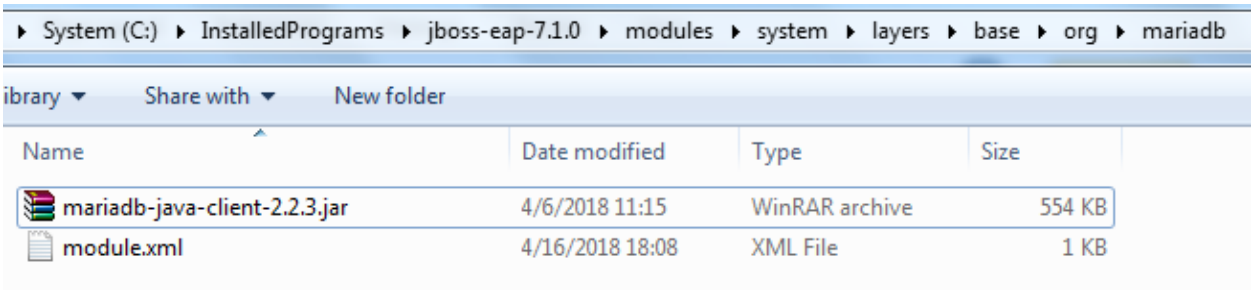
```
<?xml version="1.0" ?>
<module xmlns="urn:jboss:module:1.1" name="com.oracle">
    <properties>
      <property name="jboss.api" value="unsupported" />
   </properties>
    <resources>
      <resource-root path="ojdbc7.jar"/>
   </resources>
    <dependencies>
      <module name="javax.api" />
      <module name="javax.transaction.api" />
      <module name="javax.servlet.api" optional="true" />
   </dependencies>
</module>
```

### *Configure MariaDB drivers*

2. Create a new module for MariaDB driver.

   i. Create a folder hierarchy with path `$JBOSS_HOME/modules\system\layers\base\org\mariadb`

   ii. Copy mariadb-java-client-2.2.3.jar to `$JBOSS_HOME/modules\system\layers\base\org\mariadb` folder




   iii. Create module.xml file.

iv. Add the following content:

```
<?xml version="1.0" ?>
<module xmlns="urn:jboss:module:1.1" name="org.mariadb">
    <properties>
            <property name="jboss.api" value="unsupported" />
    </properties>
    <resources>
            <resource-root path="mariadb-java-client-2.2.3.jar"/>
    </resources>
    <dependencies>
            <module name="javax.api"/>
            <module name="javax.transaction.api"/>
            <module name="javax.servlet.api" optional="true" />
    </dependencies>
</module>
```

***Configure datasources in Jboss configuration file***

Configure data source in $JBOSS_HOME/standalone/configuration/ standalone-full-ha.xml

Add only one of the below configuration in <datasources> tag as per your database type, change the host and port as per database configuration.

| Database | Data Source property to enable | Data Source configuration |
|---|---|---|
| MariaDB | spring.datasource.jndi-name= java:/YUGMDM_XAMariaDBDS | <xa-datasource jndi-name="java:/YUGMDM_XAMariaDBDS" pool-name="YUGMDM_XAMariaDBDS" enabled="true" use-ccm="false" statistics-enabled="true"><br><xa-datasource-property name="ServerName">  <hostname><br></xa-datasource-property><br><xa-datasource-property name="DatabaseName">  yug_owner<br></xa-datasource-property><br><driver>mariadb</driver><br><xa-pool><br><min-pool-size>10</min-pool-size><br><initial-pool-size>10</initial-pool-size><br><max-pool-size>200</max-pool-size><br><allow-multiple-users>false</allow-multiple-users><br></xa-pool><br><security><br><security-domain>YUGMDM_XAMariaDBDS_UserSecurityDomain</security-domain><br></security><br><validation><br><valid-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnection<br><validate-on-match>true</validate-on-match><br><background-validation>false</background-validation><br><exception-sorter class- |

| | | |
|---|---|---|
| | | name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter<br></validation><br></xa-datasource> |
| Oracle 11g | spring.datasource.jndi-name=<br>java:/YUGMDM_XAOracle11gDS | <xa-datasource jndi-name="java:/YUGMDM_XAOracle11gDS" pool-name="YUGMDM_XAOracle11gDS" enabled="true" use-ccm="false" statistic enabled="true"><br><xa-datasource-property name="URL"><br>jdbc:oracle:thin:@<hostname>:<port>/serviceName e.g.<br>jdbc:oracle:thin:@localhost:1521:MDMDB<br></xa-datasource-property><br><driver>oracle</driver><br><xa-pool><br><min-pool-size>10</min-pool-size><br><initial-pool-size>10</initial-pool-size><br><max-pool-size>200</max-pool-size><br><allow-multiple-users>false</allow-multiple-users><br><is-same-rm-override>false</is-same-rm-override><br><no-tx-separate-pools>true</no-tx-separate-pools><br></xa-pool><br><security><br><security-domain>YUGMDM_XAOracle11gDS_UserSecurityDomain</security-domain><br></security><br><validation><br><valid-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnection<br><background-validation>true</background-validation><br><stale-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnection<br><exception-sorter class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter<br></validation><br></xa-datasource> |
| Oracle 12c | spring.datasource.jndi-name=<br>java:/YUGMDM_XAOracle12cDS | <xa-datasource jndi-name="java:/YUGMDM_XAOracle12cDS" pool-name="YUGMDM_XAOracle12cDS" enabled="true" use-ccm="false" statistic enabled="true"><br><xa-datasource-property name="URL"><br>jdbc:oracle:thin:@<hostname>:<port>/serviceName e.g.<br>jdbc:oracle:thin:@localhost:1521:MDMDB<br></xa-datasource-property><br><driver>oracle</driver><br><xa-pool><br><min-pool-size>1</min-pool-size><br><initial-pool-size>1</initial-pool-size><br><max-pool-size>200</max-pool-size><br><allow-multiple-users>false</allow-multiple-users><br><is-same-rm-override>false</is-same-rm-override><br><no-tx-separate-pools>true</no-tx-separate-pools><br></xa-pool><br><security><br><security-domain>YUGMDM_XAOracle12cDS_UserSecurityDomain</security-domain><br></security><br><validation><br><valid-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnection<br><background-validation>true</background-validation><br><stale-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnection |

| | | |
|---|---|---|
| | | <exception-sorter class-name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter<br></validation><br></xa-datasource> |

Add the drivers in drivers section

| Database type | Driver |
|---|---|
| MariaDB | <driver name="oracle" module="com.oracle"><br><xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-class><br></driver> |
| Oracle11g/12c | <driver name="mariadb" module="org.mariadb"><br><driver-class>org.mariadb.jdbc.Driver</driver-class><br><xa-datasource-class>org.mariadb.jdbc.MariaDbDataSource</xa-datasource-class><br></driver> |

Add one of the security domain as per your database configuration

| Database | Security configuration |
|---|---|
| MariaDB | <security-domain name="YUGMDM_XAMariaDBDS_UserSecurityDomain" cache-type="default"><br><authentication><br><login-module code="org.picketbox.datasource.security.SecureIdentityLoginModule" flag="required"><br><module-option name="username" value="YUG_OWNER"/><br><module-option name="password" value="37c5e49128f18b4ec2b659954c9b93d"/><br><module-option name="managedConnectionFactoryName" value="jboss.jca:service=LocalTxCM,name=YUGMDM_XAMariaDBDS"/><br></login-module><br></authentication><br></security-domain> |
| Oracle 11g | <security-domain name="YUGMDM_XAOracle11gDS_UserSecurityDomain" cache-type="default"><br><authentication><br><login-module code="org.picketbox.datasource.security.SecureIdentityLoginModule" flag="required"><br><module-option name="username" value="YUG_OWNER"/><br><module-option name="password" value="30c92153714c5ee91f0b0525519389a9"/><br><module-option name="managedConnectionFactoryName" value="jboss.jca:service=LocalTxCM,name=YUGMDM_XAOracle11gDS"/><br></login-module><br></authentication><br></security-domain> |
| Oracle 12c | <security-domain name="YUGMDM_XAOracle12cDS_UserSecurityDomain" cache-type="default"><br><authentication><br><login-module code="org.picketbox.datasource.security.SecureIdentityLoginModule" flag="required">> |

```
            <module-option name="username" value="YUG_OWNER"/>
            <module-option name="password"
value="37c5e49128f18b4ec2b659954c9b93d"/>
            <module-option name="managedConnectionFactoryName"
value="jboss.jca:service=LocalTxCM,name=YUGMDM_XAOracle11gDS"/>
        </login-module>
    </authentication>
</security-domain>
```

Encrypt the database password in the security domain is done through picket box. You may use the below command to encrypt the password.

java -cp $JBOSS_HOME\modules\system\layers\base\org\picketbox\main\picketbox-5.0.2.Final-redhat-1.jar org.picketbox.datasource.security.SecureIdentityLoginModule *<password String to encrypt>*

Change the jar name and version as per the jar available in your jboss modules.

3. Disable default JPA in jboss 7.1 by removing below entry altogether

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

4. 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">

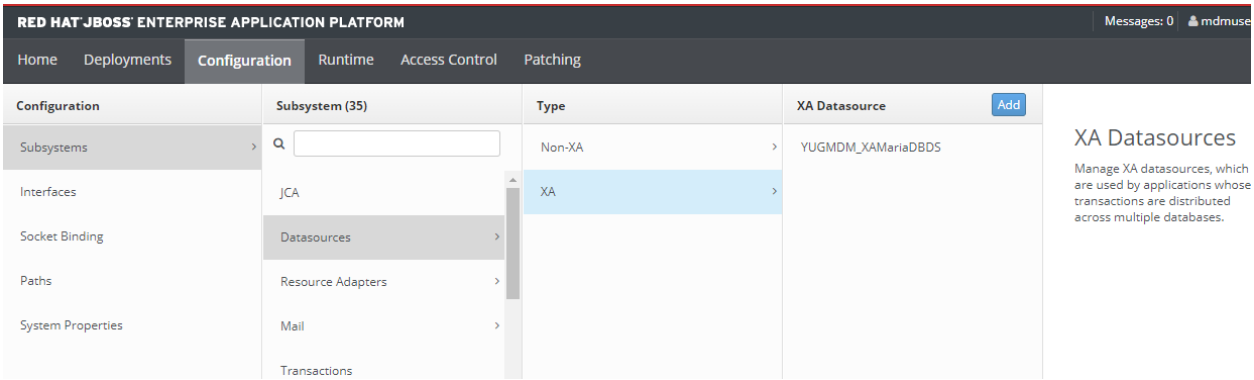
```
<server name="Yug">
            <security enabled="false"/>
            <management address="jms.queue.activemq.management1"/>
            <statistics enabled="true"/>
            <security-setting name="#">
                <role name="guest" send="true" consume="true" manage="true"/>
            </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>
```

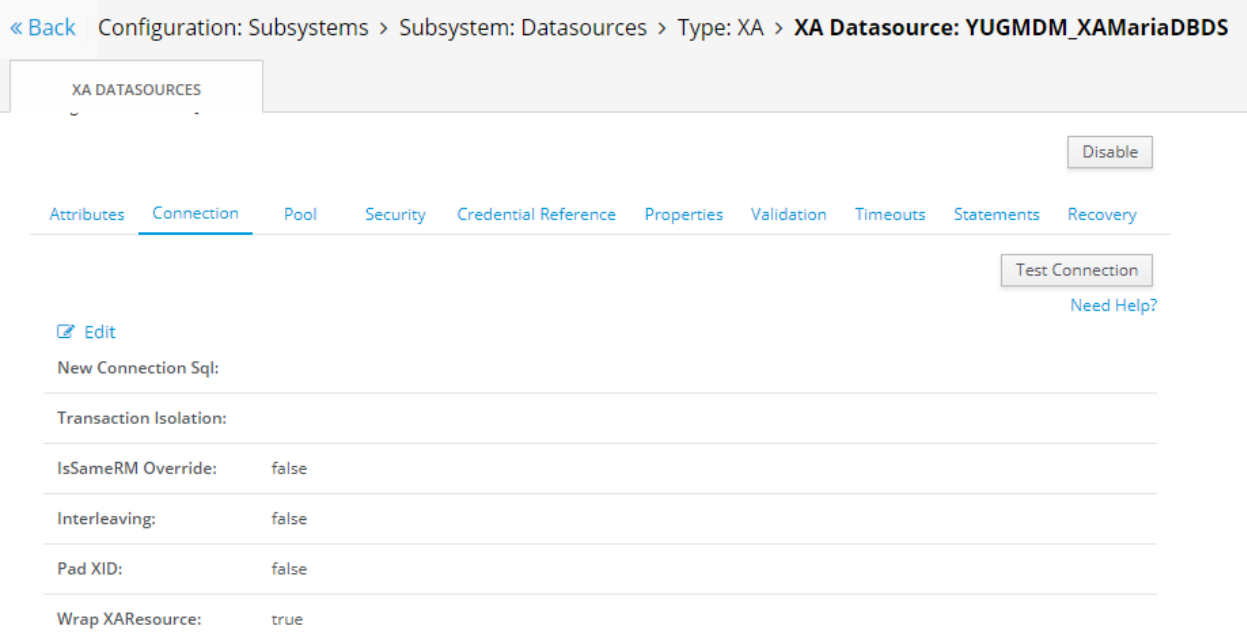
### *Verify the Datasource*

Login to jboss console using using the user created in 'add User to Access JBoss Console' step. For the purpose of this document we have configured MariaDB so only that datasource will be available on the server. If you use any other datasource then check the respective datasource under subsystem.

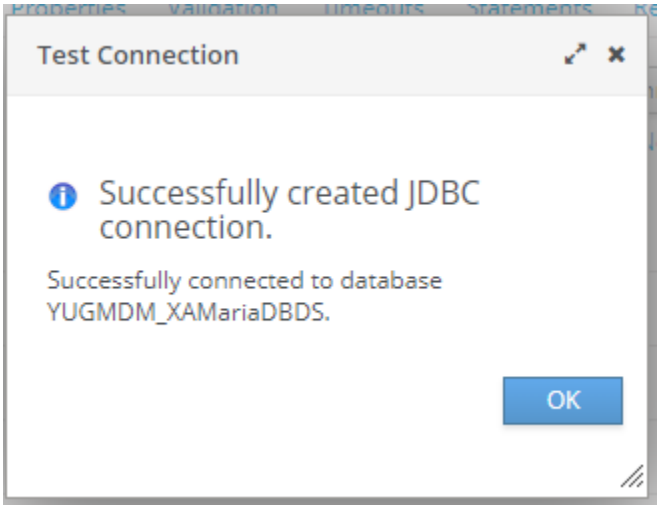
Navigate to datasources




Click View and then navigate to Connection tab.

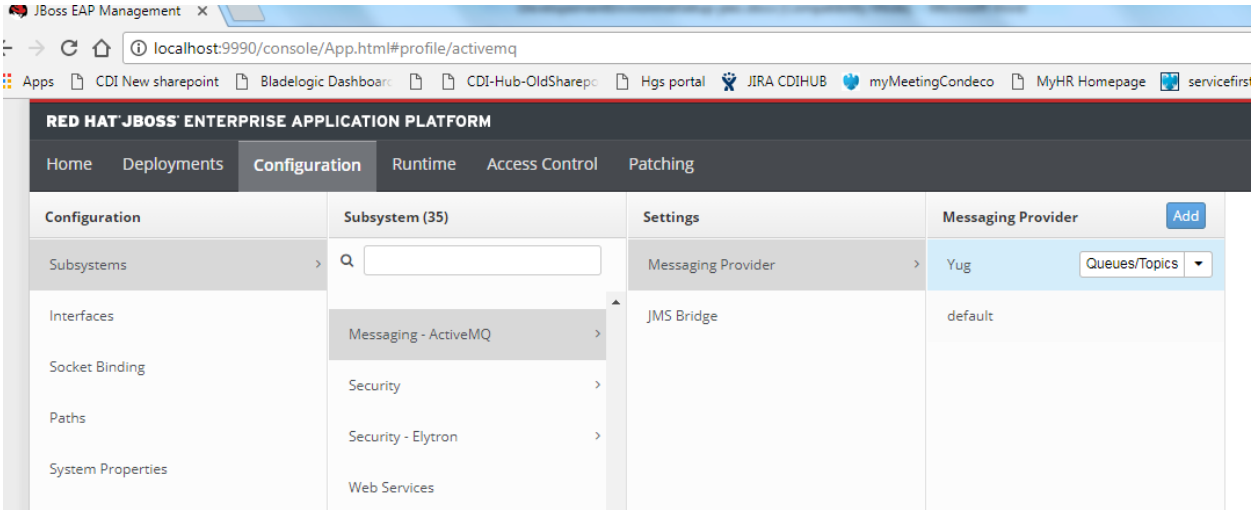



Click on the test connection and verify that the connection is successful

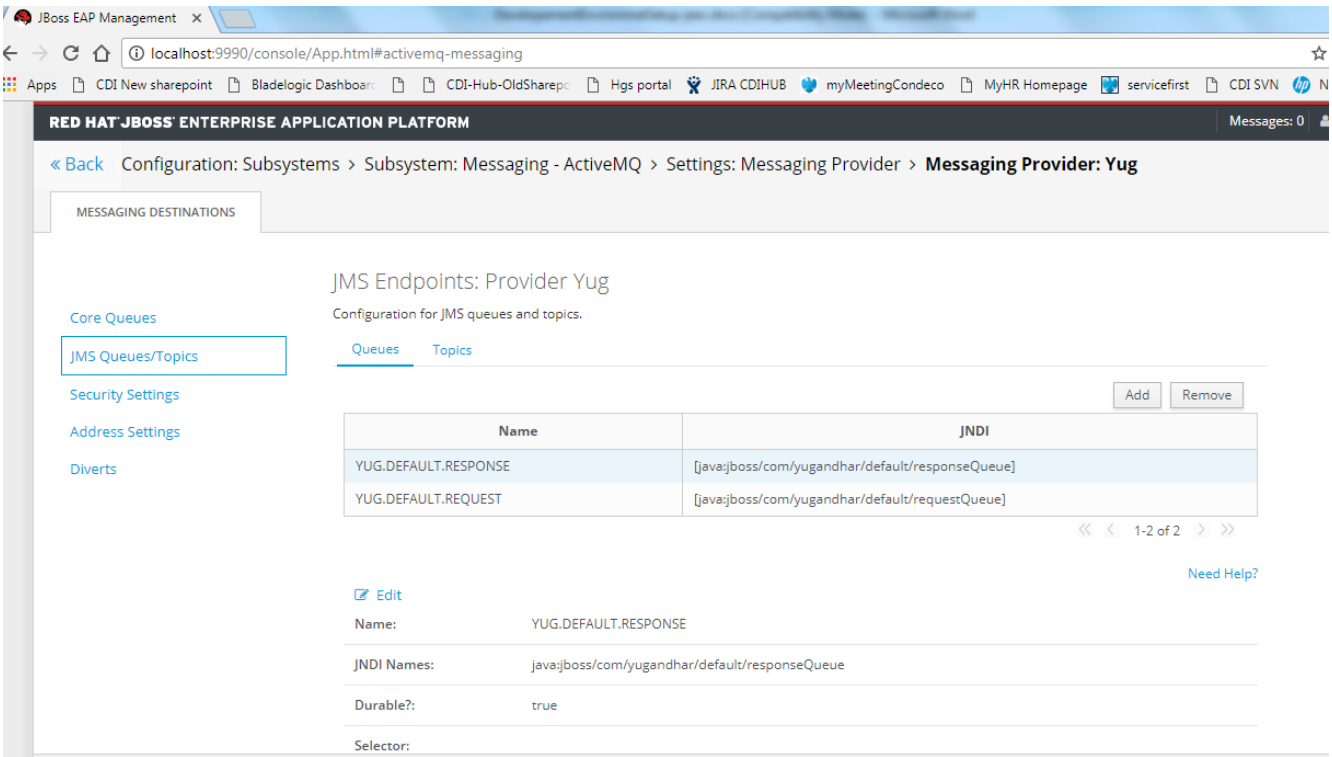



***Verify ActiveMQ***

Navigate to ActiveMQ subsystem → Yug, click Queues/Topics

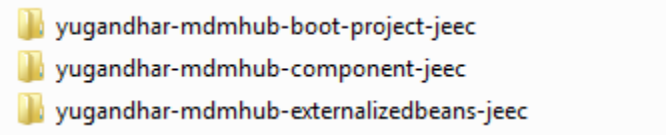



You should see the Active MQs as shown in below image

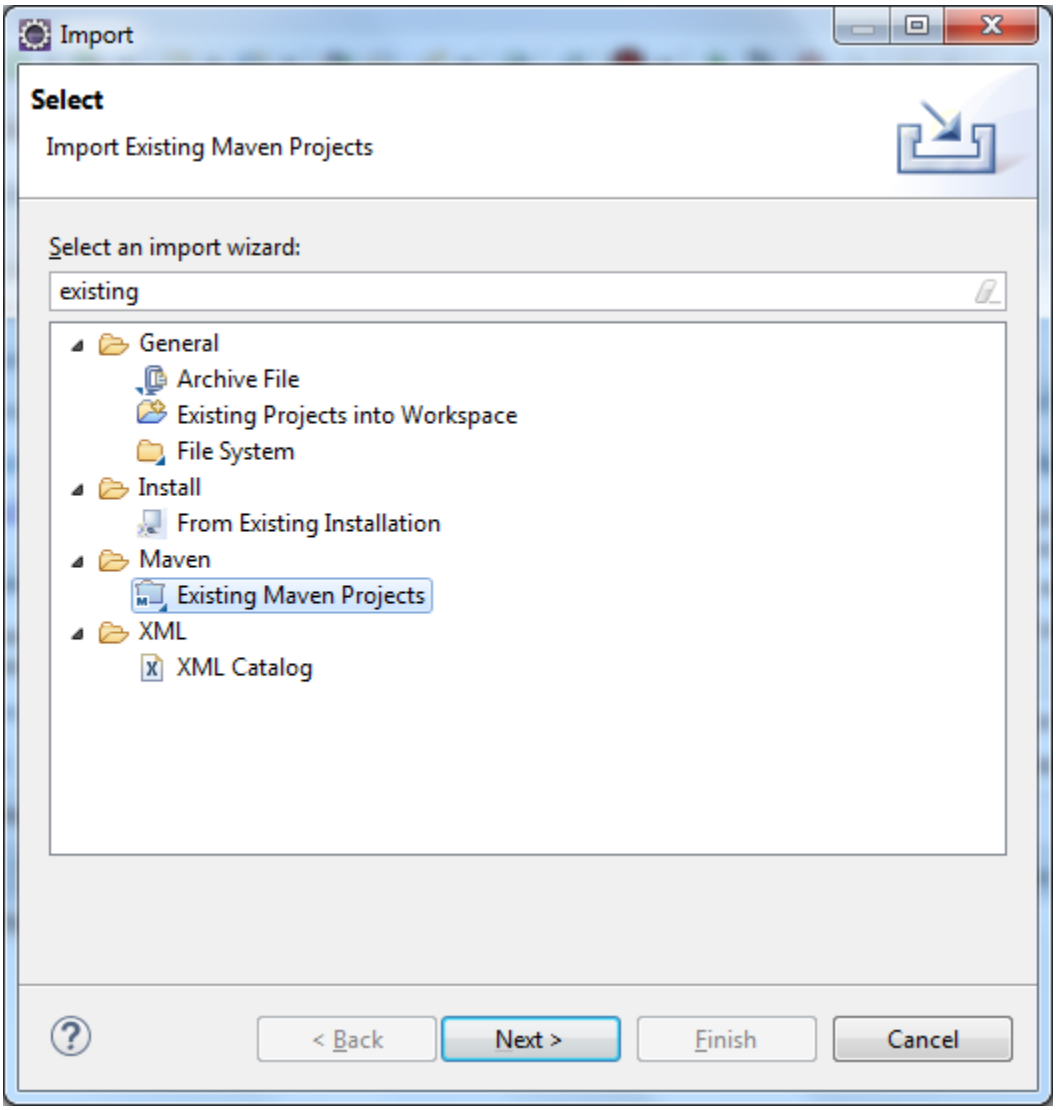


## Import Yugandhar MDM Hub java projects in the workspace

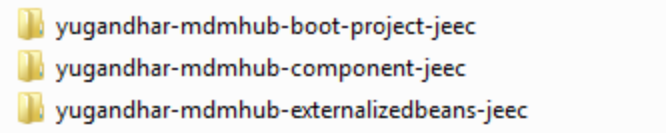
Download the yugandhar-open-mdmhub repository, go to the yugandhar-open-mdmhub-jeec directory and copy the below three projects in the workspace directory (e.g. C:\Workspaces\yugandhar-open-mdmhub) and import in Workspace




Go to File → Import Menu → Existing Maven Projects →

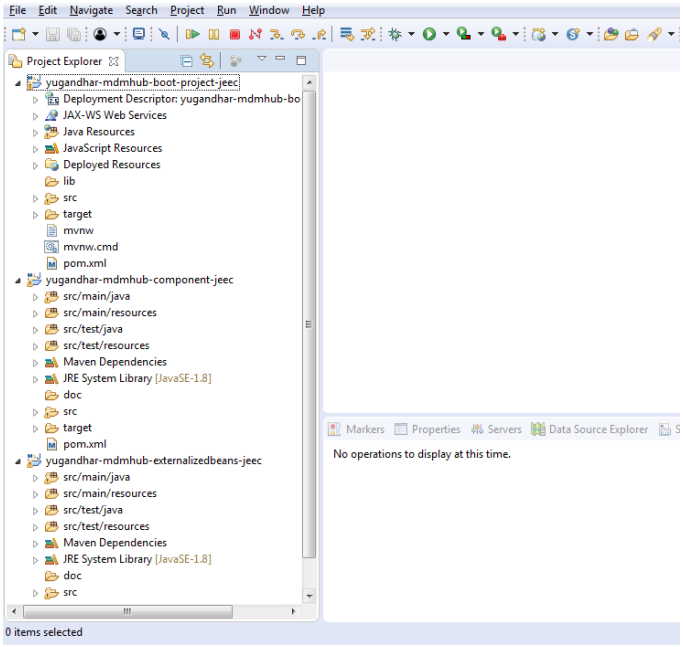



Select the below three projects which must be visible on the next screen and click Finish




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




Open MDM Hub does the logging to default folder C:/Yugandhar/logs so create this folder or change the log directory to the directory of your choice in /yugandhar-mdmhub-boot-project-jeec/src/main/resources/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">
```

## Properties file changes

There are below two properties files in the Microservice platform

### application.properties

The application properties file covers the below properties

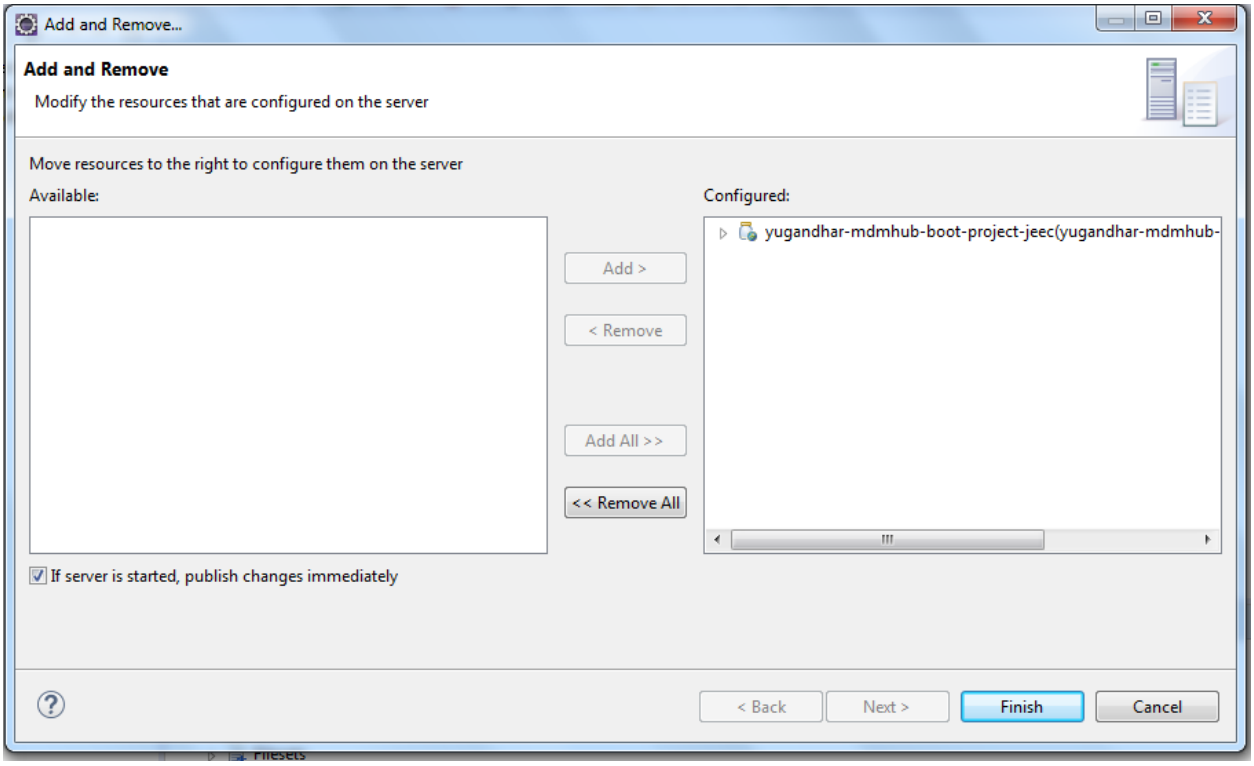
- **Springboot trace:** enable/disable the trace
- **Server port:** set the port number for the tomcat server
- **JPA:** If the generated ddl needs to be logged the enable the property spring.jpa.show-sql. By default this is enabled.
- **mariaDB specific settings:**
  ```
  #Enable both the below properties for mysql/MariaDB, if you are using oracle then comment both the properties.
  #spring.jpa.properties.hibernate.globally_quoted_identifiers=true
  #spring.jpa.database-platform=org.hibernate.dialect.MariaDB53Dialect
  ```
  Note- "`spring.jpa.properties.hibernate.globally_quoted_identifiers=true`" with Oracle database may result in errors.
- **Oracle specific settings:**
  Oracle Specifc configuration, use 10g dialect for Oracle 11g database else use 12c
  ```
  #spring.jpa.database-platform=org.hibernate.dialect.Oracle10gDialect
  #spring.jpa.database-platform=org.hibernate.dialect.Oracle12cDialect
  ```
- **Logging:** Logback configuration
- **JTA:** Atomikos is the default JTA provider being used by Yugandhar MDM Hub, change the properties as needed.
- **Ehcache:** ehcache properties
- **Json:** json parser related properties
- **Active mq:** active MQ properties
- **Actuator:** spring boot actuator properties
- **Eureka integration:** Eureka integration properties, by default it's disabled.

### yugandhar-mdmhub-app.properties

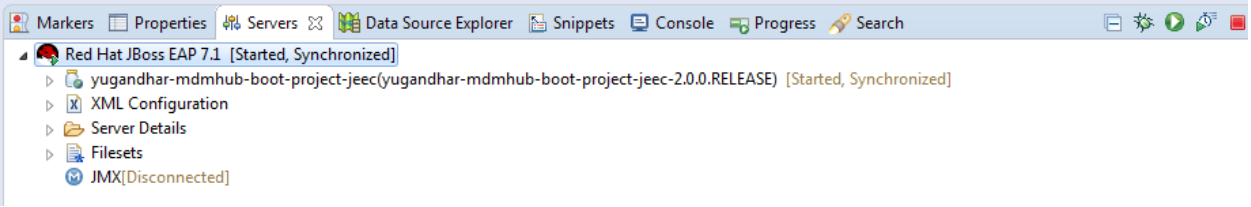
The yugandhar-mdmhub-app.properties file is custom properties file not used by JEEC version.

## Running the application

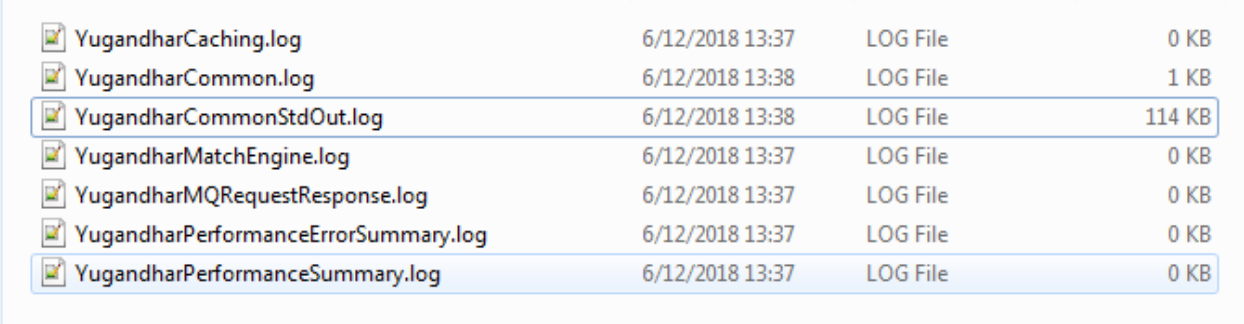
Right click the jboss server in the Servers tab and add the MSP application, click finish.




Verify the application is started in the servers tab




Check the logs files getting generated in the folder as mentioned in the yugandhar_logback.xml



| | | | |
|---|---|---|---|
| YugandharCaching.log | 6/12/2018 13:37 | LOG File | 0 KB |
| YugandharCommon.log | 6/12/2018 13:38 | LOG File | 1 KB |
| YugandharCommonStdOut.log | 6/12/2018 13:38 | LOG File | 114 KB |
| YugandharMatchEngine.log | 6/12/2018 13:37 | LOG File | 0 KB |
| YugandharMQRequestResponse.log | 6/12/2018 13:37 | LOG File | 0 KB |
| YugandharPerformanceErrorSummary.log | 6/12/2018 13:37 | LOG File | 0 KB |
| YugandharPerformanceSummary.log | 6/12/2018 13:37 | LOG File | 0 KB |

## TEST With SOAPUI

### Test JSON message

The rest url is as below.

http://localhost:8091/yugandhar-mdmhub-boot-project-jeec-2.0.0.RELEASE/rest/YugandharRequestProcessor

Yugandhar MDM HUB uses the port 8091 as default; you may change it through jboss configuration.

Sample json message

```
{"txnHeader": {
                "requesterLanguage": "1",
                "userName": "admin",
                "userRole": "admin",
                "txnMessageId": "12312311115999",
                "transactionServiceName": "findAllRefCountryIsoByLanguageCodeBase"
        },
        "txnPayload": {
                "paginationIndexOfCurrentSlice": 2,
                "paginationPageSize": 25,
                        "refCountryIsoDO":      {
                "configLanguageCodeKey": "1"
    }

        }

}
```

For the soap xml message, add the below headers for the request xmls

| Header | Value | use |
|---|---|---|
| Accept | application/json | This header tells yugandhar rest controller that the response must be sent in json format. |
| Content-Type | application/json | This header tells yugandhar rest controller that the request message is of type json |

Create 'New REST service from URI' in the soapui project, and execute it with attached json message

Request 1

Method: POST
Endpoint: http://localhost:8091
Resource: /yugandhar-mdmhub-boot-project-jeec-2.0.0.RELEASE/rest/YugandharRequestProcessor

Request | Raw

| Name | Value | Style | Level |
|---|---|---|---|

Required: Sets if parameter is required

Media Type: application/json  Post QueryString


```
        "userRole": "admin",
        "txnMessageId": "12312311115999",
        "transactionServiceName": "findAllRefCountryIsoByLanguageCodeBase"
    },
    "txnPayload": {
        "paginationIndexOfCurrentSlice": 2,
        "paginationPageSize": 25,
            "refCountryIsoDO":        {
        "configLanguageCodeKey": "1"
     }
```


| Header | Value |
|---|---|
| Accept | application/json |
| Content-Type | application/json |

Auth | Headers (2) | Attachments (0) | Representations (1) | JMS Headers | JMS Property (0)

Check the response as SUCCESS

Request 1

Method: POST
Endpoint: http://localhost:8091
Resource: /yugandhar-mdmhub-boot-project-jeec-2.0.0.RELEASE/rest/YugandharRequestProcessor
Parameters

Request | Raw

| Name | Value | Style | Level |
|---|---|---|---|

Media Type: application/ison
Post QueryStrin


```
        "userRole": "admin",
        "txnMessageId": "12312311115999",
        "transactionServiceName": "findAllRe
    },
    "txnPayload": {
        "paginationIndexOfCurrentSlice": 2,
        "paginationPageSize": 25,
            "refCountryIsoDO":        {
        "configLanguageCodeKey": "1"
    }
```


| Header | Value |
|---|---|
| Accept | application/json |
| Content-Type | application/json |

A... | Head... | Attachm... | Represent... | JMS H... | JMS Pro...

XML | JSON | HTML | Raw


```
1  {
2      "responseCode": "SUCCESS",
3      "txnHeader":    {
4          "requesterLanguage": "1",
5          "userName": "admin",
6          "userRole": "admin",
7          "txnMessageId": "12312311115999",
8          "transactionServiceName": "findAllRefCountryIsoByLanguageCod
9          "totalExecutionTimeMillies": "5644"
10     },
11     "txnPayload":    {
12         "paginationIndexOfCurrentSlice": 2,
13         "paginationPageSize": 25,
14         "paginationInfoElementsOnCurrentSlice": 25,
15         "paginationInfoTotalElements": 247,
16         "paginationInfoTotalPages": 10,
17         "refCountryIsoDOList":       [
18                 {
19             "idPk": "144",
20             "version": 0,
21             "createdTs": "2018-04-16T12.57.17.000+0000",
22             "updatedTs": "2018-04-16T12.57.17.000+0000",
23             "updatedByUser": "admin",
24             "updatedByTxnId": "000000",
25             "configLanguageCodeKey": "1",
```


Headers (7) | Attachments (0) | SSL Info | Representations (1) | Schema (conflicts) | JMS (0)

**Test XML message**

The rest url is as below.

http://localhost:8091/yugandhar-mdmhub-boot-project-jeec-2.0.0.RELEASE/rest/YugandharRequestProcessor

Yugandhar MDM HUB uses the port 8091 as default; you may change it through jboss configuration.

For the soap xml message, add the below headers for the request xmls

| Header | Value | use |
|---|---|---|
| Accept | application/xml | This header tells yugandhar rest controller that the response must be sent in xml format. |
| Content-Type | application/xml | This header tells yugandhar rest controller that the request message is of type xml |

Sample XML message:

```
<TxnTransferObj>
  <txnHeader>
    <requesterLanguage>1</requesterLanguage>
    <userName>admin</userName>
    <userRole>admin</userRole>
    <txnMessageId>12312311115999</txnMessageId>

<transactionServiceName>findAllRefCountryIsoByLanguageCodeBase</transactionServiceN
ame>
  </txnHeader>
  <txnPayload>
    <paginationIndexOfCurrentSlice>1</paginationIndexOfCurrentSlice>
    <paginationPageSize>50</paginationPageSize>
      <refCountryIsoDO>
        <configLanguageCodeKey>1</configLanguageCodeKey>
      </refCountryIsoDO>
  </txnPayload>
</TxnTransferObj>
```

Request 1

Method: POST
Endpoint: http://localhost:8091
Resource: /yugandhar-mdmhub-boot-project-jeec-2.0.0.RELEASE/rest/YugandharRequestProcessor

Request | Raw

| Name | Value | Style | Level |
|---|---|---|---|

Required: Sets if parameter is required

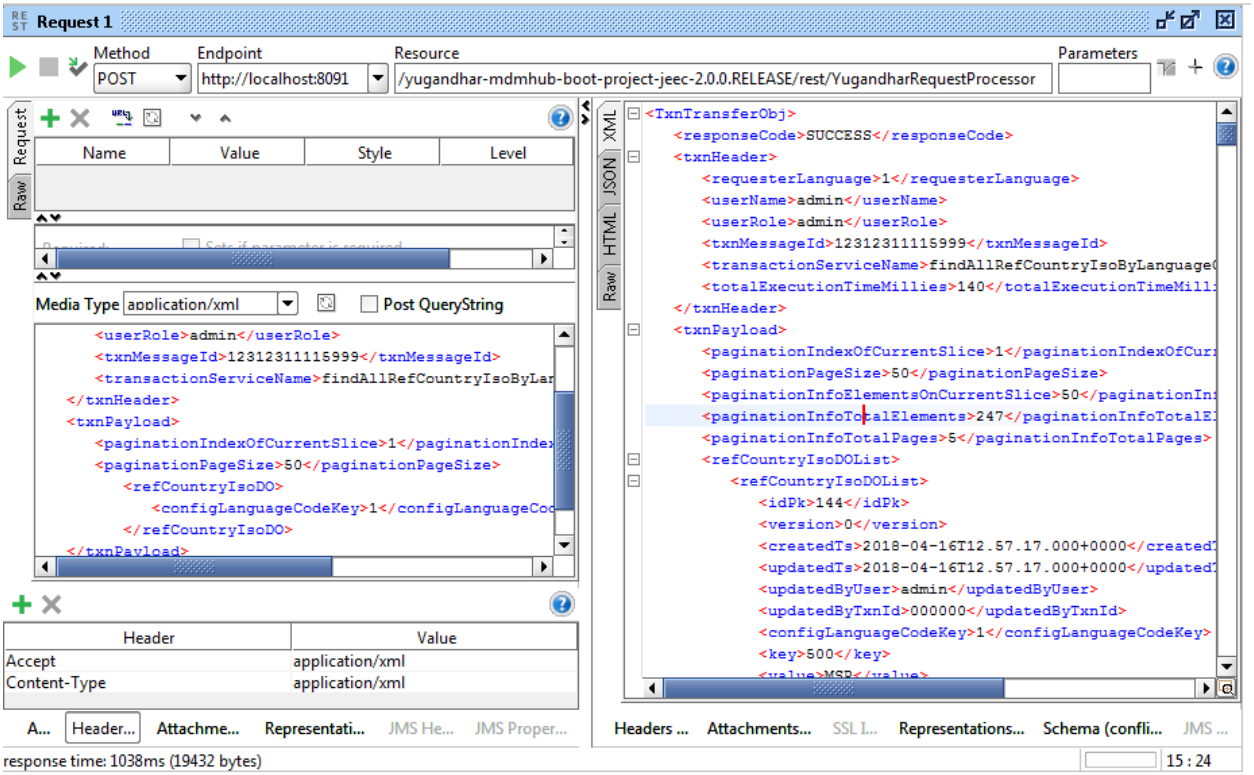
Media Type: application/xml  Post QueryString


```
      <txnMessageId>12312311115999</txnMessageId>
      <transactionServiceName>findAllRefCountryIsoByLanguageCodeBase</transactionServiceName>
   </txnHeader>
   <txnPayload>
      <paginationIndexOfCurrentSlice>1</paginationIndexOfCurrentSlice>
      <paginationPageSize>50</paginationPageSize>
         <refCountryIsoDO>
            <configLanguageCodeKey>1</configLanguageCodeKey>
         </refCountryIsoDO>
   </txnPayload>
</TxnTransferObj>
```


| Header | Value |
|---|---|
| Content-Type | application/xml |
| Accept | application/xml |

Auth | Headers (2) | Attachments (0) | Representations (1) | JMS Headers | JMS Property (0)

Check the success response.




This certifies your workspace.

You may test a few more transactions like createLegalentity, createLeAccount etc for which sample messages are provided in the resources/Testing folder.

Go ahead with Development and customization guide, API Transaction reference guide and Code generation guide to understand more.