

JAVA DEVELOPER HANDBOOK

JAVA DEVELOPER HANDBOOK

- From Dummy to Pro**
 - Java, Maven, Spring, DB, VM, Docker**
-

Hongsong Zhou

Copyright ©2016 by Hongsong Zhou. All rights reserved.

Published by Hongsong Zhou, Dallas, Texas.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Hongsong Zhou, zhouhongsong@hotmail.com.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact zhouhongsong@hotmail.com.

Hongsong Zhou also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic format.

Library of Congress Cataloging-in-Publication Data:

Beginner Developer Handbook / Hongsong Zhou . . . [et al.].
p. cm.—(Hongsong Zhou series in java technology)
“Hongsong Zhou-JavaTechnology.”
Includes bibliographical references and index.
ISBN 0-000-00000-0 (pbk.)
1. Java—Technology. 2. Infomration Technology
technologies—Information—Java. I. Zhou, Hongsong. II. Java.

XX00.0.S000 2016
000.0'00—xx00 2016080500
Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

*To my wife, my daughter,
and my son*

CONTENTS IN BRIEF

PART I INSTALLATION AND SIMPLE PROJECTS

1 Installation and Configuration	3
2 Simple Projects	47
3 Spring MVC	85
4 More Spring	97

PART II VIRTUAL MACHINE

5 Virtual Machine	119
6 Linux on Virtual Machine	131
7 Connect to Virtual Machine	161
8 Oracle on Virtual Machine	177

PART III DOCKER

9 Ubuntu Linux on Virtual Machine	199
10 Docker on Ubuntu Linux	223
11 MySQL on Docker	233

PART IV DATABASE

12 Database	243
--------------------	------------

13 JDBC	277
14 Spring DAO	299
15 Spring ORM	325

PART V ADVANCED PROJECTS

16 Web Service	357
-----------------------	------------

PART VI APPENDIX

CONTENTS

Foreword	xix
Preface	xxi
Acknowledgments	xxiii

PART I INSTALLATION AND SIMPLE PROJECTS

1 Installation and Configuration	3
1.1 Java	3
1.1.1 Download Java	3
1.1.2 Install Java - With Administrator Rights	3
1.1.3 Install Java - Without Administrator Rights	7
1.1.4 Configure Java	14
1.2 Java - Mac	15
1.2.1 Download Java	15
1.2.2 Install Java	15
1.2.3 Verify Java Installation	17
1.2.4 Setup JAVA_HOME	18
1.2.5 Uninstall Java	18
1.3 Tomcat	19
1.3.1 Download Tomcat	19
1.3.2 Extract Tomcat	19

ix

1.3.3	Setup Tomcat Home	19
1.4	Tomcat - Mac	19
1.4.1	Download Tomcat	19
1.4.2	Extract Tomcat	19
1.4.3	Start Tomcat	20
1.4.4	Test Tomcat	20
1.4.5	Stop Tomcat	20
1.5	Maven	21
1.5.1	Download Maven	21
1.5.2	Extract Maven	21
1.5.3	Setup Maven Environment	21
1.5.4	Configure Maven Settings	21
1.5.5	Test Maven	22
1.6	Maven - Mac	22
1.6.1	Download Maven	22
1.6.2	Extract Maven	22
1.6.3	Setup Maven Environment	22
1.6.4	Configure Maven Settings	23
1.6.5	Test Maven	23
1.6.6	Maven Help	23
1.7	Eclipse	24
1.7.1	Eclipse - Windows	24
1.7.2	Eclipse - Mac	29
1.7.3	Configure Eclipse - Preferences	35
1.8	Maven Central	45
2	Simple Projects	47
2.1	Simple maven project - app-hello-world	47
2.1.1	Create project using command line	47
2.1.2	Create project using Eclipse	53
2.2	Simple web maven project - hello-world	55
2.2.1	Create project	55
2.2.2	Make this project an eclipse project	55
2.2.3	Update JDK version	56
2.2.4	Import this project to eclipse	57
2.2.5	Add Tomcat server to eclipse	59
2.2.6	Configure Tomcat server	61
2.2.7	Add hello-world to Tomcat server	64
2.2.8	Run hello-world on Tomcat server	65
2.3	Simple spring maven project - app-hello-spring	68
2.3.1	Create project using command line	68
2.3.2	Modify pom.xml	68

2.3.3	Make this project an eclipse project	69
2.3.4	Import this project to eclipse	69
2.3.5	Add a simple java bean	69
2.3.6	Configure spring bean	72
2.3.7	Instantiating a container	77
2.3.8	Run App	78
2.4	Simple spring maven project - app-hello-spring-annotation	79
2.4.1	Create project using command line	79
2.4.2	Modify pom.xml	79
2.4.3	Make this project an eclipse project	79
2.4.4	Import this project to eclipse	80
2.4.5	Add a simple java bean	80
2.4.6	Configure spring bean	80
2.4.7	Instantiating a container	80
2.4.8	Run App	81
2.4.9	Using JSR 330 Standard Annotations	81
2.5	Simple spring maven project - app-hello-spring-appconfig	81
2.5.1	Create project using command line	81
2.5.2	Modify pom.xml	81
2.5.3	Make this project an eclipse project	82
2.5.4	Import this project to eclipse	82
2.5.5	Add a simple java bean	83
2.5.6	Configure spring bean - NO xml anymore	83
2.5.7	Instantiating a container	83
2.5.8	Run App	84
3	Spring MVC	85
3.1	Spring MVC - hello-spring	85
3.1.1	Create project using command line	85
3.1.2	Modify pom.xml	85
3.1.3	Make this project an eclipse project	86
3.1.4	Import this project to eclipse	86
3.1.5	Controller	86
3.1.6	JSP Views	87
3.1.7	Spring servlet configuration	87
3.1.8	Integrate web application with Spring - web.xml	88
3.1.9	Build the project - command line	88
3.1.10	Build the project - eclipse	88
3.1.11	Run	91
3.2	Separate Spring configuration xml files	92
3.2.1	Create project using command line	92
3.2.2	Modify pom.xml	92

3.2.3	Make this project an eclipse project	93
3.2.4	Import this project to eclipse	93
3.2.5	Controller	93
3.2.6	JSP Views	94
3.2.7	Spring resources configuration	94
3.2.8	Spring servlet configuration	94
3.2.9	Integrate web application with Spring - web.xml	95
3.2.10	Build the project	95
3.2.11	Run	95
3.2.12	Add a controller which returns null view name to project	96
3.2.13	Build the project	96
3.2.14	Run	96
4	More Spring	97
4.1	Spring JSON - hello-spring-json	97
4.1.1	Create project using command line	97
4.1.2	Modify pom.xml	97
4.1.3	Make this project an eclipse project	98
4.1.4	Import this project to eclipse	98
4.1.5	Model	99
4.1.6	Controller	99
4.1.7	Spring servlet configuration	100
4.1.8	Integrate web application with Spring - web.xml	100
4.1.9	Build the project	100
4.1.10	Run	101
4.2	Spring Security - hello-spring-security	101
4.2.1	Create project using command line	101
4.2.2	Modify pom.xml	101
4.2.3	Make this project an eclipse project	102
4.2.4	Import this project to eclipse	102
4.2.5	Controller	102
4.2.6	JSP Views	103
4.2.7	Spring Security Namespace Configuration	103
4.2.8	Spring servlet configuration	104
4.2.9	Integrate web application with Spring - web.xml	104
4.2.10	Build the project	105
4.2.11	Run	105
4.3	Spring Batch - app-hello-spring-batch	106
4.3.1	Create project using command line	106
4.3.2	Modify pom.xml	107
4.3.3	Make this project an eclipse project	107
4.3.4	Import this project to eclipse	107

4.3.5	Tasklet	108
4.3.6	Configure spring bean	108
4.3.7	Configure job xml	109
4.3.8	Build the project	110
4.3.9	Run App in command line	110
4.3.10	Run App - in eclipse	111

PART II VIRTUAL MACHINE

5	Virtual Machine	119
5.1	Hardware	119
5.2	Oracle VM VirtualBox	119
5.2.1	Download Oracle VM VirtualBox	120
5.2.2	Install Oracle VM VirtualBox	120
5.2.3	Configurate Oracle VM VirtualBox	123
5.3	Oracle VM VirtualBox - Mac	125
5.3.1	Download Oracle VM VirtualBox	125
5.3.2	Install Oracle VM VirtualBox	125
5.3.3	Install Oracle VM VirtualBox Extension Pack	127
5.3.4	Configurate Oracle VM VirtualBox	128
6	Linux on Virtual Machine	131
6.1	Linux	131
6.1.1	Register for Download	131
6.1.2	Download Oracle Linux - Windows	131
6.1.3	Download Oracle Linux - Mac	136
6.1.4	Install Oracle Linux - Create VM	140
6.1.5	Install Oracle Linux - From Oracle iso	143
6.2	Install Linux Guest Additions	151
6.2.1	Use CD-ROM in VM Window	151
6.2.2	Mount CD-ROM in OL7	152
6.2.3	Change yum repo to use local CD-ROM	152
6.2.4	Prepare for Guest Additions	152
6.2.5	Install Guest Additions	154
6.3	Shared folders	155
6.3.1	Shared folders - Windows	155
6.3.2	Shared folders - Mac	157
6.4	Network	160
7	Connect to Virtual Machine	161
7.1	Connect to VM	161
7.1.1	VM Network Setting	161

7.1.2	Connect to VM - Windows	164
7.1.3	Connect to VM - Mac	168
7.2	Backup and Restore VM	171
7.2.1	Snapshots	171
7.2.2	Export Appliance	172
7.2.3	Import Appliance	175
8	Oracle on Virtual Machine	177
8.1	Download Oracle	177
8.2	Copy Oracle Database files to VMBox OL7	178
8.2.1	Using WinSCP to copy files - Windows	178
8.2.2	Copy files - Mac	178
8.2.3	Using VMBox shared folder to copy files	179
8.3	Network	179
8.4	Install Zip and Unzip	181
8.5	Unpack Files	181
8.6	Oracle Installation Prerequisites	181
8.6.1	Automatic Setup	181
8.6.2	Manual Setup	182
8.7	Additional Setup	185
8.8	Installation	187
8.8.1	Inventory File	187
8.8.2	Response Files	187
8.8.3	Install Oracle	188
8.9	Post Installation	189
8.10	TNS	190
8.11	Set Oracle as Service	191
8.12	Test Oracle	192
8.13	Map Oracle VM to host machine	193
8.14	Connect to Oracle using SQL Developer	194
PART III DOCKER		
9	Ubuntu Linux on Virtual Machine	199
9.1	Linux	199
9.1.1	Download Ubuntu Linux	199
9.1.2	Install Ubuntu Linux - Create VM	199
9.1.3	Ubuntu VM Configuration - Optional	203
9.1.4	Install Ubuntu Linux - From Ubuntu iso	204
9.2	Install Linux Guest Additions	216
9.3	Shared folders	217
9.4	Network	221

9.5	Connect to VM - VM Network Setting	222
10	Docker on Ubuntu Linux	223
10.1	Prerequisites	223
10.1.1	Ubuntu kernel	223
10.1.2	Update apt sources	223
10.1.3	Recommend package	225
10.2	Install Docker	226
10.3	Control and configure Docker with systemd	228
10.3.1	Custom Docker daemon options	228
10.3.2	HTTP proxy	228
10.4	Docker - Hello World	230
10.5	Docker group	230
10.6	Upgrade Docker	231
10.7	Uninstallation	232
11	MySQL on Docker	233
11.1	MySQL official image	233
11.2	MySQL Port Forwarding	234
11.3	Run MySQL	236
11.4	Connect to MySQL	237
11.5	Remove MySQL Container	238

PART IV DATABASE

12	Database	243
12.1	Oracle XE - Windows	243
12.1.1	Download Oracle XE	243
12.1.2	Install Oracle XE	243
12.1.3	Making Oracle Database XE Available to Remote Clients	246
12.1.4	Start Oracle XE	246
12.1.5	Connect to Oracle XE	247
12.1.6	Stop Oracle XE	247
12.1.7	Stop Oracle XE Service	248
12.1.8	Changing Listener Port Numbers	249
12.2	MySQL	254
12.2.1	MySQL - Windows	254
12.2.2	MySQL - Mac	260
12.3	SQLDeveloper	266
12.3.1	Download SQLDeveloper	266
12.3.2	Install SQLDeveloper	266
12.3.3	Start SQLDeveloper	266

12.3.4	Configurate Proxy	267
12.3.5	Configure JDBC Driver in SQL Developer	268
12.3.6	Connect to Oracle XE through SQL Developer	270
12.3.7	Connect to MySQL through SQL Developer	273
12.4	HyperSQL (HSQLDB)	274
12.4.1	Download HyperSQL	274
12.4.2	Install HyperSQL	274
12.4.3	Start HyperSQL	274
12.4.4	Connect to HyperSQL	274
13	JDBC	277
13.1	Quick Start	277
13.1.1	Connect to MySQL	277
13.1.2	Connect to Oracle	279
13.1.3	Connect to HyperSQL	283
13.2	Statement	286
13.3	PreparedStatement	289
13.4	Stored Procedure	293
13.5	Transaction	295
14	Spring DAO	299
14.1	JDBC	299
14.1.1	Create project using command line	299
14.1.2	Modify pom.xml	299
14.1.3	Make this project an eclipse project	300
14.1.4	Import this project to eclipse	301
14.1.5	Add a java model class	301
14.1.6	Add java dao classes	301
14.1.7	Configure spring bean	305
14.1.8	Instantiating a container	305
14.1.9	Run App	306
14.2	Spring Framework JDBC	307
14.2.1	JdbcTemplate	307
14.2.2	NamedParameterJdbcTemplate	311
14.3	Embedded Database	316
14.3.1	Create project using command line	316
14.3.2	Modify pom.xml	316
14.3.3	Make this project an eclipse project	317
14.3.4	Import this project to eclipse	317
14.3.5	Add a java model class	318
14.3.6	Add java dao classes	318
14.3.7	Configure spring bean	321

14.3.8	SQL	322
14.3.9	Instantiating a container	322
14.3.10	Run App - HSQL	323
14.3.11	Run App - H2 and Derby	323
15	Spring ORM	325
15.1	MyBatis	325
15.1.1	Create project using command line	325
15.1.2	Modify pom.xml	325
15.1.3	Make this project an eclipse project	327
15.1.4	Import this project to eclipse	327
15.1.5	Add a java model class	327
15.1.6	Add java dao classes	328
15.1.7	Configure spring bean	328
15.1.8	SQL	330
15.1.9	Instantiating a container	330
15.1.10	Run App	331
15.2	Hibernate	331
15.2.1	Create project using command line	331
15.2.2	Modify pom.xml	331
15.2.3	Make this project an eclipse project	332
15.2.4	Import this project to eclipse	332
15.2.5	Add a java model class	333
15.2.6	Add java dao classes	333
15.2.7	Configure spring bean	335
15.2.8	SQL	336
15.2.9	Instantiating a container	336
15.2.10	Run App	337
15.3	JPA	338
15.3.1	JPA	338
15.3.2	JPA - Spring Data	346

PART V ADVANCED PROJECTS

16	Web Service	357
16.1	REST	357
16.1.1	REST Service - Using Jersey	357
16.1.2	REST Client - Using Java URL	362
16.1.3	REST Client - Using Apache HTTP Client	365
16.1.4	REST Client - Using Jersey Client	368
16.1.5	REST Service - Using Jersey2 - Hello	371
16.1.6	REST Client - Using Jersey2 - Hello	374

16.1.7	REST Service - Using Jersey2 - Employee	377
16.1.8	REST Client - Using Jersey2 - Employee	380
16.1.9	REST Service - Using Jersey2 - EmployeeList	384
16.1.10	REST Client - Using Jersey2 - EmployeeList	385
16.1.11	REST Service - Using Jersey2 - Employee Oper	387
16.1.12	REST Client - Using Jersey2 - Employee Oper	393
16.2	SOAP	399
16.2.1	SOAP Service - Using Endpoint Publisher	399
16.2.2	Java Web Service Client via wsimport tool	405
16.2.3	Tracing SOAP Traffic	409
16.2.4	SOAP Service - Deploy JAX-WS web services on Tomcat	413
16.2.5	SoapUI	417

PART VI APPENDIX

A	IntelliJ	429
A.1	Download IntelliJ	429
A.2	Install IntelliJ	429
A.3	IntelliJ - hello-world web application	431
A.3.1	Lanuch IntelliJ	431
A.3.2	Import hello-world Maven Project	433
A.3.3	Configure Tomcat in IntelliJ	439
A.3.4	hello-world	443
B	SQL Developer	445
B.1	Download Show Me Password	445
B.2	Install Show Me Password	446
B.3	Use Show Me Password	447
C	Virtual Machine - Oracle Response Files	449
C.1	db_install.rsp	449
C.2	dbcac.rsp	456

FOREWORD

Hello World!

PREFACE

Hello World!

ACKNOWLEDGMENTS

To all the wonderful people I owe a deep sense of gratitude especially now that this project has been completed.

H. Z.

PART I

**INSTALLATION
AND
SIMPLE PROJECTS**

CHAPTER 1

INSTALLATION AND CONFIGURATION

1.1 Java

1.1.1 Download Java

- Download Java 7 64 bit (jdk-7u79-windows-x64.exe) from:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
<http://www.oracle.com/technetwork/archive-139210.html>

1.1.2 Install Java - With Administrator Rights

Double click jdk-7u79-windows-x64.exe

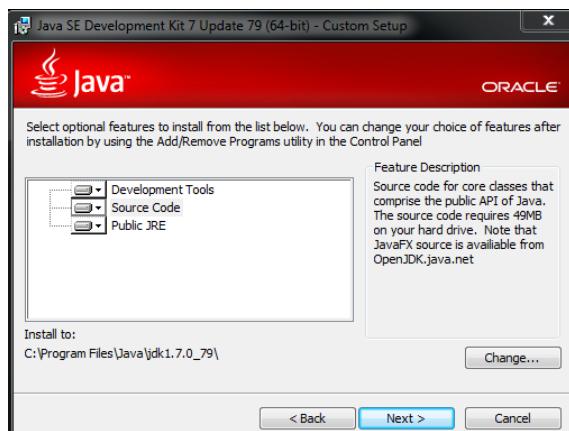


Click Next

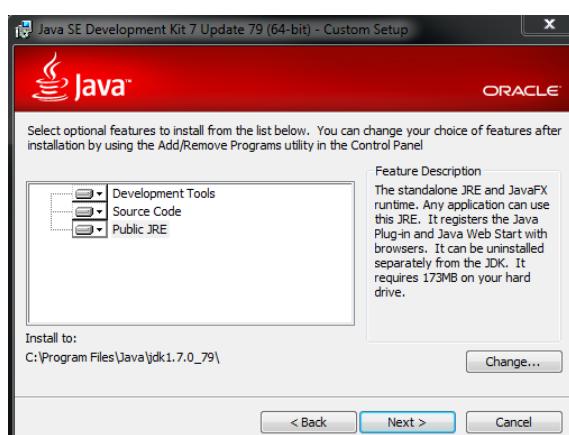
4 INSTALLATION AND CONFIGURATION



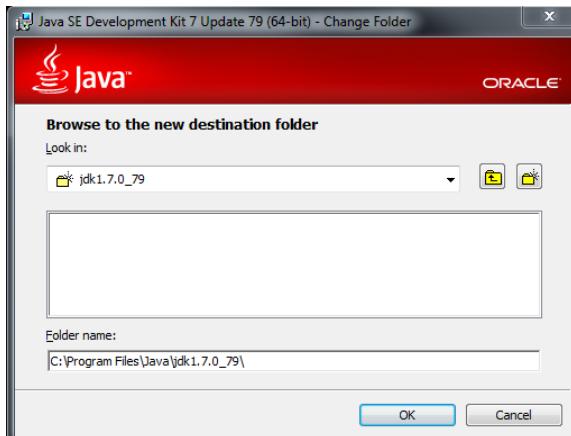
Click Drop Down arrow of Source Code, and select This feature, and all subfeatures, will be installed on local hard drive.



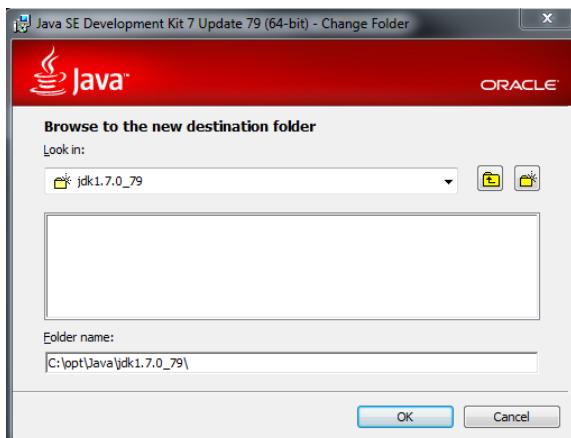
Click Drop Down arrow of Public JRE, and select This feature, and all subfeatures, will be installed on local hard drive.



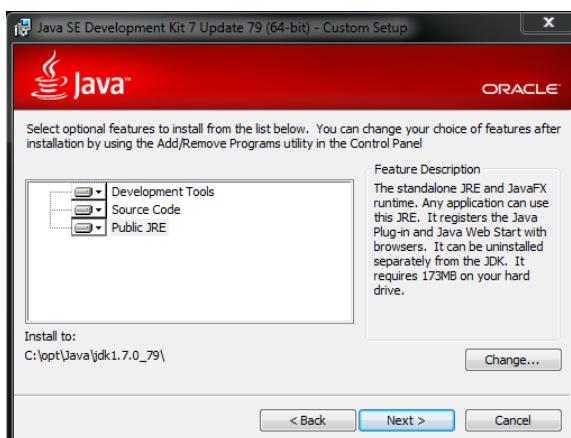
Click Change



Change Folder name to: C:\opt\Java\jdk1.7.0_79\

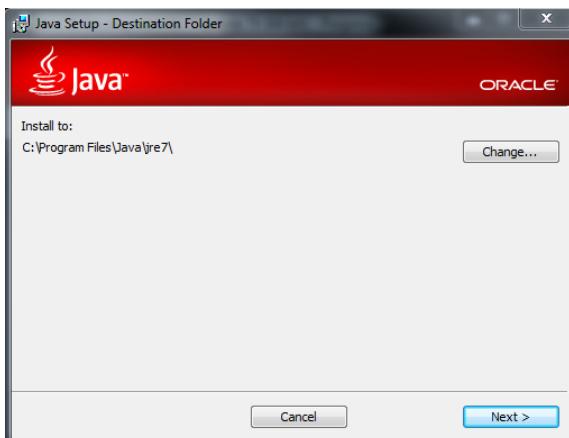


Click OK



Click Next

6 INSTALLATION AND CONFIGURATION



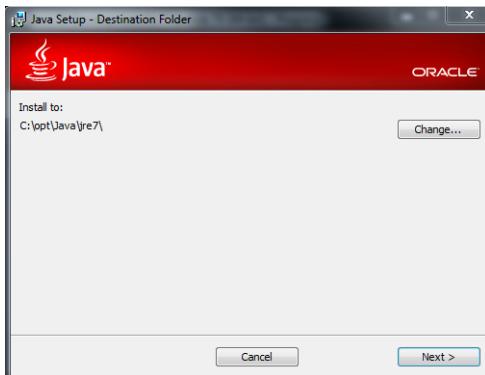
Click Change



Change Folder name to: C:\opt\Java\jre7\



Click OK



Click Next

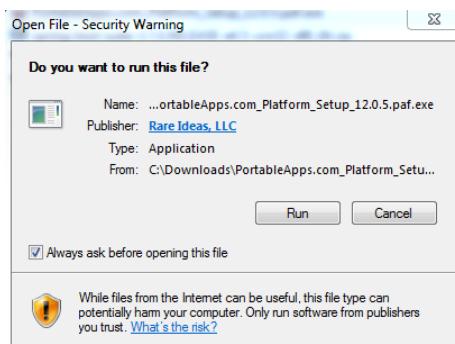


Click Close

1.1.3 Install Java - Without Administrator Rights

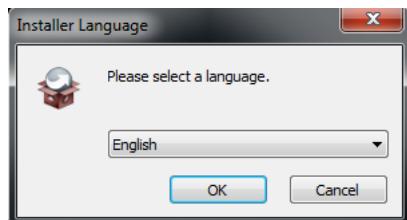
1.1.3.1 Install PortableApps and 7-Zip

- Download PortableApps (PortableApps.com_Platform_Setup_12.0.5.paf.exe) from: <http://portableapps.com/download>
- Install PortableApps and 7-Zip
Double click PortableApps.com_Platform_Setup_12.0.5.paf.exe

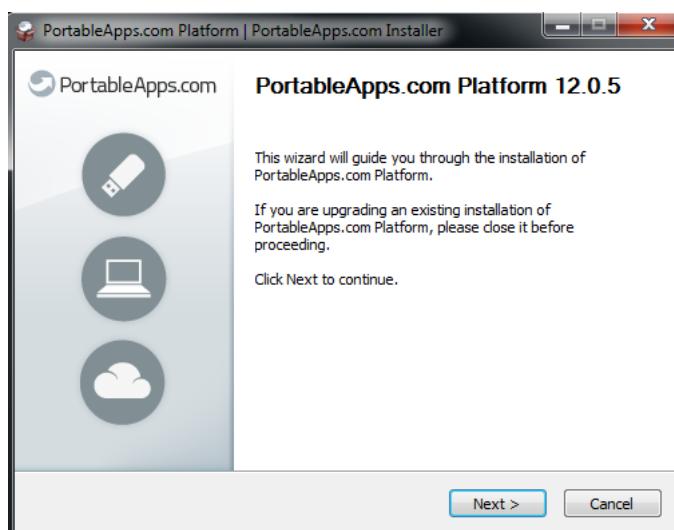


Click Run

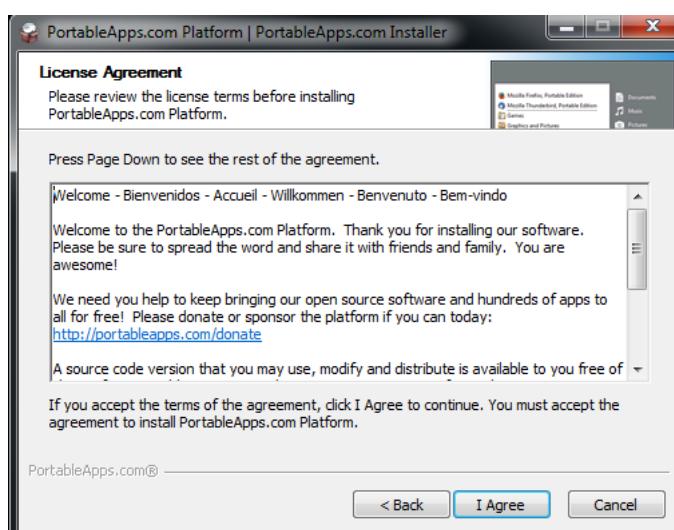
8 INSTALLATION AND CONFIGURATION



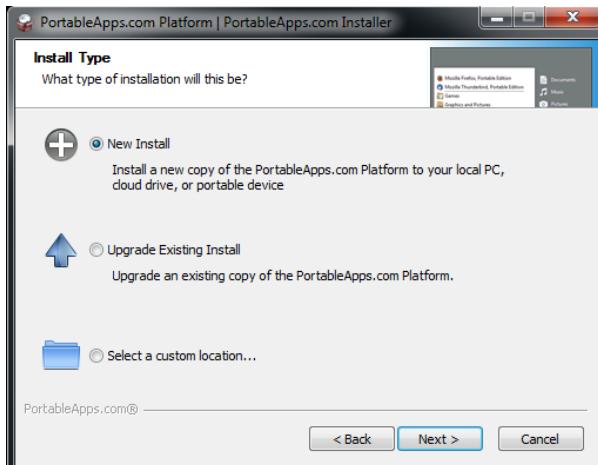
Click OK



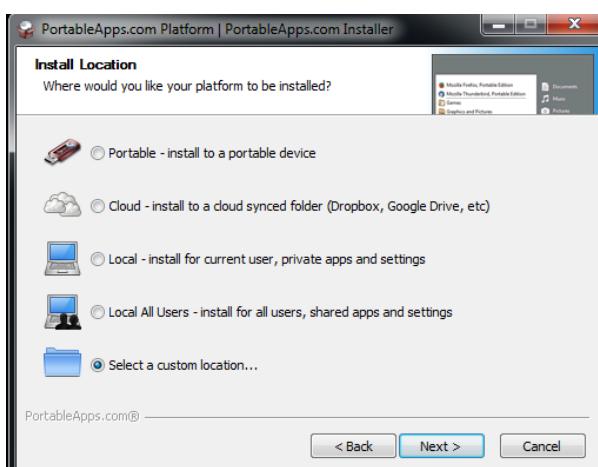
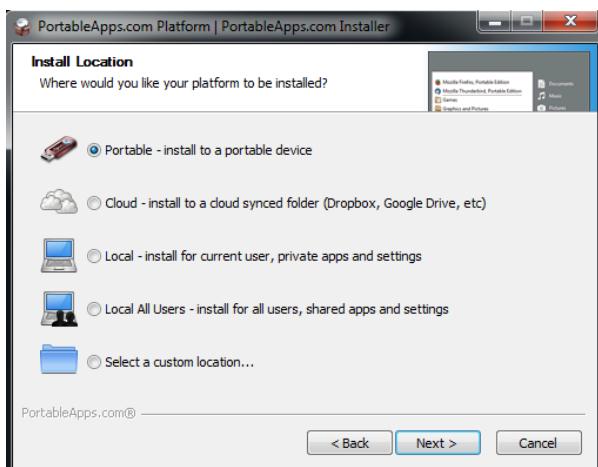
Click Next



Click I Agree

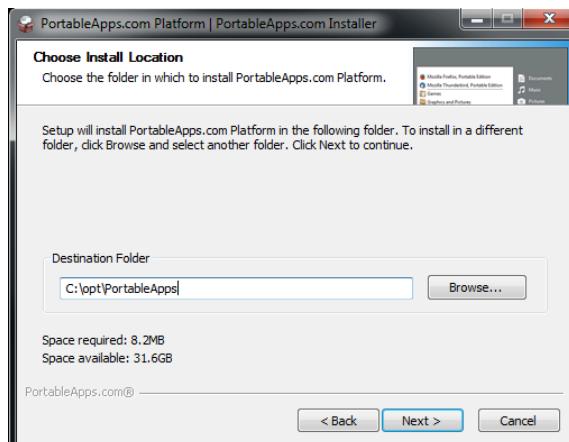


Click Next

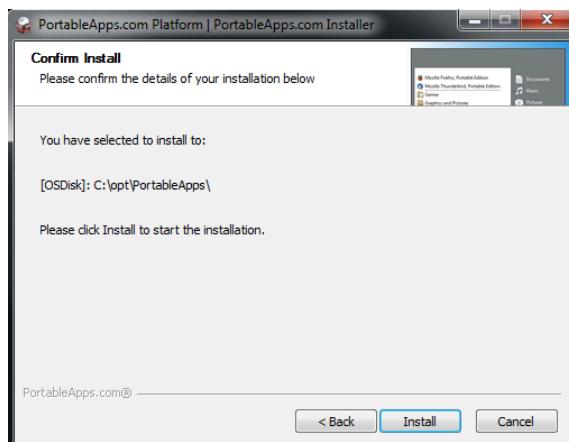


Select: Select a custom location, then click Next

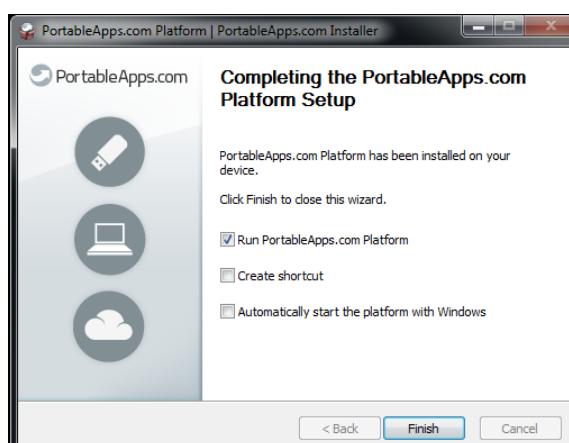
10 INSTALLATION AND CONFIGURATION



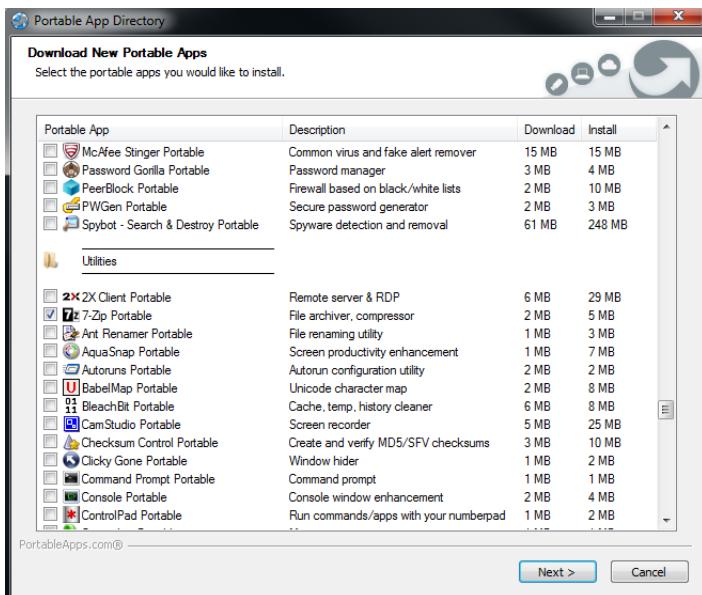
Change Desination Folder to your target location, C:\opt\PortableApps, then click Next



Click Install



Click Finish



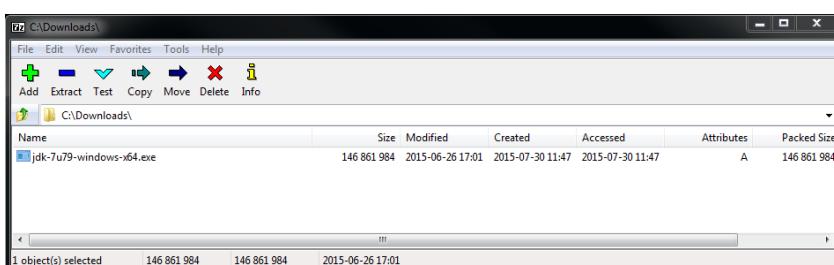
Select 7-Zip Portable, then click Next



Click Finish

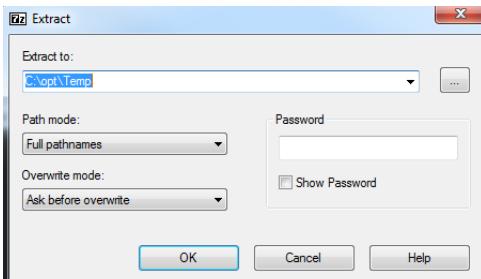
1.1.3.2 Install Java 7

- Download Java 7 64 bit (jdk-7u79-windows-x64.exe) from:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Install Java
Extract jdk-7u79-windows-x64.exe using 7-Zip Portable

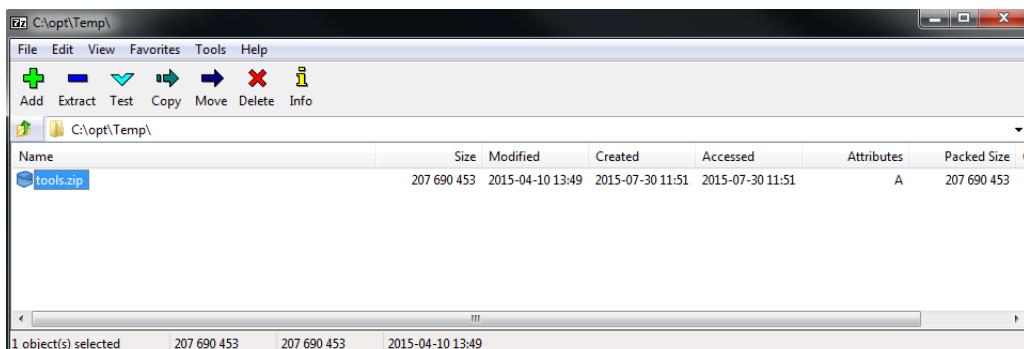


Click Extract

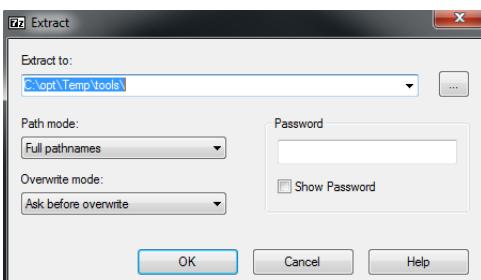
12 INSTALLATION AND CONFIGURATION



Change Extract to location to your temporary location, C:\opt\Temp, then click OK
Use 7-Zip Portable to unzip C:\opt\Temp\tools.zip



Click Extract



Click OK

Go to command line

Change directory to C:\opt\Temp\tools

Execute command:

```
for /r %x in (*.pack) do .\bin\unpack200 -r "%x" "%~dx%~px%~nx.jar"
```

This will convert all pack files into jar

Copy whole directory and all subdir of C:\opt\Temp\tools where you want your JDK to be (C:\opt\Java\jdk1.7.0_79) and setup manually JAVA_HOME and PATH to point to your JDK dir and its BIN subdir.

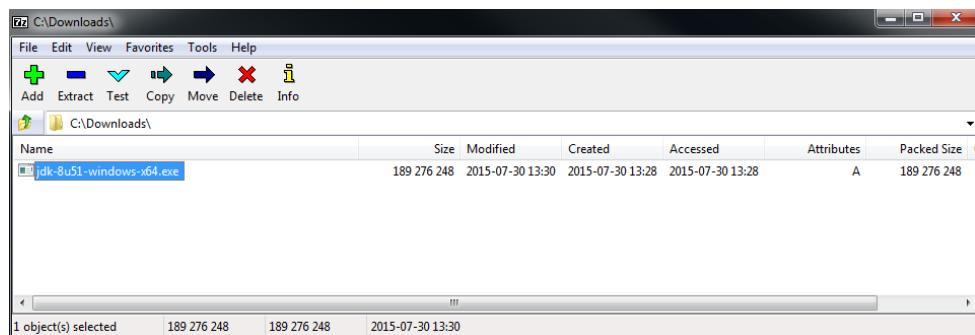
```
1 SET JAVA_HOME=C:\opt\Java\jdk1.7.0_79
2 SET PATH=%JAVA_HOME%\bin;%PATH%
```

Check Java version

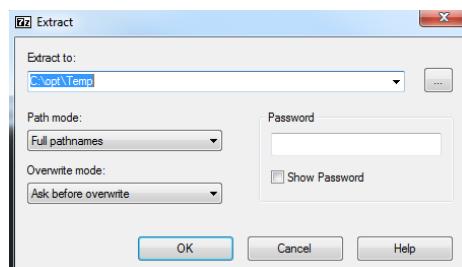
```
c:\>java -version
java version "1.7.0_79"
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)
```

1.1.3.3 Install Java 8

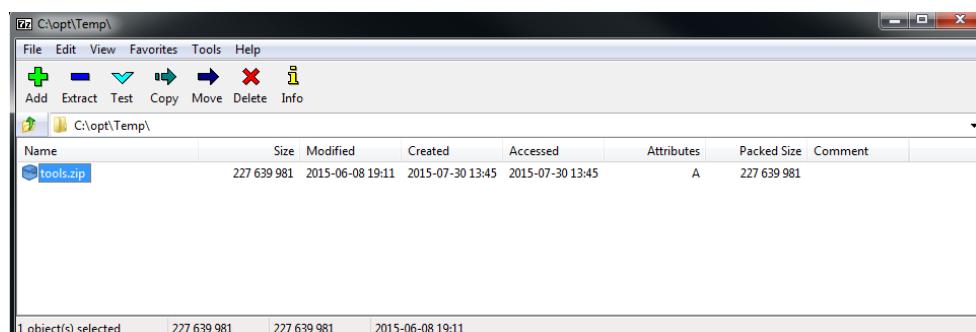
- Download Java 8 64 bit (jdk-8u51-windows-x64.exe) from:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Install Java
Extract jdk-8u51-windows-x64.exe using 7-Zip Portable



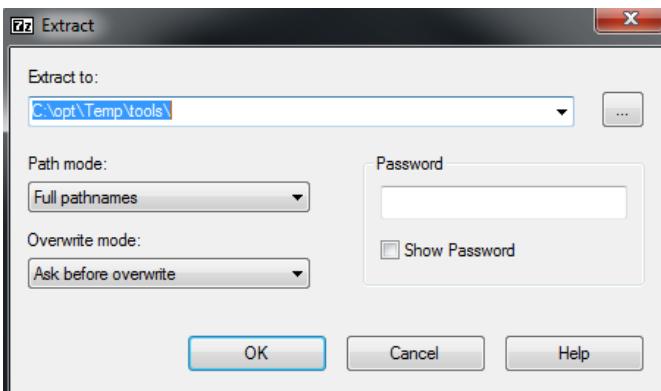
Click Extract



Change Extract to location to your temporary location, C:\opt\Temp, then click OK
Use 7-Zip Portable to unzip C:\opt\Temp\tools.zip



Click Extract



Click OK

Go to command line

Change directory to C:\opt\Temp\tools

Execute command:

```
for /r %x in (*.pack) do .\bin\unpack200 -r "%x" "%~dx%~px%~nx.jar"
```

This will convert all pack files into jar

Copy whole directory and all subdir of C:\opt\Temp\tools where you want your JDK to be (C:\opt\Java\jdk1.8.0_51) and setup manually JAVA_HOME and PATH to point to your JDK dir and its BIN subdir.

```
1 SET JAVA_HOME=C:\opt\Java\jdk1.8.0_51
2 SET PATH=%JAVA_HOME%\bin;%PATH%
```

Check Java version

```
c:\>java -version
java version "1.8.0_51"
Java(TM) SE Runtime Environment (build 1.8.0_51-b16)
Java HotSpot(TM) 64-Bit Server VM (build 25.51-b03, mixed mode)
```

1.1.4 Configure Java

Setup Windows 7 User Environment Variables

- Method 1

Go to Windows → Search programs and files → type in env

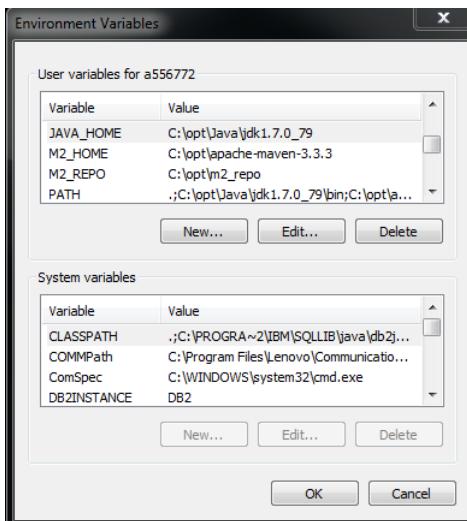
Select Edit environment variables for your account

New

- Method 2

Go to Control Panel → In Search Control Panel, type in env

Select Edit environment variables for your account



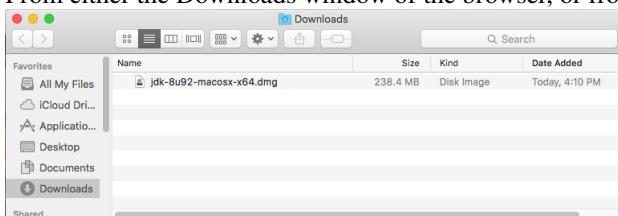
1.2 Java - Mac

1.2.1 Download Java

- Download Java 8 64 bit (jdk-8u92-macosx-x64.dmg) from:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
JDK DOWNLOAD ->Java SE Development Kit 8u92 ->Mac OS X ->jdk-8u92-macosx-x64.dmg

1.2.2 Install Java

From either the Downloads window of the browser, or from the file browser

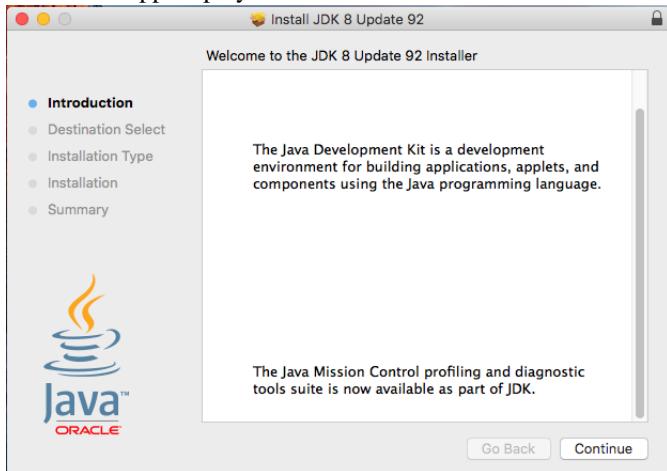


Double click jdk-8u92-macosx-x64.dmg file to lanuch it



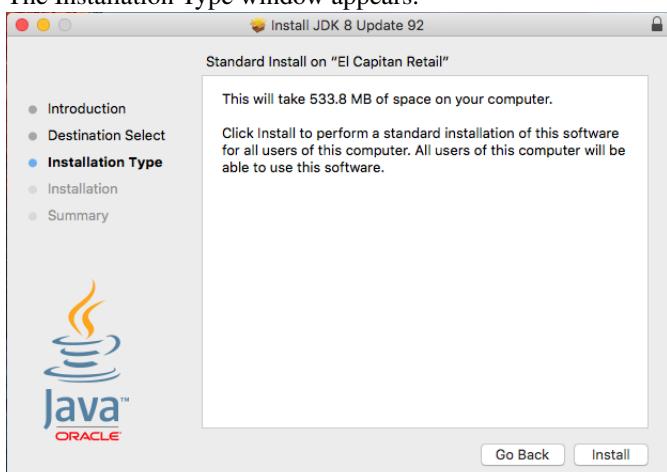
A Finder window appears containing an icon of an open box and the name of the .pkg file.

Double click the package icon to launch the Install app
The Install app displays the Introduction window.



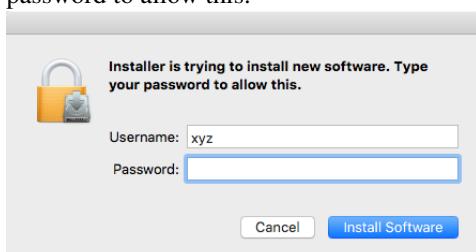
Click Continue

The Installation Type window appears.



Click Install

A window appears that says "Installer is trying to install new software. Type your password to allow this."



Enter the Administrator login and password and click Install Software.
The software is installed and a confirmation window appears.



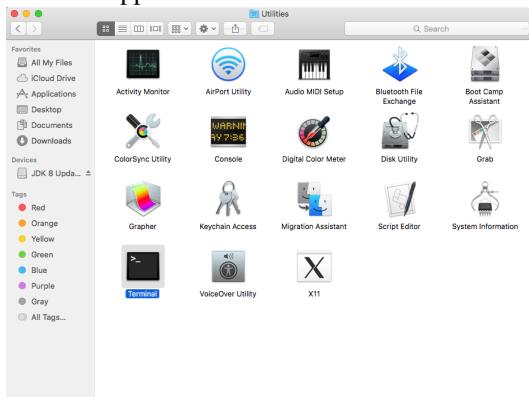
Click Close

After the software is installed, delete the .dmg file if you want to save disk space.

1.2.3 Verify Java Installation

Open a Terminal window

On OS X, open your Applications folder, then open the Utilities folder. Open the Terminal application



Type java -version

```
Last login: Wed Jun 22 15:13:46 on console
[xyz-Mac:~ xyz$ java -version
java version "1.8.0_92"
Java(TM) SE Runtime Environment (build 1.8.0_92-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.92-b14, mixed mode)
xyzs-Mac:~ xyz$ ]
```

The screenshot shows a terminal window with the title "xyz — bash — 80x24". The command "java -version" is typed and executed, displaying the Java version information: "java version "1.8.0_92"" and "Java(TM) SE Runtime Environment (build 1.8.0_92-b14)". The prompt "[xyzs-Mac:~ xyz\$]" is visible at the bottom.

1.2.4 Setup JAVA_HOME

In Mac OSX 10.5 or later, Apple recommends to set the \$JAVA_HOME variable to /usr/libexec/java_home, just export \$JAVA_HOME in file ~/.bash_profile or ~/.profile. Open a Terminal window

```

1 $ cd
2
3 $ vi .bash_profile
4
5 export JAVA_HOME=$(/usr/libexec/java_home)
6
7 $ source .bash_profile
8
9 $ echo $JAVA_HOME
10 /Library/Java/JavaVirtualMachines/jdk1.8.0_92.jdk/Contents/Home

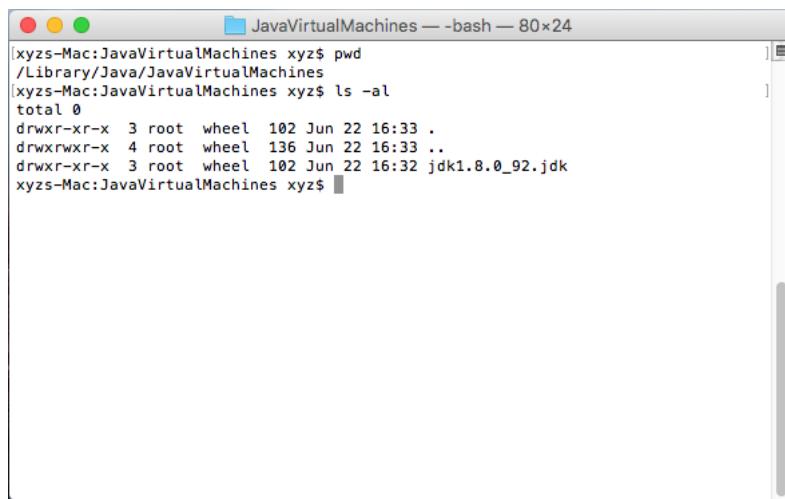
```

1.2.5 Uninstall Java

Open a Terminal window

Navigate to /Library/Java/JavaVirtualMachines and remove the directory whose name matches the following format:

/Library/Java/JavaVirtualMachines/jdkmajor.minor.macro[_update].jdk



```

xyzs-Mac:JavaVirtualMachines xyz$ pwd
/Library/Java/JavaVirtualMachines
[xyzs-Mac:JavaVirtualMachines xyz$ ls -al
total 0
drwxr-xr-x  3 root  wheel  102 Jun 22 16:33 .
drwxrwxr-x  4 root  wheel  136 Jun 22 16:33 ..
drwxr-xr-x  3 root  wheel  102 Jun 22 16:32 jdk1.8.0_92.jdk
xyzs-Mac:JavaVirtualMachines xyz$ 

```

For example, to uninstall 8u92:

```
$ rm -rf jdk1.8.0_92.jdk
```

Do not attempt to uninstall Java by removing the Java tools from /usr/bin. This directory is part of the system software and any changes will be reset by Apple the next time you perform an update of the OS.

1.3 Tomcat

1.3.1 Download Tomcat

- Download Tomcat7 64 bit (apache-tomcat-7.0.63-windows-x64.zip) from:
<http://tomcat.apache.org/download-70.cgi>

1.3.2 Extract Tomcat

Extract apache-tomcat-7.0.63-windows-x64.zip to C:\opt\apache-tomcat-7.0.63

1.3.3 Setup Tomcat Home

```
SET TOMCAT_HOME=C:\opt\apache-tomcat-7.0.63
```

1.4 Tomcat - Mac

1.4.1 Download Tomcat

- Download Tomcat7 64 bit (apache-tomcat-7.0.70.tar.gz) from:
<http://tomcat.apache.org/download-70.cgi>

1.4.2 Extract Tomcat

Double click apache-tomcat-7.0.70.tar.gz file in Downloads folder, it will extract to
~/Downloads/apache-tomcat-7.0.70

Open a Terminal window
move the unarchived distribution to /opt

```
1 $ cd
2
3 $ sudo mkdir -p /opt
4
5 $ sudo mv ~/Downloads/apache-tomcat-7.0.70 /opt
```

To make it easy to replace this release with future releases, we are going to create a symbolic link that we are going to use when referring to Tomcat (after removing the old link, you might have from installing a previous version):

```
1 $ sudo rm -f /Library/Tomcat
2
3 $ sudo ln -s /opt/apache-tomcat-7.0.70 /Library/Tomcat
```

Change ownership of the /Library/Tomcat folder hierarchy:

```
$ sudo chown -R xyz /Library/Tomcat
$ sudo chgrp -R staff /Library/Tomcat
```

Make all scripts executable:

```
$ sudo chmod +x /Library/Tomcat/bin/*.sh
```

1.4.3 Start Tomcat

Open a Terminal window

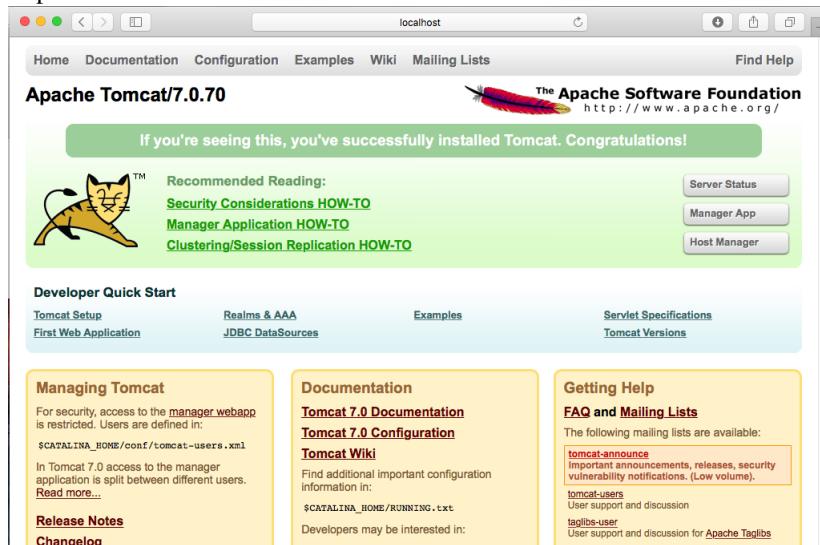
```

1 xyzs-Mac:~ xyz$ /Library/Tomcat/bin/startup.sh
2 Using CATALINA_BASE:   /Library/Tomcat
3 Using CATALINA_HOME:   /Library/Tomcat
4 Using CATALINA_TMPDIR: /Library/Tomcat/temp
5 Using JRE_HOME:
   ↗ /Library/Java/JavaVirtualMachines/jdk1.8.0_92.jdk/Contents/Home
6 Using CLASSPATH:
   ↗ /Library/Tomcat/bin/bootstrap.jar:/Library/Tomcat/bin/tomcat-juli.jar
7 Tomcat started.
8 xyzs-Mac:~ xyz$
```

1.4.4 Test Tomcat

Open a browser

<http://localhost:8080/>



Open a Terminal window

```

1 xyzs-Mac:~ xyz$ /Library/Tomcat/bin/shutdown.sh
2 Using CATALINA_BASE:   /Library/Tomcat
3 Using CATALINA_HOME:   /Library/Tomcat
4 Using CATALINA_TMPDIR: /Library/Tomcat/temp
5 Using JRE_HOME:
   ↗ /Library/Java/JavaVirtualMachines/jdk1.8.0_92.jdk/Contents/Home
6 Using CLASSPATH:
   ↗ /Library/Tomcat/bin/bootstrap.jar:/Library/Tomcat/bin/tomcat-juli.jar
7 xyzs-Mac:~ xyz$
```

1.5 Maven

1.5.1 Download Maven

- Download Maven 3 (apache-maven-3.3.3-bin.zip) from:
<https://maven.apache.org/download.cgi>

1.5.2 Extract Maven

Extract apache-maven-3.3.3-bin.zip to C:\opt\apache-maven-3.3.3

1.5.3 Setup Maven Environment

You can setup these environment variables in a batch file, or in your windows user environment variables settings.

```

1 SET M2_HOME=C:\opt\apache-maven-3.3.3
2 SET M2_REPO=C:\opt\m2_repo
3 SET MAVEN_OPTS="-XX:MaxPermSize=128m"
4 SET PATH=%M2_HOME%\bin;%PATH%

```

1.5.4 Configure Maven Settings

- config local repo
- config proxy
- C:\opt\apache-maven-3.3.3\conf\settings.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
5     http://maven.apache.org/xsd/settings-1.0.0.xsd">
6   <localRepository>${env.M2_REPO}</localRepository>
7   <pluginGroups>
8     </pluginGroups>
9   <proxies>
10    <proxy>
11      <id></id>
12      <active>true</active>
13      <protocol>http</protocol>
14      <username></username>
15      <password></password>
16      <host>http.proxy.xyz.com</host>
17      <port>8000</port>
18      <nonProxyHosts>localhost|127.0.0.1</nonProxyHosts>
19    </proxy>
20  </proxies>
21  <servers>
22    </servers>
23  <mirrors>
24    </mirrors>
25  <profiles>
26    </profiles>
27  </settings>
28
29
30
31
32

```

1.5.5 Test Maven

```

1 c:\opt>mvn --version
2 Apache Maven 3.3.3 (7994120775791599e205a5524ec3e0dfe41d4a06;
   → 2015-04-22T06:57:37-05:00)
3 Maven home: C:\opt\apache-maven-3.3.3
4 Java version: 1.7.0_79, vendor: Oracle Corporation
5 Java home: C:\opt\Java\jdk1.7.0_79\jre
6 Default locale: en_US, platform encoding: Cp1252
7 OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"
8
9 c:\opt>

```

1.6 Maven - Mac

1.6.1 Download Maven

- Download Maven 3 (apache-maven-3.3.9-bin.tar.gz) from:
<https://maven.apache.org/download.cgi>

1.6.2 Extract Maven

Double click apache-maven-3.3.9-bin.tar.gz file in Downloads folder, it will extract to
~/Downloads/apache-maven-3.3.9

Open a Terminal window
move the unarchived distribution to /opt

```

1 $ cd
2
3 $ sudo mv ~/Downloads/apache-maven-3.3.9 /opt
4
5 $ sudo mkdir -p /opt/m2_repo
6
7 $ sudo chown -R xyz /opt/m2_repo
8
9 $ sudo chgrp -R staff /opt/m2_repo

```

1.6.3 Setup Maven Environment

Open a Terminal window

```

1 $ cd
2
3 $ vi .bash_profile
4
5 export JAVA_HOME=$(/usr/libexec/java_home)
6 export M2_HOME=/opt/apache-maven-3.3.9
7 export M2_REPO=/opt/m2_repo
8 export PATH=$M2_HOME/bin:$PATH
9
10 $ source .bash_profile
11
12 $ echo $M2_HOME
13 /opt/apache-maven-3.3.9
14
15 $ echo M2_REPO
16 /opt/m2_repo
17
18 $ echo $PATH
19 /opt/apache-maven-3.3.9/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin

```

1.6.4 Configure Maven Settings

▪ /opt/apache-maven-3.3.9/conf/settings.xml

We cannot use \$env.M2_REPO, because Mac OS cannot read bash environments variables for UI apps.

So we hard code /opt/m2_repo here.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
3    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
5        http://maven.apache.org/xsd/settings-1.0.0.xsd">
6    <localRepository>/opt/m2_repo</localRepository>
7
8    <pluginGroups>
9    </pluginGroups>
10
11   <proxies>
12     <proxy>
13       <id></id>
14       <active>true</active>
15       <protocol>http</protocol>
16       <username></username>
17       <password></password>
18       <host>http.proxy.xyz.com</host>
19       <port>8000</port>
20       <nonProxyHosts>localhost|127.0.0.1</nonProxyHosts>
21     </proxy>
22   </proxies>
23
24   <servers>
25   </servers>
26
27   <mirrors>
28   </mirrors>
29
30   <profiles>
31   </profiles>
32 </settings>
```

1.6.5 Test Maven

Open a Terminal window

```

1  $ which mvn
2  /opt/apache-maven-3.3.9/bin/mvn
3
4  $ mvn --version
5  Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5;
6    → 2015-11-10T10:41:47-06:00)
7  Maven home: /opt/apache-maven-3.3.9
8  Java version: 1.8.0_92, vendor: Oracle Corporation
9  Java home: /Library/Java/JavaVirtualMachines/jdk1.8.0_92.jdk/Contents/Home/jre
10 Default locale: en_US, platform encoding: UTF-8
11 OS name: "mac os x", version: "10.11", arch: "x86_64", family: "mac"
```

1.6.6 Maven Help

Open a Terminal window

```
1  $ mvn --help
```

1.7 Eclipse

1.7.1 Eclipse - Windows

1.7.1.1 Download Eclipse

- Download Eclipse IDE for Java EE Developers 64 bit (eclipse-jee-mars-R-win32-x86_64.zip) from:
<https://eclipse.org/downloads/>

1.7.1.2 Extract Eclipse

Extract eclipse-jee-mars-R-win32-x86_64.zip to C:\opt\eclipse

1.7.1.3 Setup Eclipse Path

```
SET PATH=c:\opt\java\jdk1.7.0_79
```

1.7.1.4 Environment Configuration

- Summarized configuration, we can create C:\opt\opt.bat:

```
1 SET JAVA_HOME=C:\opt\Java\jdk1.7.0_79
2 SET M2_HOME=C:\opt\apache-maven-3.3.3
3 SET M2_REPO=C:\opt\m2_repo
4 SET MAVEN_OPTS="-XX:MaxPermSize=128m"
5 SET TOMCAT_HOME=C:\opt\apache-tomcat-7.0.63
6 PATH=%JAVA_HOME%\bin;%M2_HOME%\bin;C:\opt\eclipse
```

1.7.1.5 Configure Eclipse - Maven

- Create eclipse workspaces

```
1 c:\>mkdir workspaces
2 c:\>cd workspaces
3 c:\workspaces>mkdir eclipse
4 c:\workspaces>cd eclipse
5 c:\workspaces\.eclipse>
```

- Add maven repo to eclipse – DO NOT NEEDED FOR ECLIPSE MARS (4.5)

```
1 c:\workspaces\.eclipse>mvn -Dworkspace=my-projects
  ↳   eclipse:configure-workspace
```

or

```
1 c:\workspaces\.eclipse>mvn
  ↳   org.apache.maven.plugins:maven-eclipse-plugin:2.10:configure-workspace
  ↳   -Dworkspace=my-projects
```

This will create a eclipse project (my-projects) folder structure with maven repo variable added to eclipse.

C:\workspaces\.eclipse\my-projects\.metadata\.plugins\org.eclipse.core.runtime\.settings\org.eclipse.jdt.core.preferences

```
org.eclipse.jdt.core.classpathVariable.M2_REPO=C:\\\\opt\\\\m2_repo
```

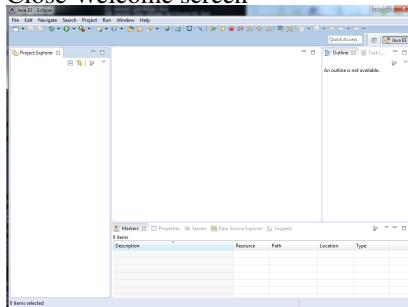
- Launch eclipse

```
c:\workspaces\workspace>eclipse -data my-projects
```

This will launch eclipse with my-projects workspace, or you can double click eclipse, then point to my-projects workspace.

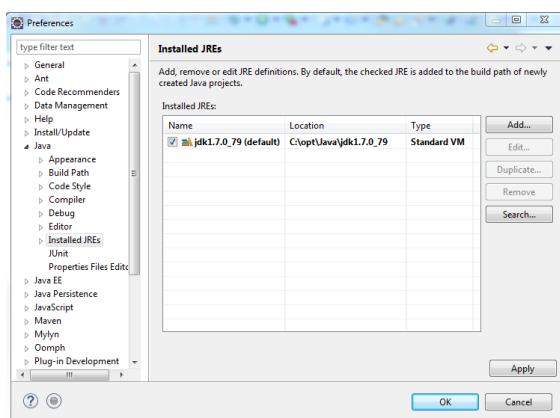


Close Welcome screen



- Check Java version in eclipse

Go to eclipse → Windows → Preferences
Java → Installed JREs

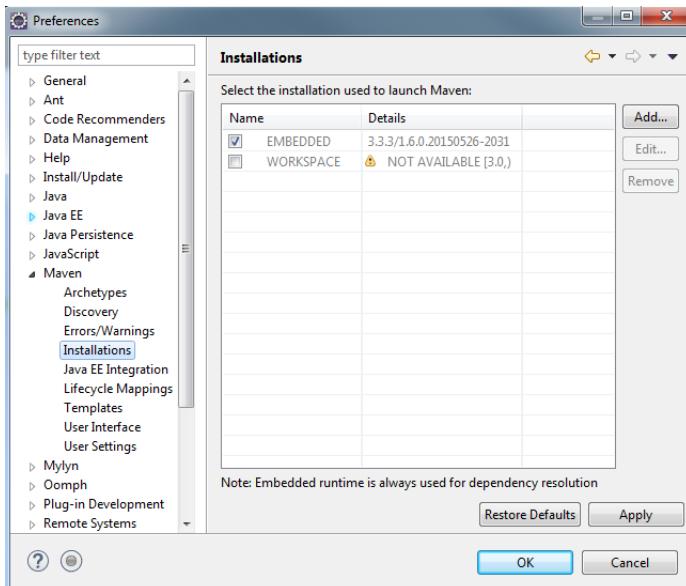


If it is not C:\opt\Java\jdk1.7.0_79, then click Add..., to add Java

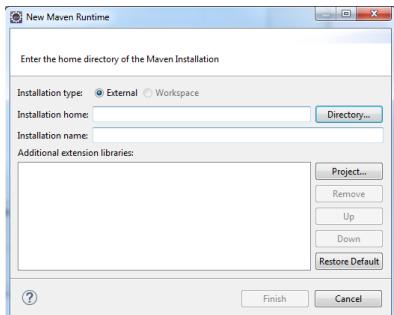
- Config maven in eclipse

Go to eclipse → Windows → Preferences

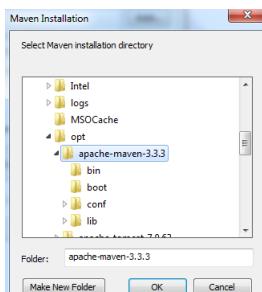
Maven → Installations



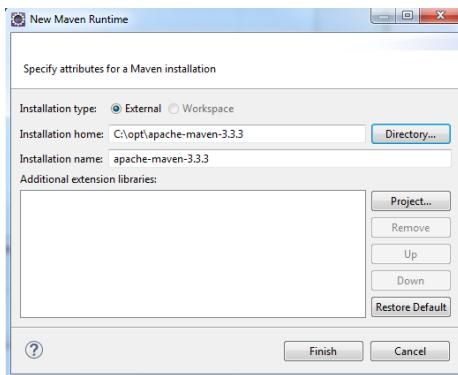
Click Add...



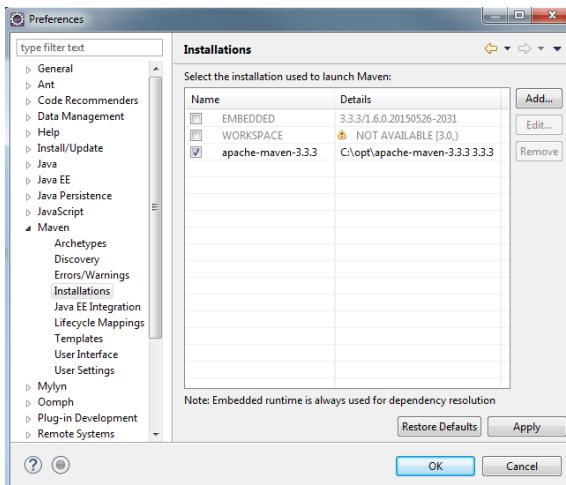
Click Directory...



Browse to c:\opt\apache-maven-3.3.3, click OK

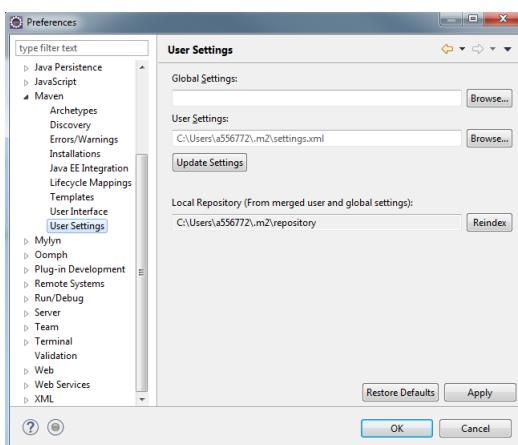


Click Finish

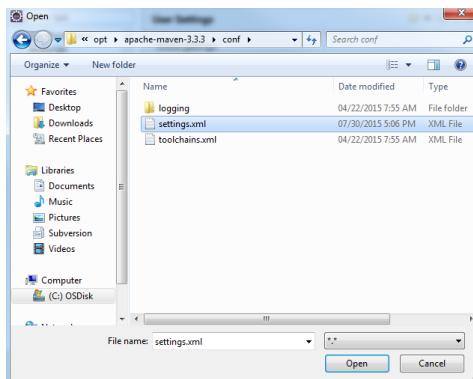


Check apache-maven-3.3.3, then click Apply

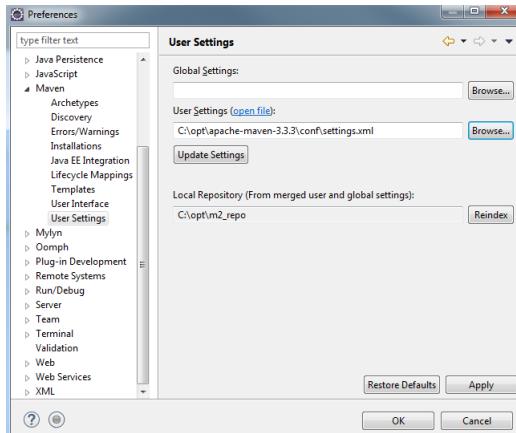
Go to left panel, select User Settings



Click Browse for User Settings



Browse to c:\opt\apache-maven-3.3.3\conf\settings.xml, click Open



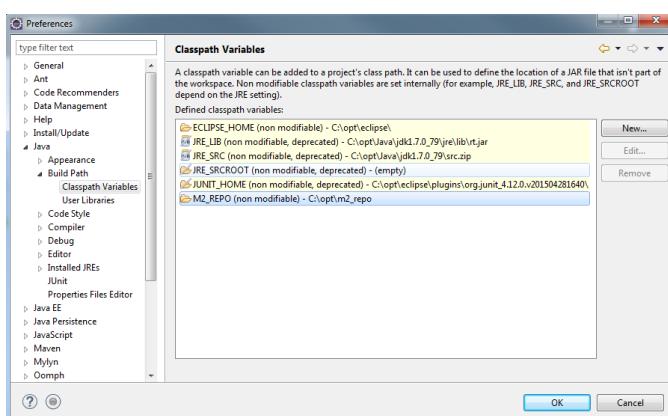
Click OK

- Check maven repo variable

Go to eclipse ->Windows ->Preferences

Java ->Build Path ->Classpath Variables

You will see M2_REPO variable was setup to C:\opt\m2_repo



1.7.2 Eclipse - Mac

1.7.2.1 Download Eclipse

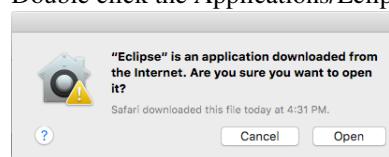
- Download Eclipse IDE for Java EE Developers 64 bit (eclipse-jee-neon-R-macosx-cocoa-x86_64.tar.gz) from:
<https://eclipse.org/downloads/>
 Click Download Packages → Eclipse IDE for Java EE Developers → Mac OS X 64 bit

1.7.2.2 Extract Eclipse

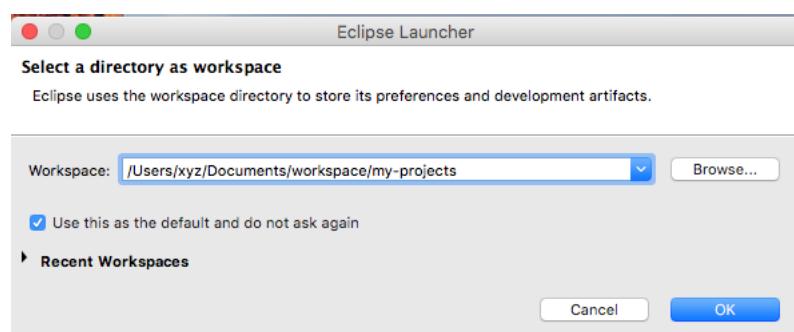
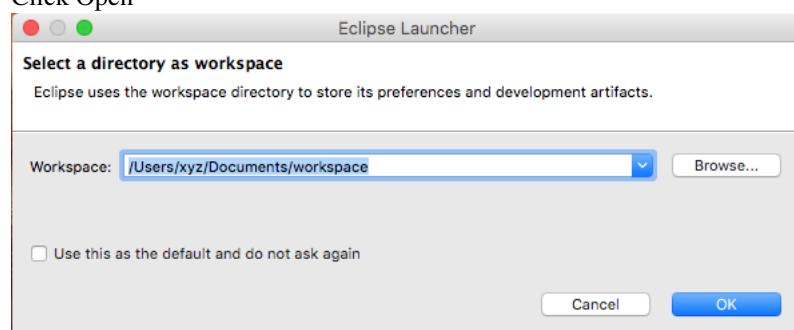
Double click eclipse-jee-neon-R-macosx-cocoa-x86_64.tar.gz file in Downloads folder, it will extract to ~/Downloads/Eclipse folder
 Drag Eclipse folder to Applications folder

1.7.2.3 Configure Eclipse - Maven

- Launch eclipse
 Double click the Applications/Eclipse icon.

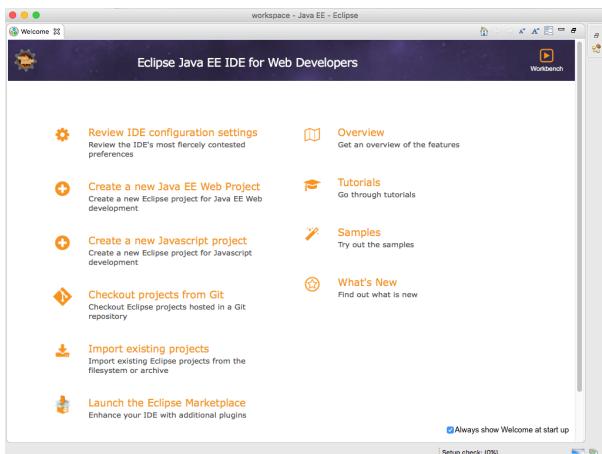


Click Open

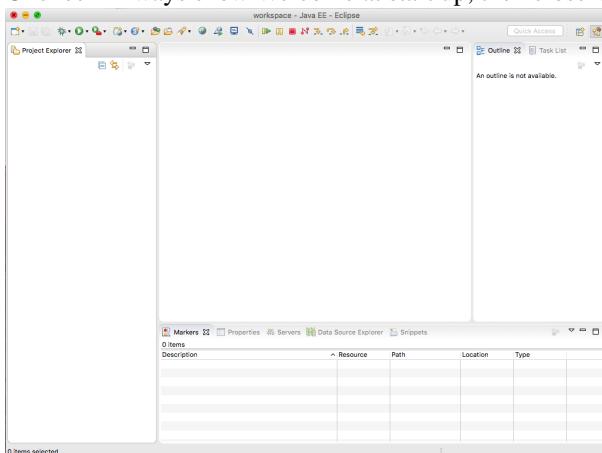


Type in /Users/xyz/Documents/workspace/my-projects, check Use this as the default and do not ask again, then click OK

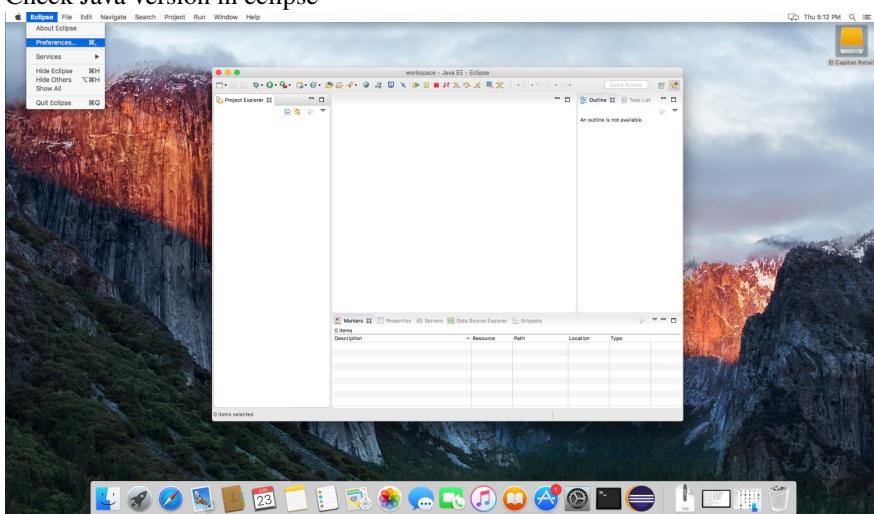
30 INSTALLATION AND CONFIGURATION



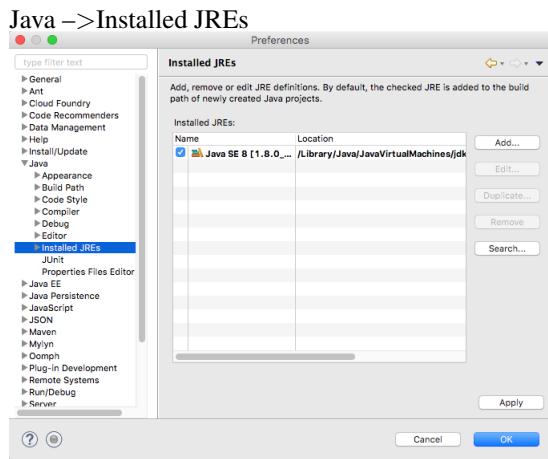
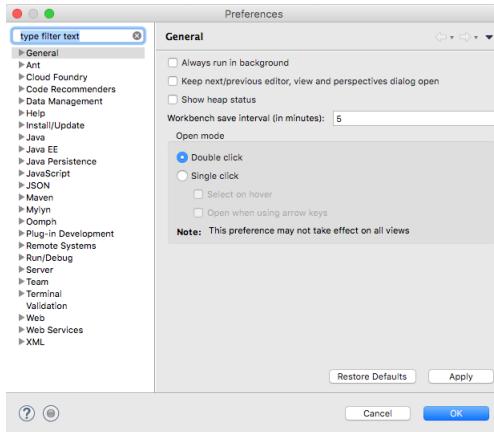
Uncheck Always show Welcome at start up, then close Welcome screen



▪ Check Java version in eclipse

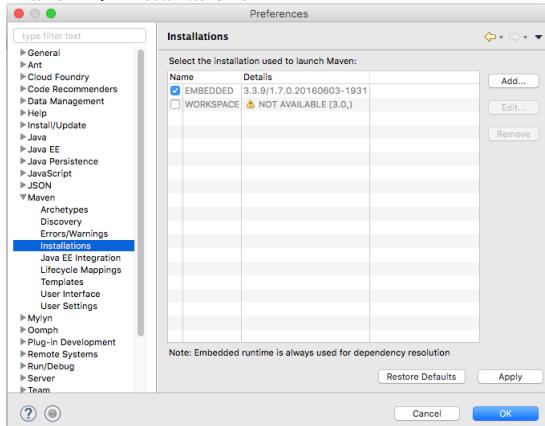


Go to Mac top menu bar → Eclipse → Preferences...



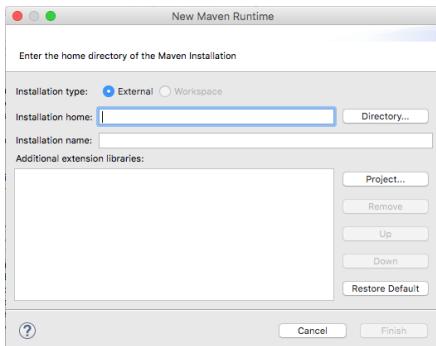
If it is not /Library/Java/JavaVirtualMachines/jdk1.8.0_92.jdk/Contents/Home, then click Add..., to add Java

- Config maven in eclipse
Go to Eclipse → Preferences
Maven → Installations

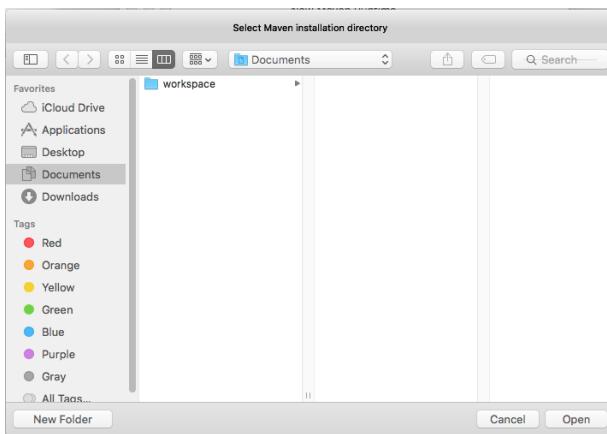


Click Add...

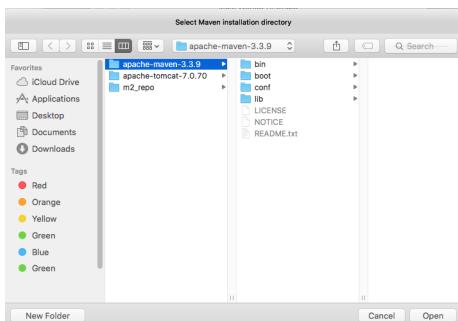
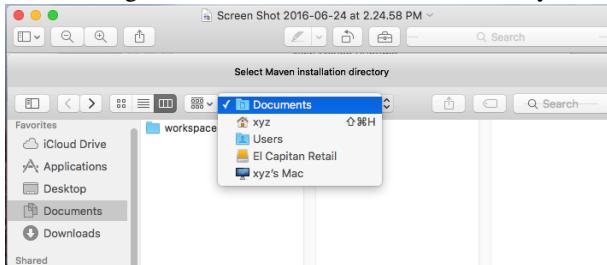
32 INSTALLATION AND CONFIGURATION



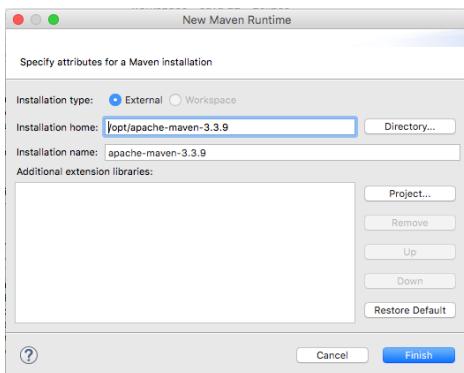
Click Directory...



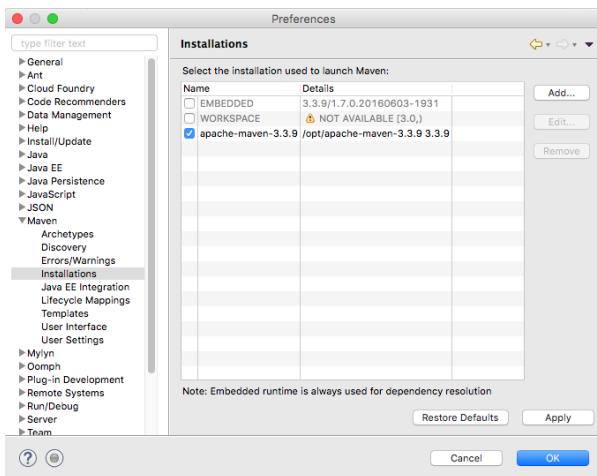
Click the right side arrows of Documents, select xyz or Users



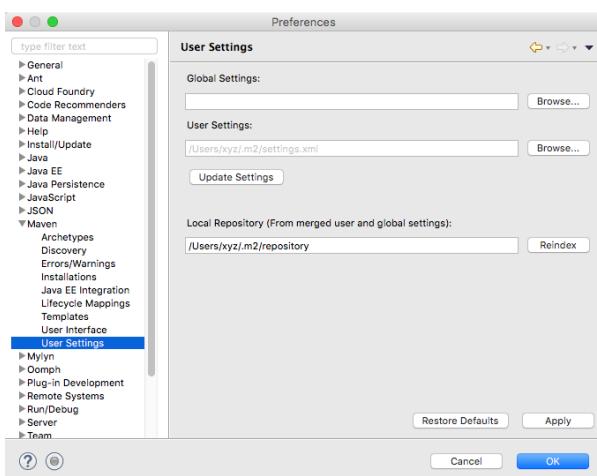
Browse to /opt/apache-maven-3.3.9, click Open



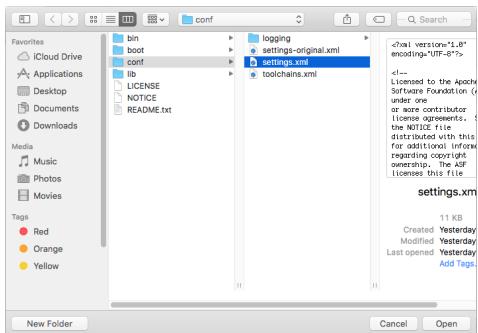
Click Finish



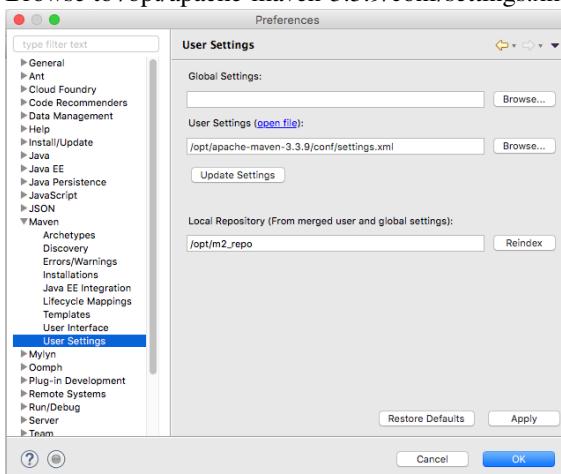
Check apache-maven-3.3.9, then click Apply
Go to left panel, select User Settings



Click Browse for User Settings



Browse to /opt/apache-maven-3.3.9/conf/settings.xml, click Open



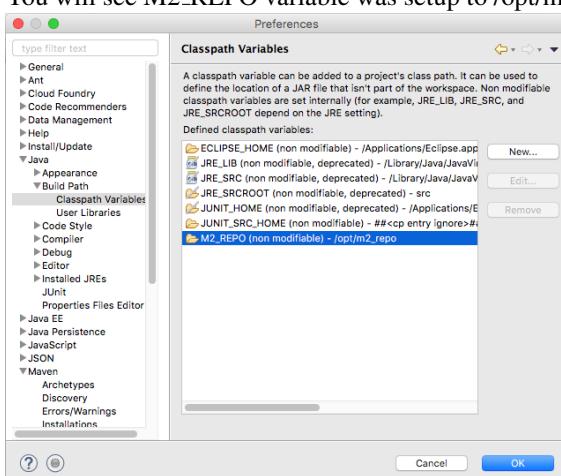
Click OK

- Check maven repo variable

Go to Eclipse → Preferences

Java → Build Path → Classpath Variables

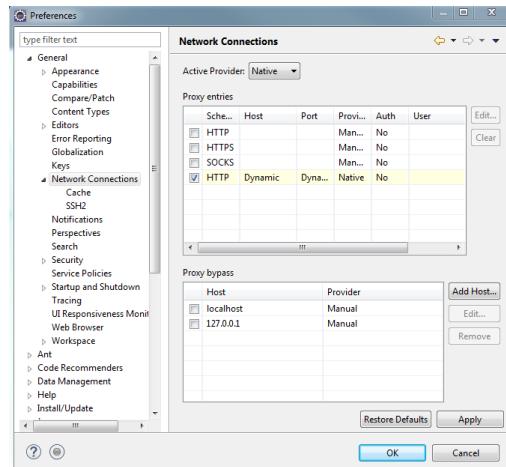
You will see M2_REPO variable was setup to /opt/m2_repo



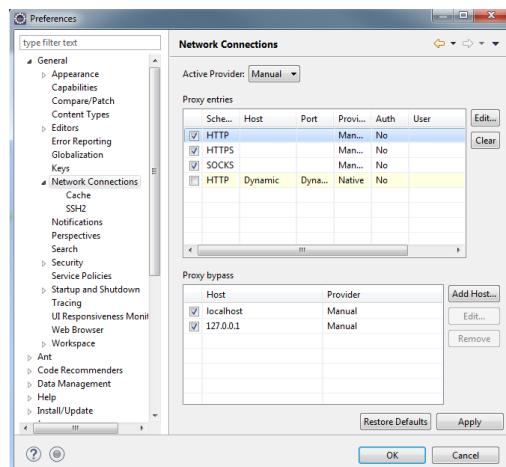
1.7.3 Configure Eclipse - Preferences

- Proxy

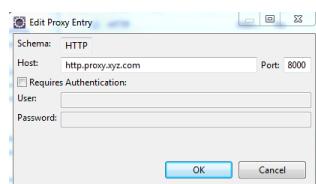
Go to eclipse → Windows → Preferences
General → Network Connections



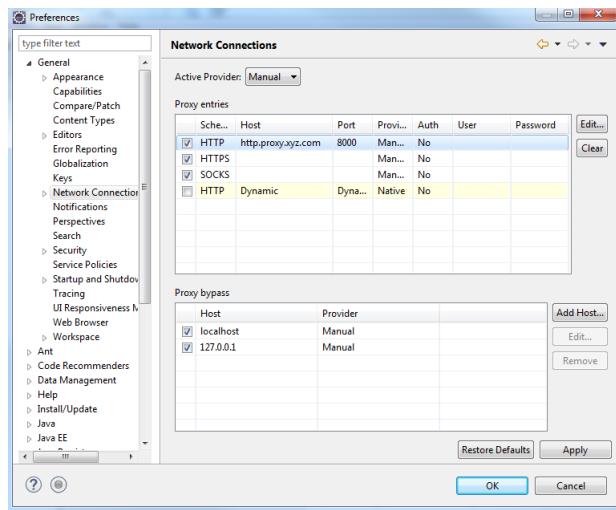
Select Active Provider to Manual
Then click HTTP, Edit



Type in your proxy server host and port

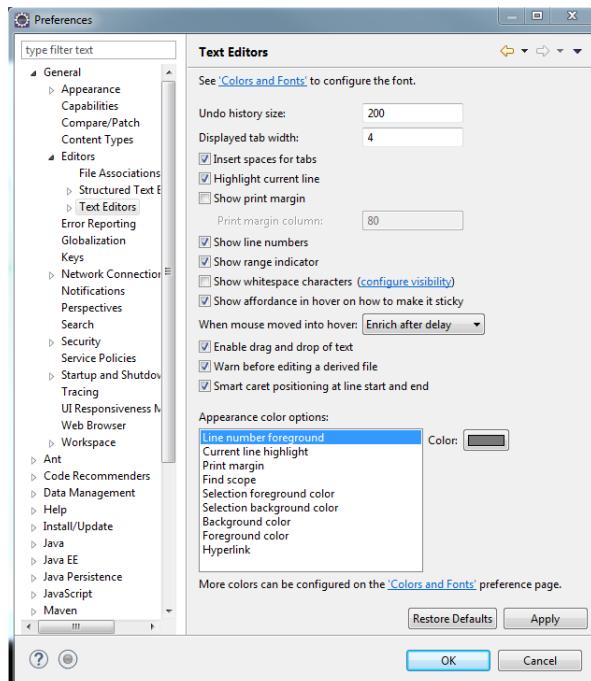


Click OK



▪ Editor

Go to eclipse → Windows → Preferences
General → Editors → Text Editors



Check Insert spaces for tabs

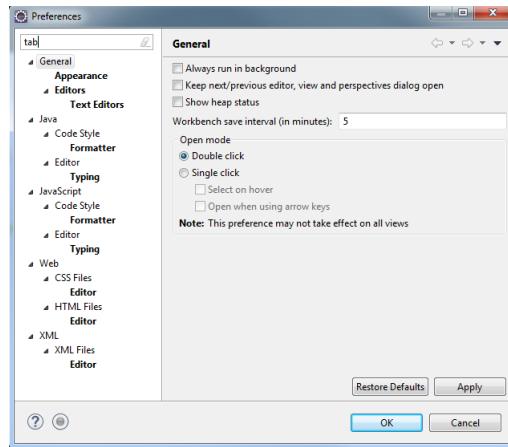
Check Show line numbers

Click OK

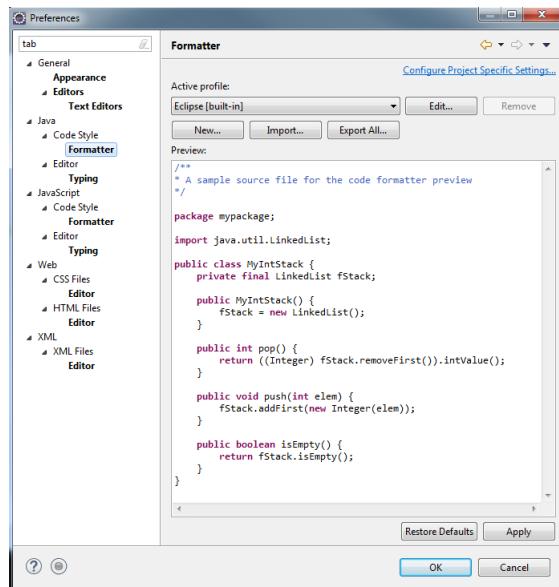
▪ Tab

Go to eclipse → Windows → Preferences

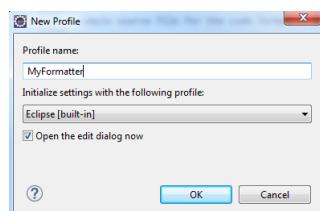
In type filter text, type tab



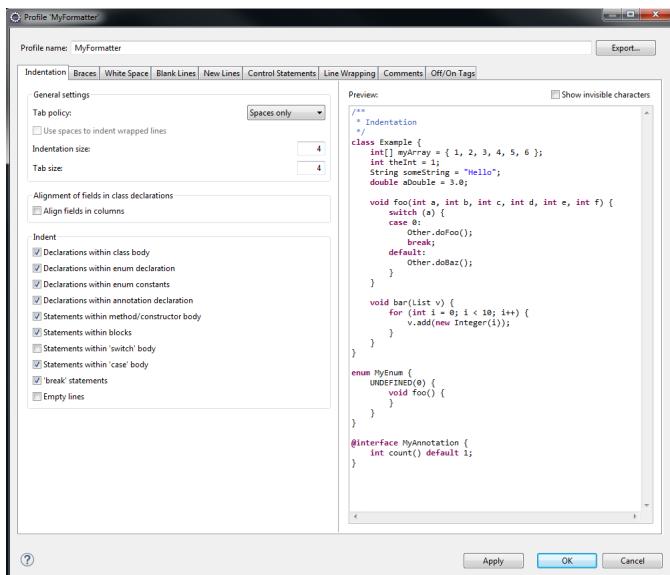
Click Java ->Code Style ->Formatter



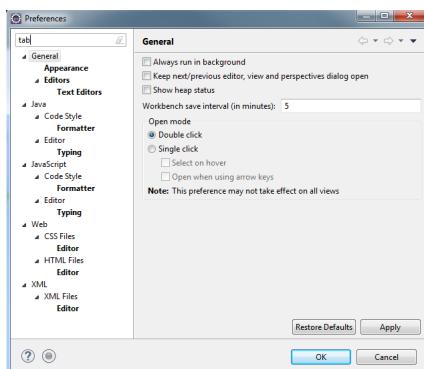
Click New...



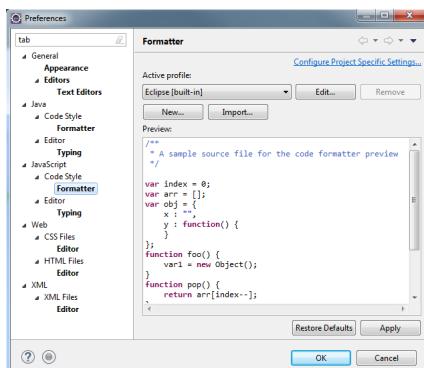
In Profile name, type in MyFormatter, click OK



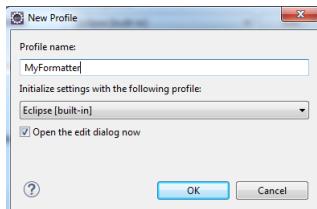
Select Tab policy to Spaces only
Click OK



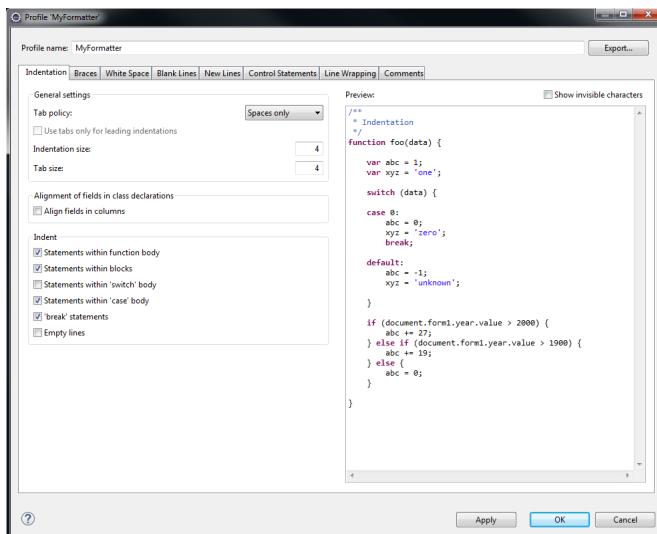
Click JavaScript → Code Style → Formatter



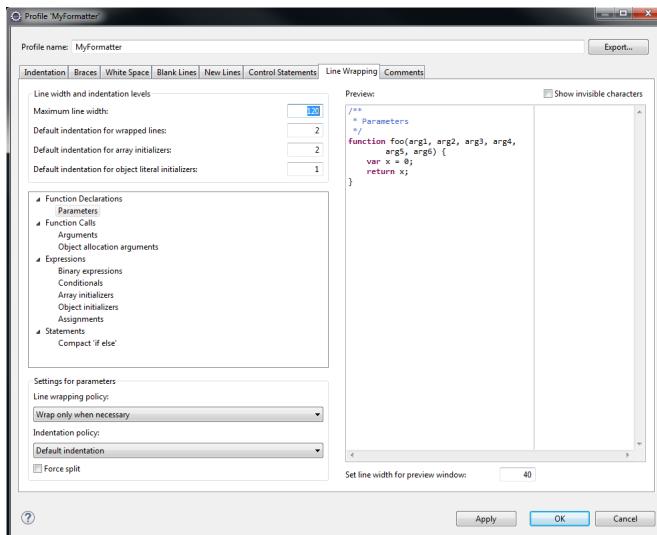
Click New...



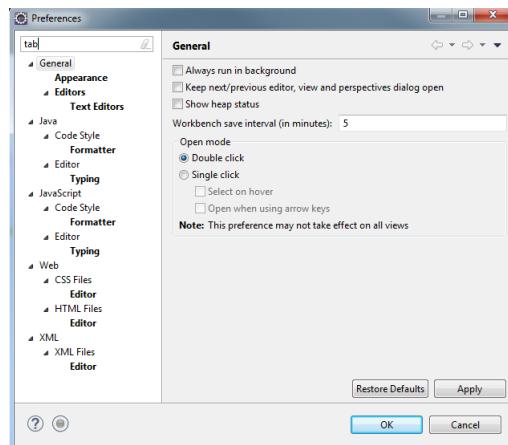
In Profile name, type in MyFormatter, click OK



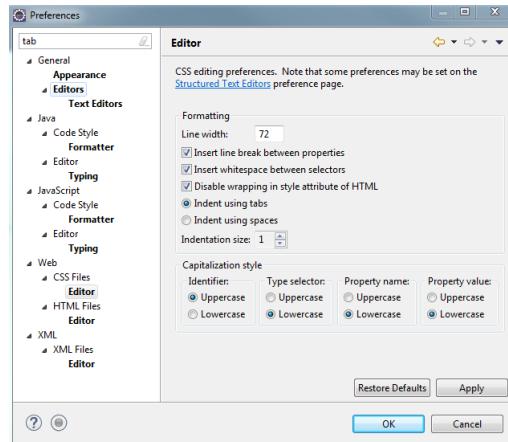
Select Tab policy to Spaces only
Select Line Wrapping Tab



Change Maximum line width to 120
Click OK

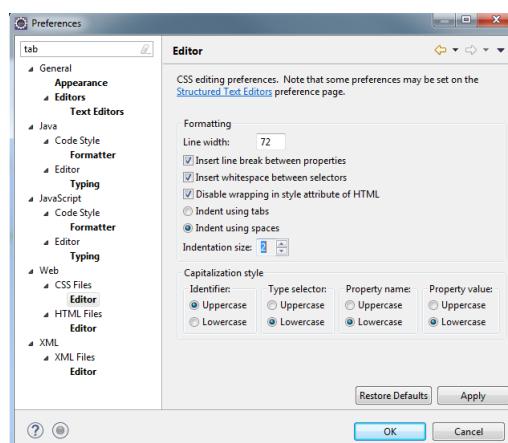


Click Web ->CSS Files ->Editor



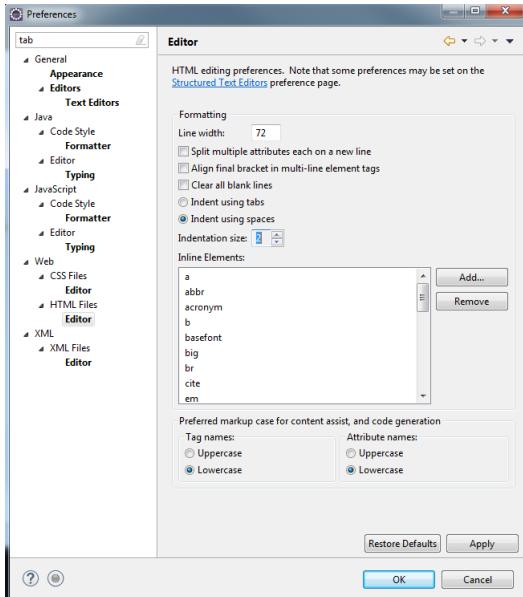
Check Indent using spaces

Increase Indentation size to 2



Click Apply

Click Web ->HTML Files ->Editor

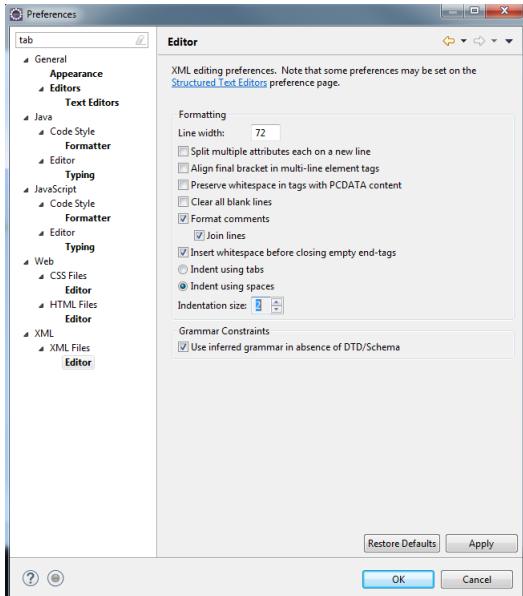


Check Indent using spaces

Increase Indentation size to 2

Click Apply

Click XML ->XML Files ->Editor



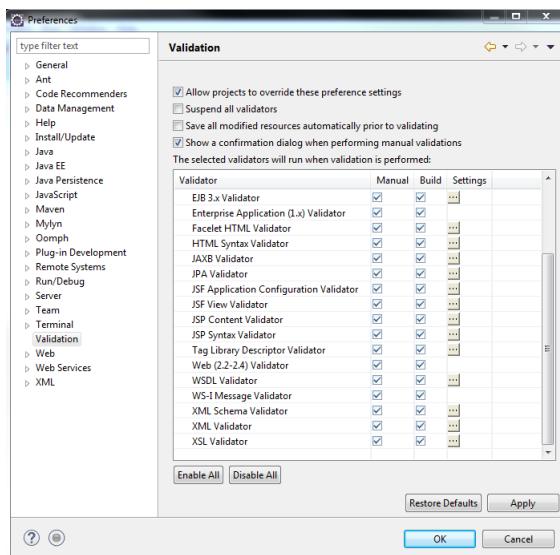
Check Indent using spaces

Increase Indentation size to 2

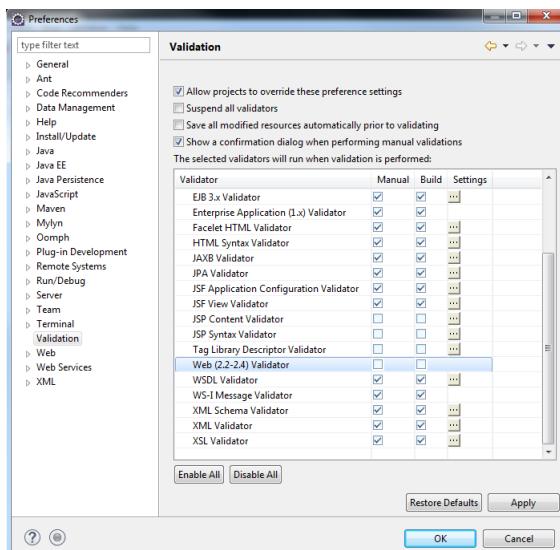
Click OK

▪ Validation

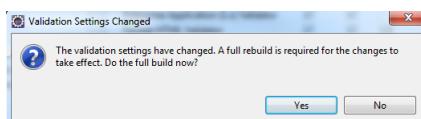
Go to eclipse ->Windows ->Preferences ->Validation



Uncheck JSP Content Validator, JSP Syntax Validator, Tag Library Descriptor Validator



Click OK

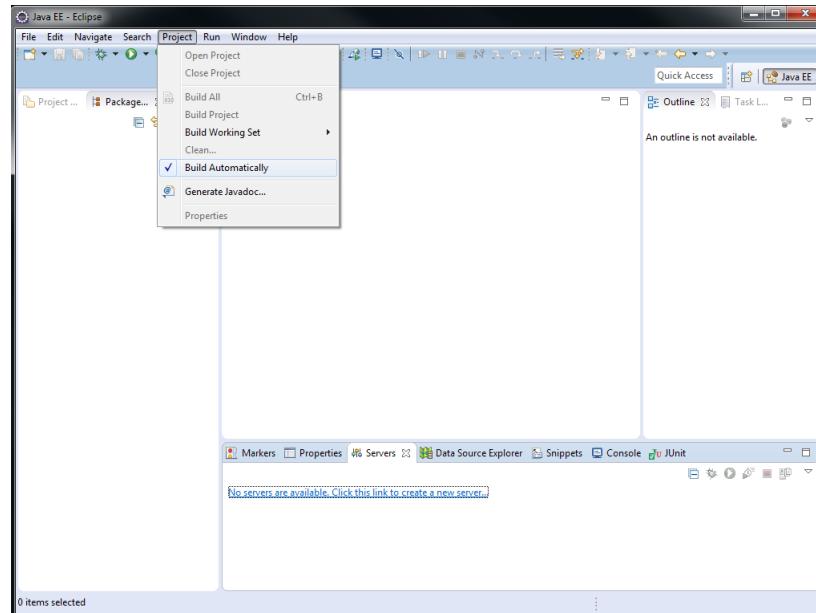


Click Yes

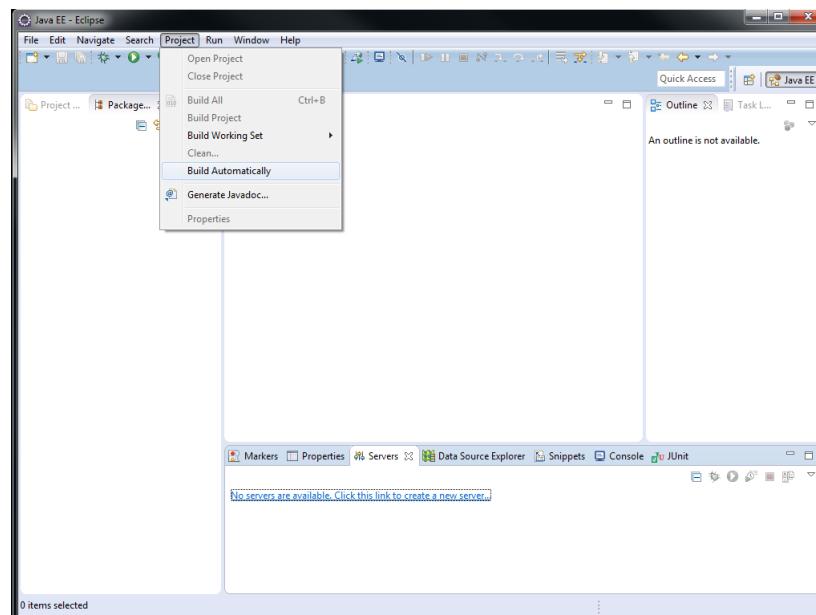
- Build Automatically

Go to eclipse ->Project ->Build Automatically

Other ->Java ->Package Explorer



Uncheck Build Automatically

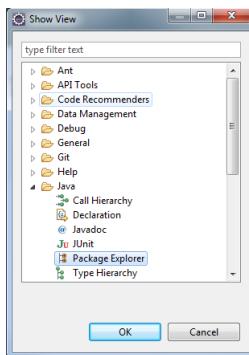


- View

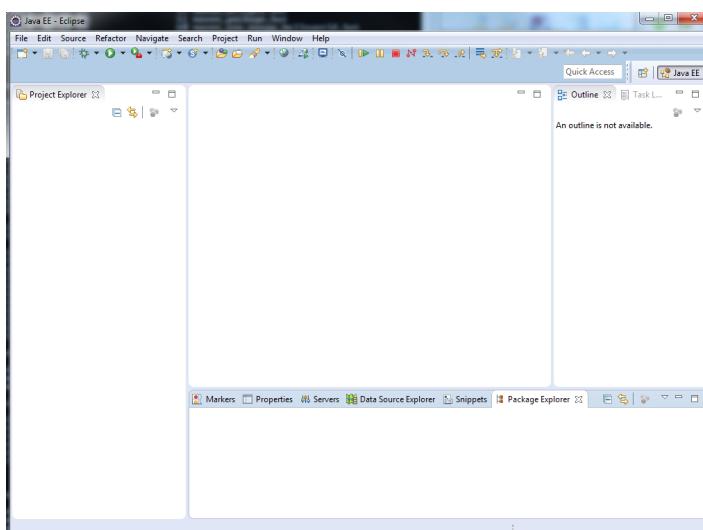
Go to eclipse ->Windows ->Show View

Other ->Java ->Package Explorer

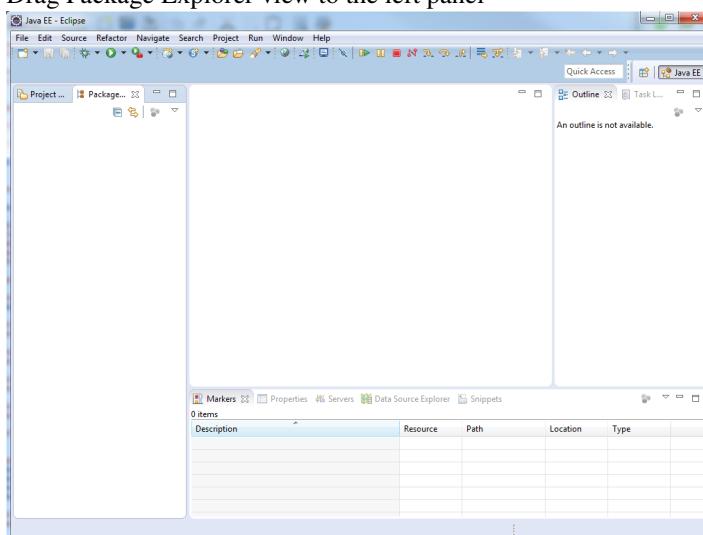
44 INSTALLATION AND CONFIGURATION



Click OK



Drag Package Explorer view to the left panel



1.8 Maven Central

When you build a Mavens project, Maven will check your pom.xml file, to identify which dependency to download.

First, Maven will get the dependency from your local repository Maven local repository, if not found, then get it from the default Maven central repository <http://repo1.maven.org/maven2/>
Search maven central <http://search.maven.org/>

This is how the Maven Central repository website looks like :

GroupID	ArtifactID	Latest Version	Updated	Download
org.tychelevel	junit	2.11.7	14-Aug-2015	pom jar javadoc.jar sources.jar
com.github.adedavy.intellijsdk	junit	142.1	13-Jul-2015	pom jar javadoc.jar sources.jar
junit	junit	4.12 all (24)	04-Dec-2014	pom jar javadoc.jar sources.jar
org.technibolts	junit	1.0.1	14-May-2014	pom jar javadoc.jar sources.jar
com.orougon.robojure.robojureclient	junit	0.4 all (3)	08-Jan-2013	pom jar javadoc.jar sources.jar
org.apache.tuscany.sca.samples	junit	2.0	18-Jun-2012	pom jar sources.jar
org.apache.isis.viewer	junit	0.2-incubating all (2)	10-Feb-2012	pom jar javadoc.jar site.xml sources.jar test-sources.jar tests.jar
com.kenai.npower	junit	4.7-201002241900	26-Feb-2010	pom jar nbm tests.jar
org.mod4j.org	junit	3.8.2	14-Aug-2009	pom jar
org.codehaus.mevenide	junit	3.1.4 all (5)	05-Aug-2008	pom jar nbm javadoc.jar sources.jar
org.eclipse.jdt	junit	3.3.0-20070605-0010 all (2)	27-Nov-2007	pom jar
junit	junit-dep	4.11 all (11)	14-Nov-2012	pom
junit-addons	junit-addons	1.4	11-Mar-2009	pom jar
junit-dodet	junit-dodet	1.0.2 all (2)	08-Nov-2008	pom jar
org.jboss.arquillian.junit	arquillian-junit-parent	1.1.9.Final all (26)	01-Sep-2015	pom tests.jar
io.fares.junit.soapui	soapui-junit-mockrunner-project	0.0.2	21-Apr-2015	pom
com.googlecode.quice-junit4	quice-junit4-parent	0.2 all (3)	21-Jun-2010	pom tests.jar
com.pholser	junit-quicheck	0.5-beta-1 all (12)	04-Sep-2015	pom
pl.javadevelopers.junit	parent	0.8 all (3)	03-Jul-2014	pom
ant	ant-junit	1.6.5 all (6)	08-Nov-2005	pom jar

Example: type in junit, then click Search

GroupID	ArtifactID	Latest Version	Updated	Download
org.tychelevel	junit	2.11.7	14-Aug-2015	pom jar javadoc.jar sources.jar
com.github.adedavy.intellijsdk	junit	142.1	13-Jul-2015	pom jar javadoc.jar sources.jar
junit	junit	4.12 all (24)	04-Dec-2014	pom jar javadoc.jar sources.jar
org.technibolts	junit	1.0.1	14-May-2014	pom jar javadoc.jar sources.jar
com.orougon.robojure.robojureclient	junit	0.4 all (3)	08-Jan-2013	pom jar javadoc.jar sources.jar
org.apache.tuscany.sca.samples	junit	2.0	18-Jun-2012	pom jar sources.jar
org.apache.isis.viewer	junit	0.2-incubating all (2)	10-Feb-2012	pom jar javadoc.jar site.xml sources.jar test-sources.jar tests.jar
com.kenai.npower	junit	4.7-201002241900	26-Feb-2010	pom jar nbm tests.jar
org.mod4j.org	junit	3.8.2	14-Aug-2009	pom jar
org.codehaus.mevenide	junit	3.1.4 all (5)	05-Aug-2008	pom jar nbm javadoc.jar sources.jar
org.eclipse.jdt	junit	3.3.0-20070605-0010 all (2)	27-Nov-2007	pom jar
junit	junit-dep	4.11 all (11)	14-Nov-2012	pom
junit-addons	junit-addons	1.4	11-Mar-2009	pom jar
junit-dodet	junit-dodet	1.0.2 all (2)	08-Nov-2008	pom jar
org.jboss.arquillian.junit	arquillian-junit-parent	1.1.9.Final all (26)	01-Sep-2015	pom tests.jar
io.fares.junit.soapui	soapui-junit-mockrunner-project	0.0.2	21-Apr-2015	pom
com.googlecode.quice-junit4	quice-junit4-parent	0.2 all (3)	21-Jun-2010	pom tests.jar
com.pholser	junit-quicheck	0.5-beta-1 all (12)	04-Sep-2015	pom
pl.javadevelopers.junit	parent	0.8 all (3)	03-Jul-2014	pom
ant	ant-junit	1.6.5 all (6)	08-Nov-2005	pom jar

Click junit / junit / Latest Version 4.12

The Central Repository

SEARCH | ADVANCED SEARCH | BROWSE | QUICK STATS

junit

Advanced Search | API Guide | Help

Browse Central For [junit : junit : 4.12](#)

Click on a link above to browse the repository.

Name	Last Modified	Size	SHA1 Checksum
junit-4.12.pom	04-Dec-2014	23.1 K	35fb238baee33af739074d723279ebea2028398
junit-4.12.pom.asc	04-Dec-2014	490 B	aef36e5e2417921a900bdba5d4555aa6ca87f4e2
junit-4.12.jar	04-Dec-2014	307.6 K	2973d150c0dc1fe998fb334810d68f278ea58ec
junit-4.12.jar.asc	04-Dec-2014	490 B	4e60e5af63e3373b40fcbd4c151c13e059151e00
junit-4.12-javadoc.jar	04-Dec-2014	916.0 K	941a8be4506c65f0a9001c08812fb7da1e505e21
junit-4.12-javadoc.jar.asc	04-Dec-2014	490 B	371def617b454a12b0a42146fb5fd0050c4c066
junit-4.12-sources.jar	04-Dec-2014	195.7 K	a6c32b40bf3d76eca54e3c601e5d1470c8fcdfa
junit-4.12-sources.jar.asc	04-Dec-2014	490 B	512deac91b234bff987ecbb823e47dc74b7b89de

Apache Maven Resources | About Sonatype | Privacy Policy | Terms of Service
Apache and Apache Maven are trademarks of the Apache Software Foundation. The Central Repository is a service mark of Sonatype, Inc. The Central Repository is intended to complement Apache Maven and should not be confused with Apache Maven. Copyright ©2011 Sonatype, Inc.

Inside Apache Maven area, that is what you need put in your pom.xml:

```

1 <dependency>
2   <groupId>junit</groupId>
3   <artifactId>junit</artifactId>
4   <version>4.12</version>
5 </dependency>
```

CHAPTER 2

SIMPLE PROJECTS

2.1 Simple maven project - app-hello-world

2.1.1 Create project using command line

2.1.1.1 Create project

```
| c:\workspaces\workspace\my-projects>mvn archetype:generate -DgroupId=com.xyz  
|   -DartifactId=app-hello-world  
|   -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

or

```
| c:\workspaces\workspace\my-projects>mvn  
|   org.apache.maven.plugins:maven-archetype-plugin:2.3:generate  
|   -DgroupId=com.xyz -DartifactId=app-hello-world  
|   -DarchetypeGroupId=org.apache.maven.archetypes  
|   -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.1  
|   -DinteractiveMode=false
```

2.1.1.2 Build project

```
c:\workspaces\workspace\my-projects\app-hello-world>mvn package
```

[†]change c:\workspaces\workspace to /users/xyz/documents/workspace, then all the projects in this chapter will work in mac osx.

2.1.1.3 Run the app

```
1 c:\workspaces\ eclipse\my-projects\app-hello-world>java -cp
  ↳ target/app-hello-world-1.0-SNAPSHOT.jar com.xyz.App
```

Result:

```
1 c:\workspaces\ eclipse\my-projects\app-hello-world>java -cp
  ↳ target/app-hello-world-1.0-SNAPSHOT.jar com.xyz.App
2 Hello World!
```

2.1.1.4 Make this project an eclipse project

```
1 c:\workspaces\ eclipse\my-projects\app-hello-world>mvn eclipse:clean
  ↳ eclipse:eclipse
```

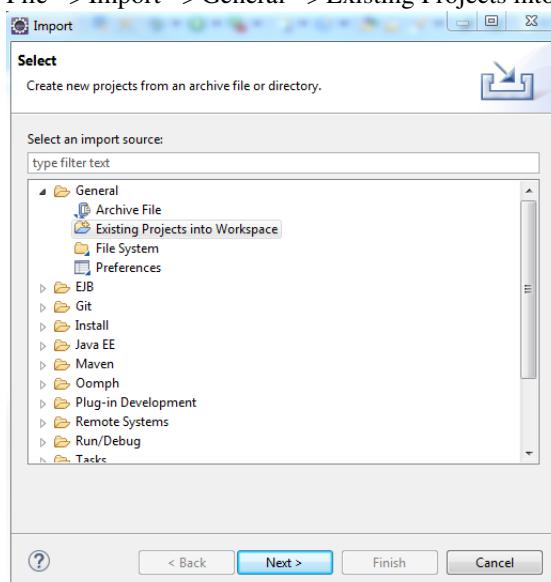
or

```
1 c:\workspaces\ eclipse\my-projects\app-hello-world>mvn
  ↳ org.apache.maven.plugins:maven-eclipse-plugin:2.10:eclipse
```

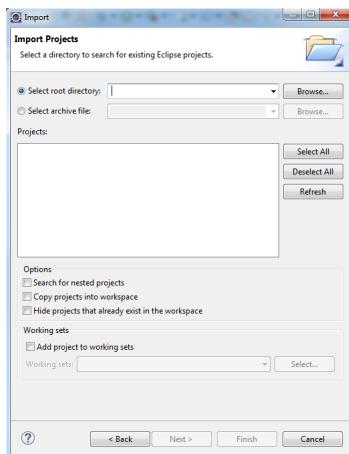
2.1.1.5 Import this project to eclipse

- Launch eclipse

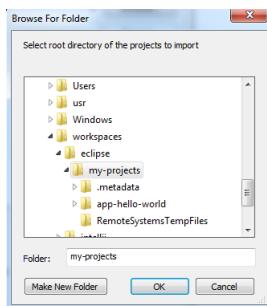
File → Import → General → Existing Projects into Workspace



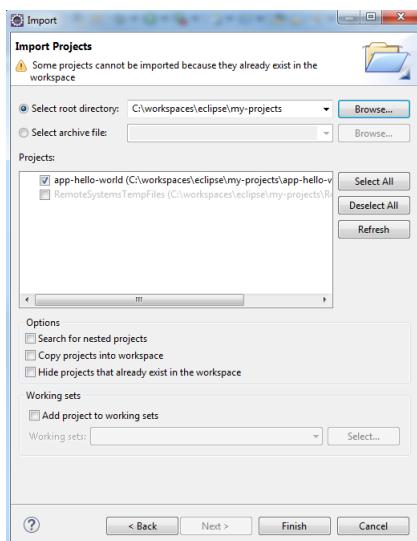
Click Next



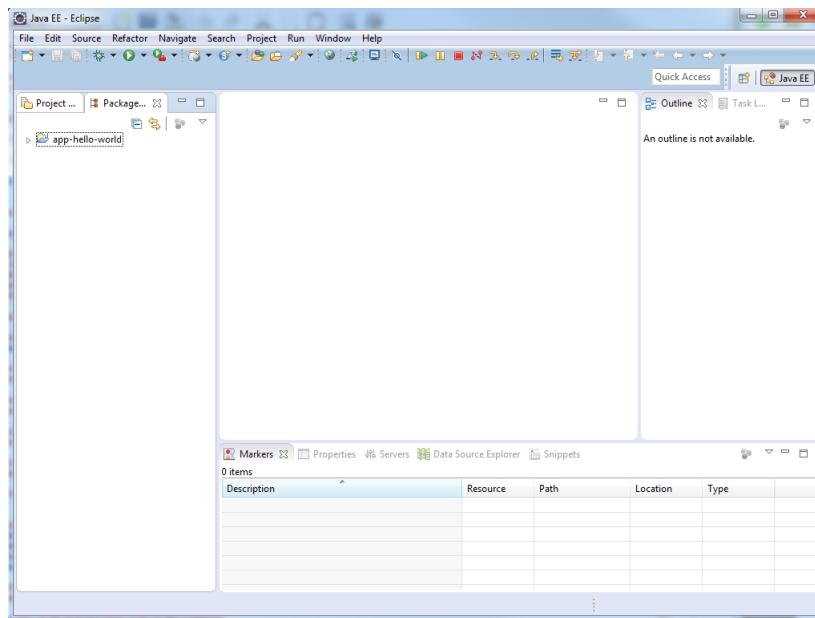
Click Browse...



Default should be c:\workspaces\workspace\my-projects
Click OK

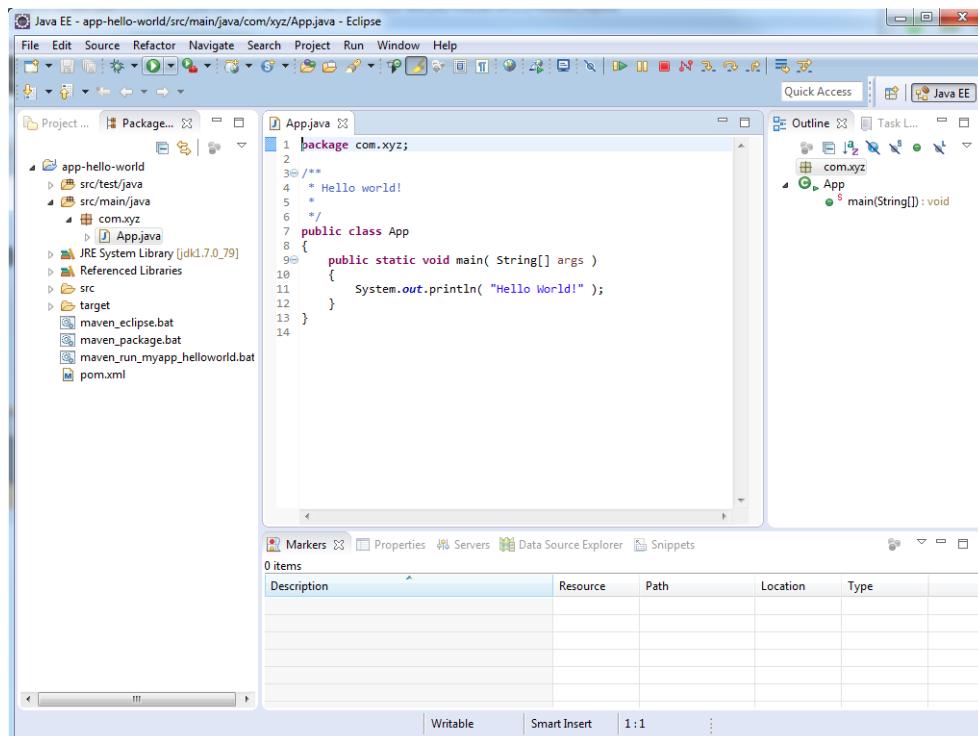


Click Finish

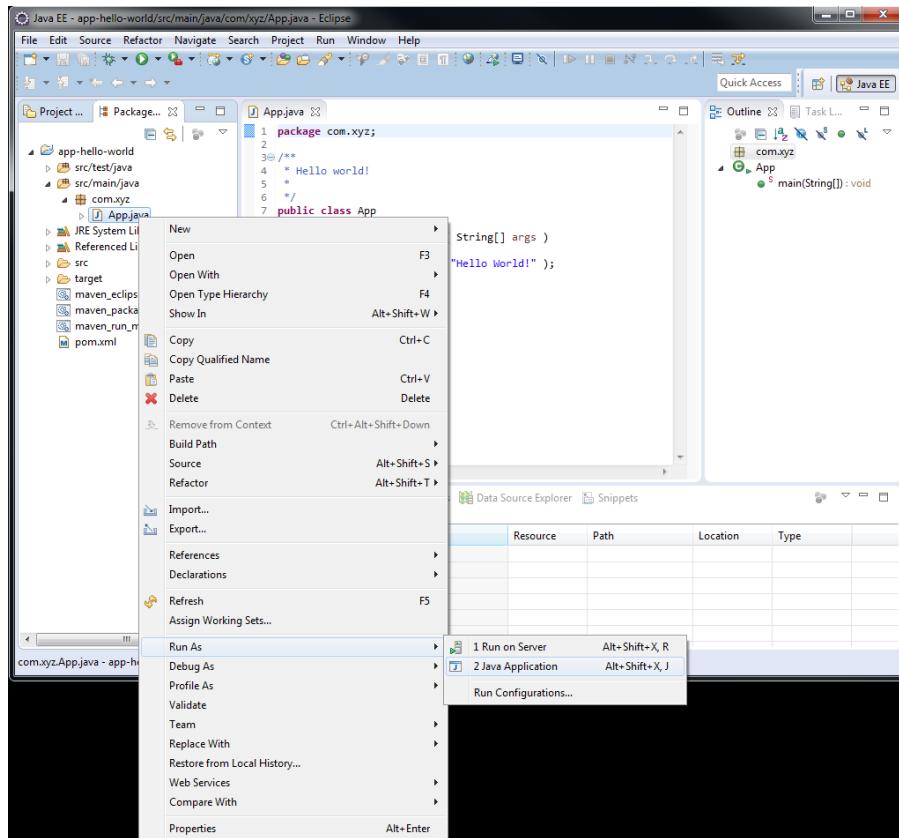


2.1.1.6 Run this project in eclipse

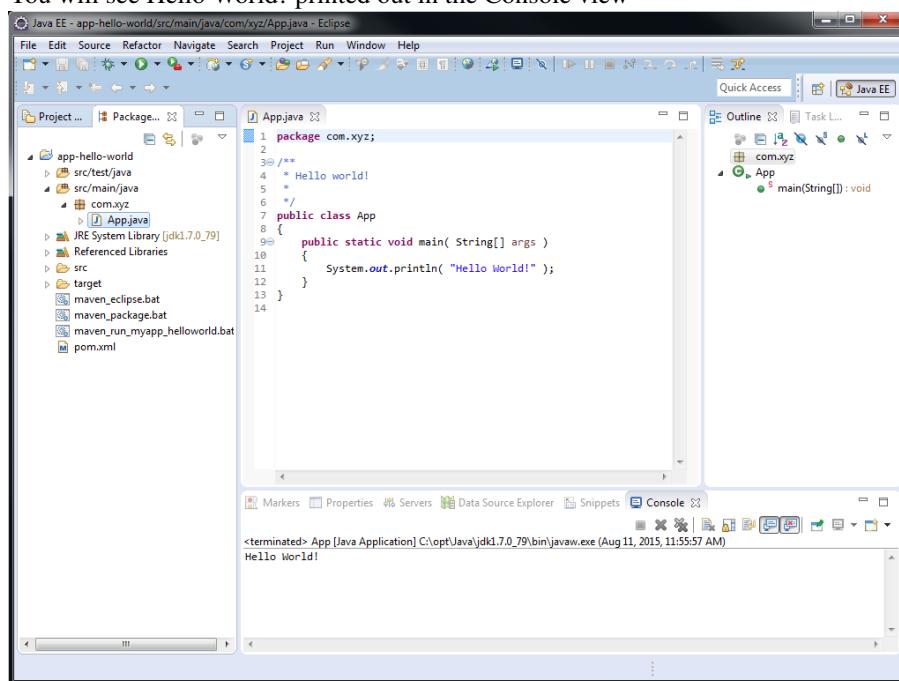
- Browse to app-hello-world ->src/main/java ->com.xyz ->App.java



Right click App.java
Select Run As ->Java Application



You will see Hello World! printed out in the Console view



- Browse to app-hello-world ->src/test/java ->com.xyz ->AppTest.java

```

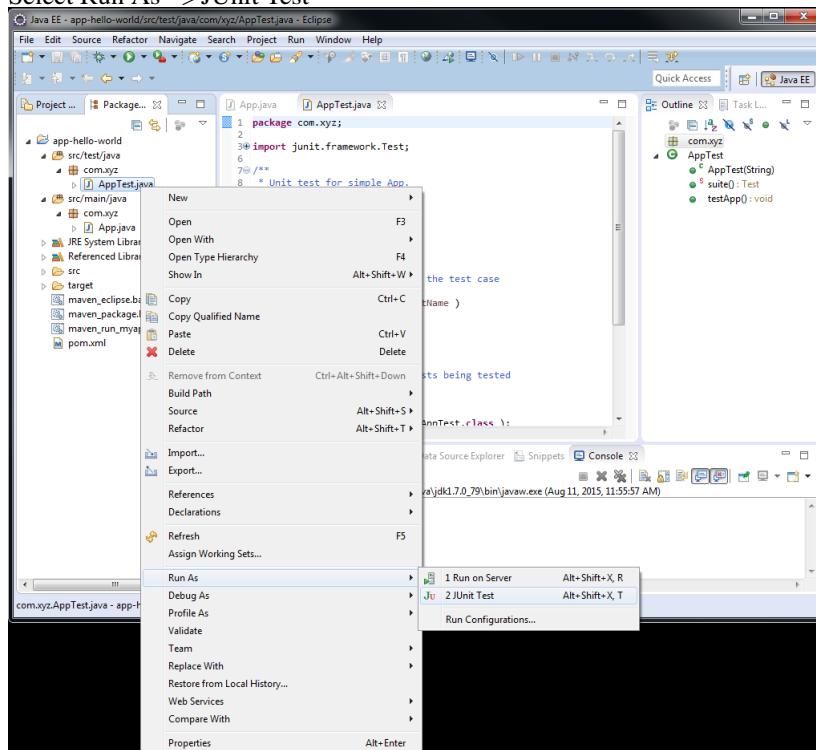
package com.xyz;
import junit.framework.TestCase;
public class AppTest extends TestCase {
    /**
     * Create the test case
     * @param testName name of the test case
     */
    public AppTest( String testName ) {
        super( testName );
    }
    /**
     * @return the suite of tests being tested
     */
    public static Test suite() {
        return new TestSuite( AnnText.class );
    }
}

```

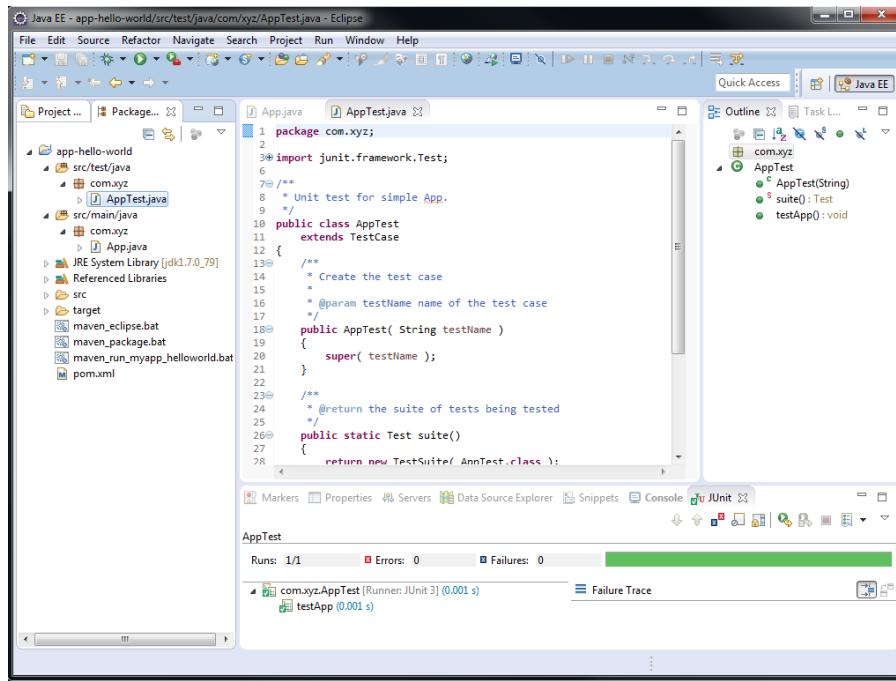
<terminated> App [Java Application] C:\opt\Java\jdk1.7.0_79\bin\javaw.exe (Aug 11, 2015, 11:55:57 AM)
Hello World!

Right click AppTest.java

Select Run As ->JUnit Test

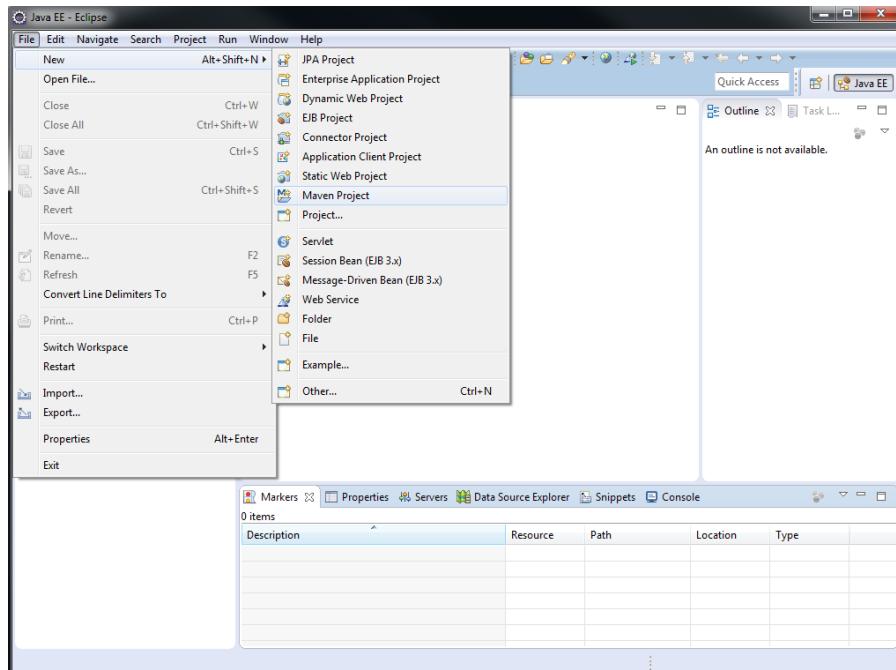


You will see JUnit results in the JUnit view



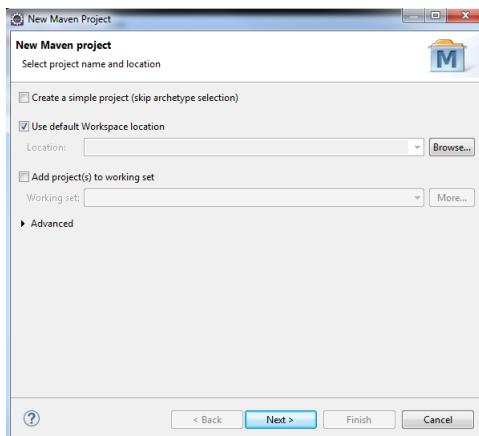
2.1.2 Create project using Eclipse

- Launch eclipse

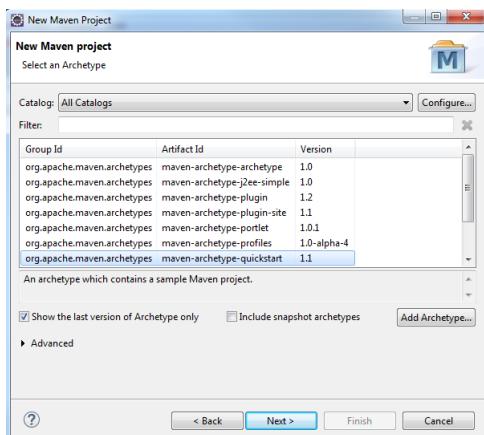


File → New → Maven Project

54 SIMPLE PROJECTS

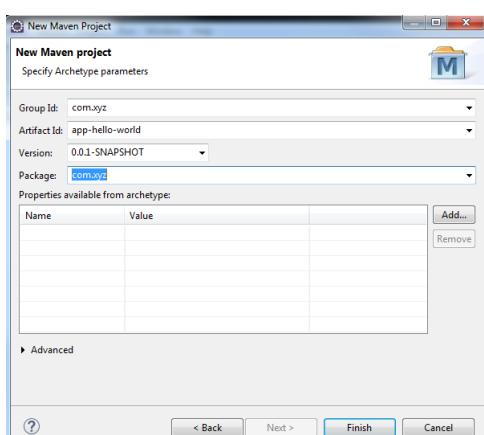


Click Next



Select org.apache.maven.archetypes / maven-archetype-quickstart / 1.1

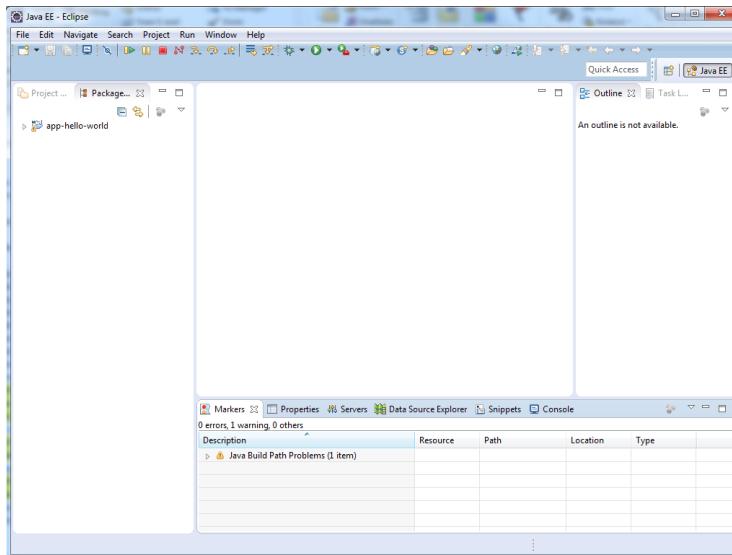
Click Next



Type in:

Group Id: com.xyz

Artifact id: app-hello-world
 Version: 0.0.1-SNAPSHOT
 Package: com.xyz
 Click Finish



- Run Application and JUnit Test

You can run Application and JUnit Test similar to the previous section.

2.2 Simple web maven project - hello-world

2.2.1 Create project

```
1 c:\workspaces\workspace\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=hello-world -DarchetypeArtifactId=maven-archetype-webapp
  ↳ -DinteractiveMode=false
```

or

```
1 c:\workspaces\workspace\my-projects>mvn
  ↳ org.apache.maven.plugins:maven-archetype-plugin:2.3:generate
  ↳ -DgroupId=com.xyz -DartifactId=hello-world
  ↳ -DarchetypeGroupId=org.apache.maven.archetypes
  ↳ -DarchetypeArtifactId=maven-archetype-webapp -DarchetypeVersion=1.0
  ↳ -DinteractiveMode=false
```

2.2.2 Make this project an eclipse project

```
1 c:\workspaces\workspace\my-projects\hello-world>mvn eclipse:clean eclipse:eclipse
  ↳ -Dwtpversion=2.0
```

or

```

1 c:\workspaces\ eclipse\my-projects\hello-world>mvn
  → org.apache.maven.plugins:maven-eclipse-plugin:2.10:eclipse -Dwtpversion=2.0

```

2.2.3 Update JDK version

- Maven 2.x/3.x generated web application with JDK1.4, which is rather outdated, you may need to upgrade it to latest JDK version.

- Method 1

Update eclipse project xml settings file

C:\workspaces\ eclipse\my-projects\hello-world\.settings\org.eclipse.wst.common.project.facet.core.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <faceted-project>
3   <fixed facet="jst.java"/>
4   <fixed facet="jst.web"/>
5   <installed facet="jst.web" version="2.4"/>
6   <installed facet="jst.java" version="1.4"/>
7 </faceted-project>

```

Update JDK to:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <faceted-project>
3   <fixed facet="jst.java"/>
4   <fixed facet="jst.web"/>
5   <installed facet="jst.web" version="2.4"/>
6   <installed facet="jst.java" version="7.0"/>
7 </faceted-project>

```

- Method 2

Update project pom.xml file, add maven-compiler-plugin

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  → xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  → http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>hello-world</artifactId>
6   <packaging>war</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>hello-world Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10  <dependencies>
11    <dependency>
12      <groupId>junit</groupId>
13      <artifactId>junit</artifactId>
14      <version>3.8.1</version>
15      <scope>test</scope>
16    </dependency>
17  </dependencies>
18  <build>
19    <finalName>hello-world</finalName>
20    <plugins>
21      <plugin>
22        <groupId>org.apache.maven.plugins</groupId>
23        <artifactId>maven-compiler-plugin</artifactId>
24        <version>3.3</version>
25        <configuration>
26          <source>1.7</source>
27          <target>1.7</target>
28        </configuration>
29      </plugin>

```

```

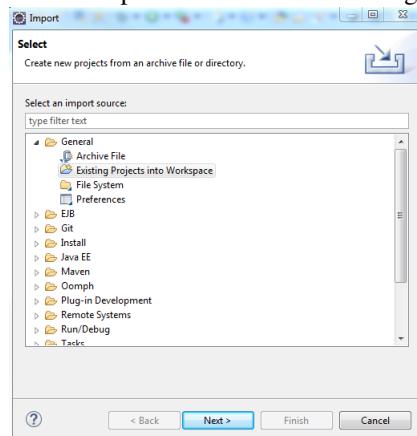
30      </plugins>
31    </build>
32  </project>
```

Build this project using maven

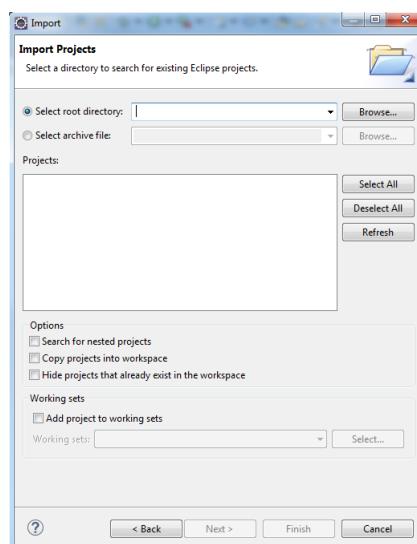
```
c:\workspaces\workspace\my-projects\hello-world>mvn install
```

2.2.4 Import this project to eclipse

- Launch eclipse
- File → Import → General → Existing Projects into Workspace

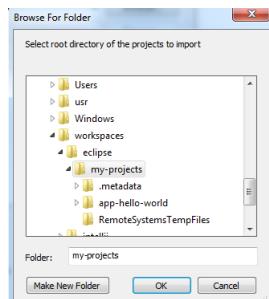


Click Next

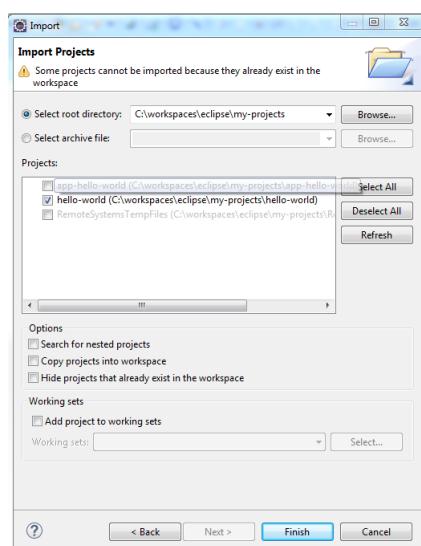


Click Browse...

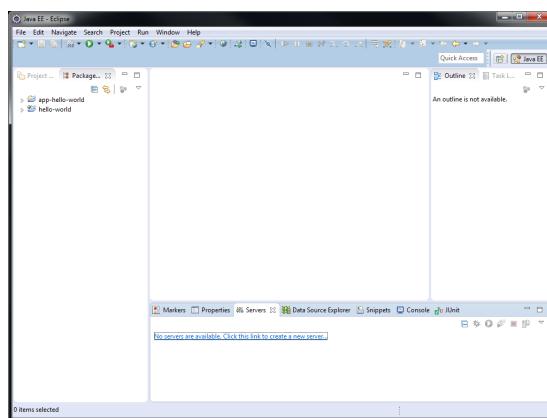
58 SIMPLE PROJECTS



Default should be c:\workspaces\.eclipse\my-projects
Click OK



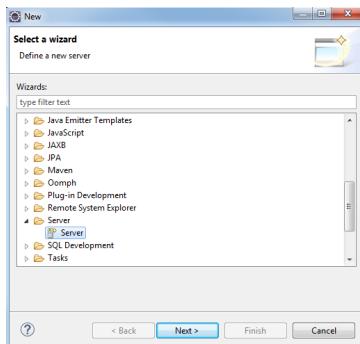
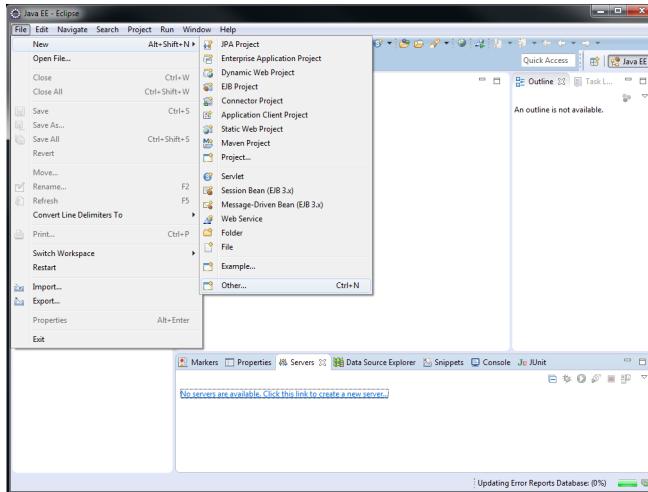
Click Finish



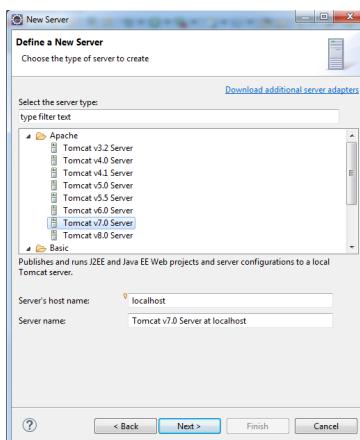
2.2.5 Add Tomcat server to eclipse

▪ eclipse

File → New → Other

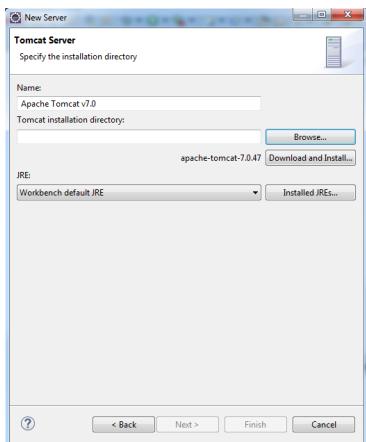


Server → Server

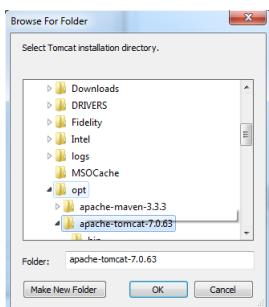


Select Apache → Tomcat v7.0 Server

Click Next

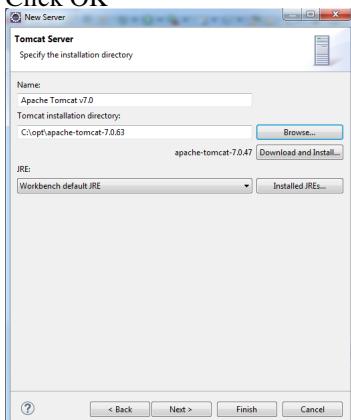


Click Browse...

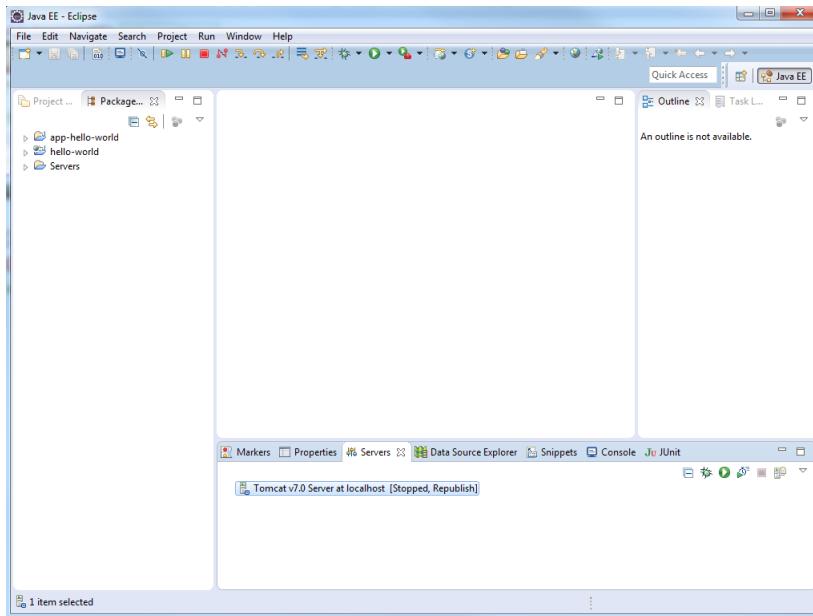


Select C:\opt\apache-tomcat-7.0.63

Click OK



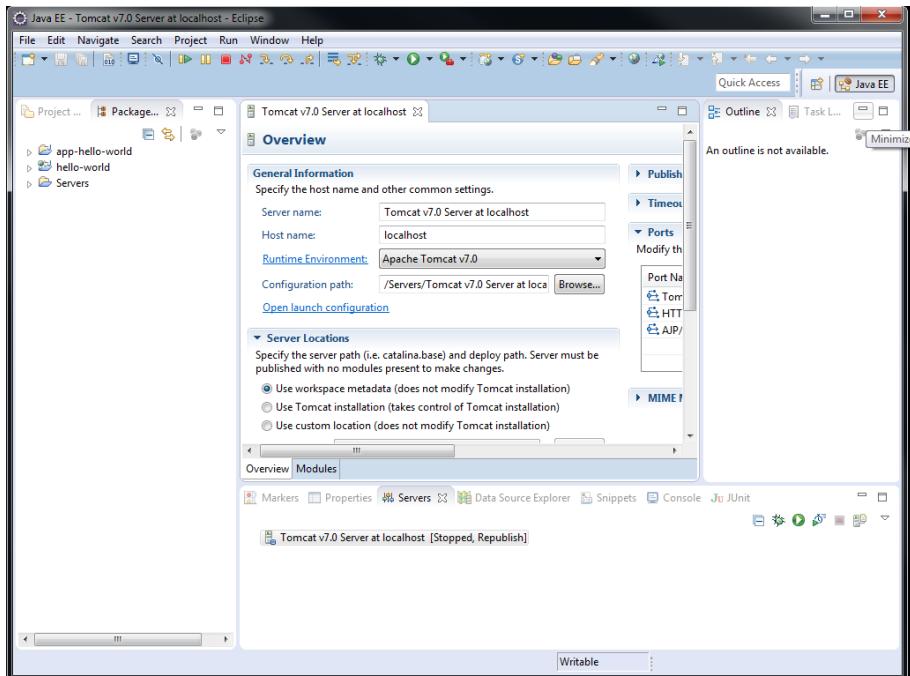
Click Finish



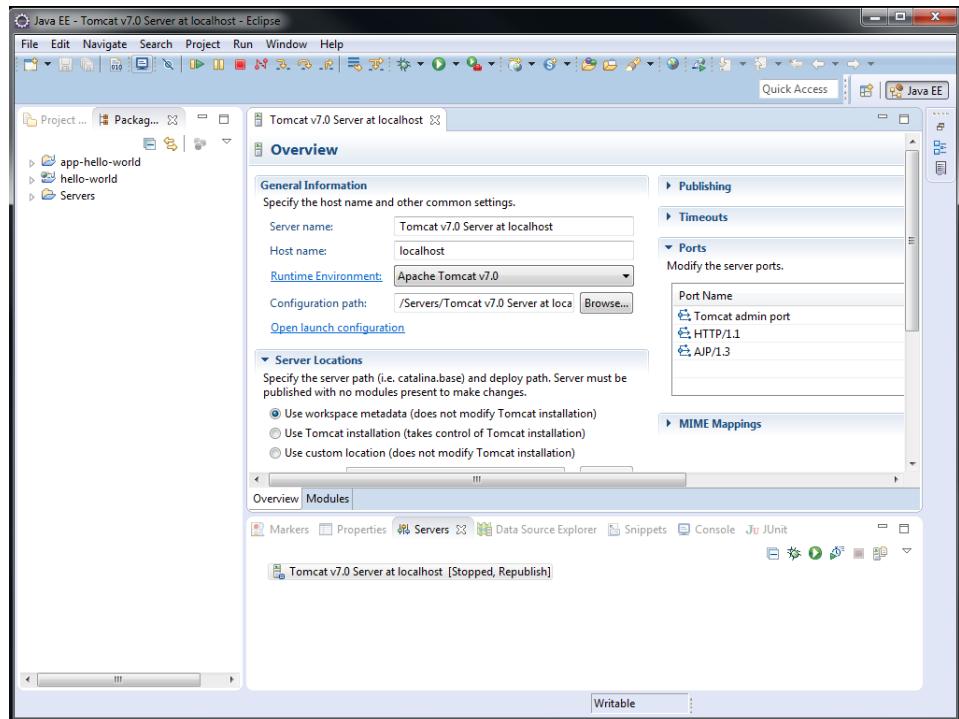
2.2.6 Configure Tomcat server

- **eclipse**

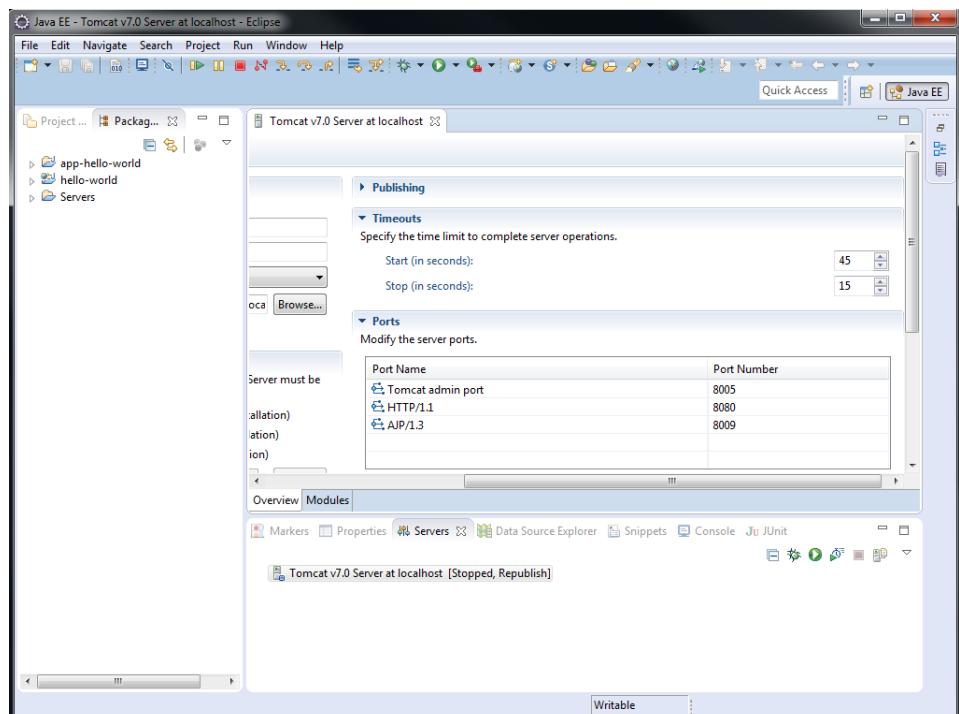
Double click Servers → Tomcat v7.0 Server at localhost[Stopped, Republish]



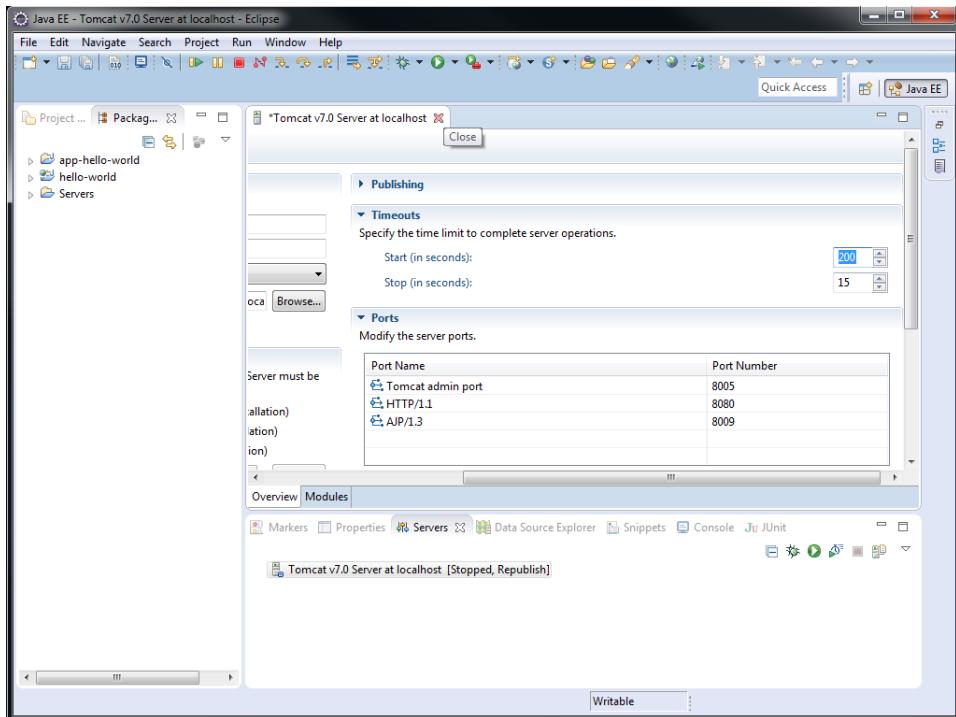
Click Minimize on the top-right corner to minimize Outline area.



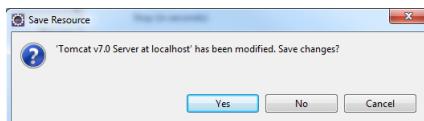
Expand Timeouts



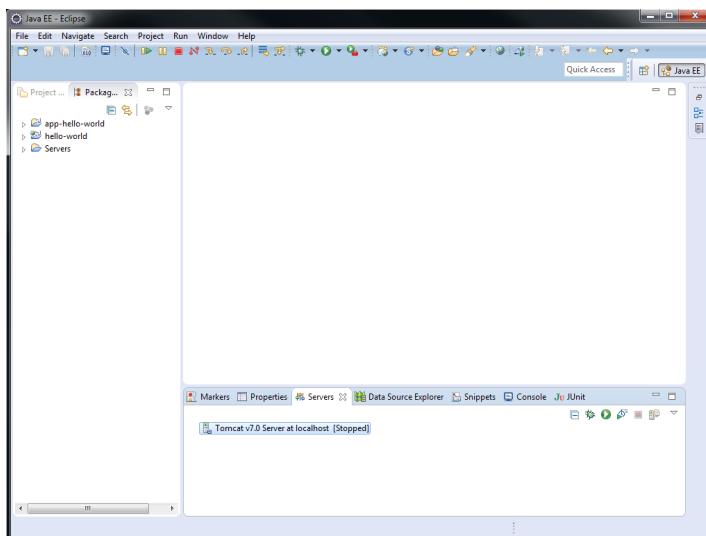
Change Start (in seconds) from 45 to 200



Click the Close mark at Tomcat v7.0 Server at localhost



Click Yes



- This actually changed:

C:\workspaces\clipse\my-projects\.metadata\.plugins\org.eclipse.wst.server.core\servers.xml

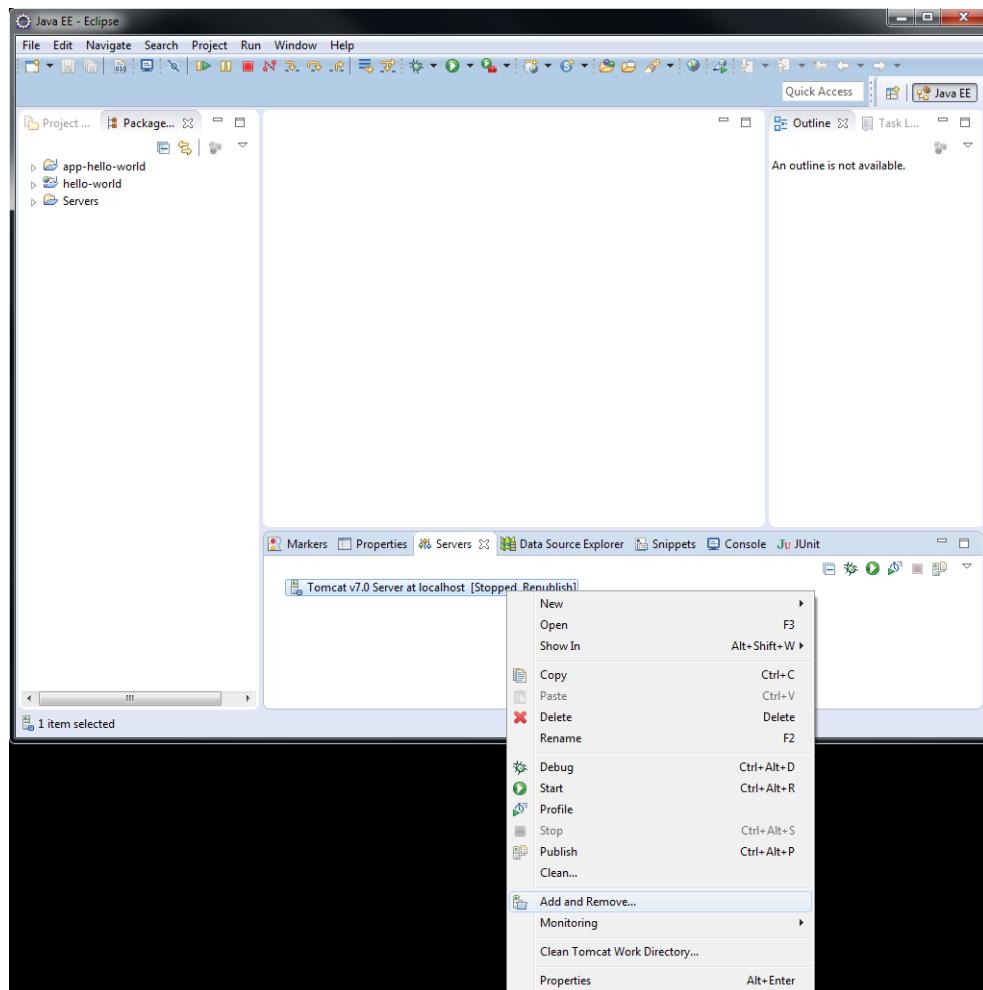
```

1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <servers>
3   <server auto-publish-setting="2" auto-publish-time="1"
4     configuration-id="/Servers/Tomcat v7.0 Server at localhost-config"
      deployDir="wtpwebapps" hostname="localhost" id="Tomcat v7.0 Server at
      localhost" name="Tomcat v7.0 Server at localhost" runtime-id="Apache
      Tomcat v7.0" server-type="org.eclipse.jst.server.tomcat.70"
      server-type-id="org.eclipse.jst.server.tomcat.70" start-timeout="200"
      stop-timeout="15" testEnvironment="true" timestamp="1"/>
4 </servers>
```

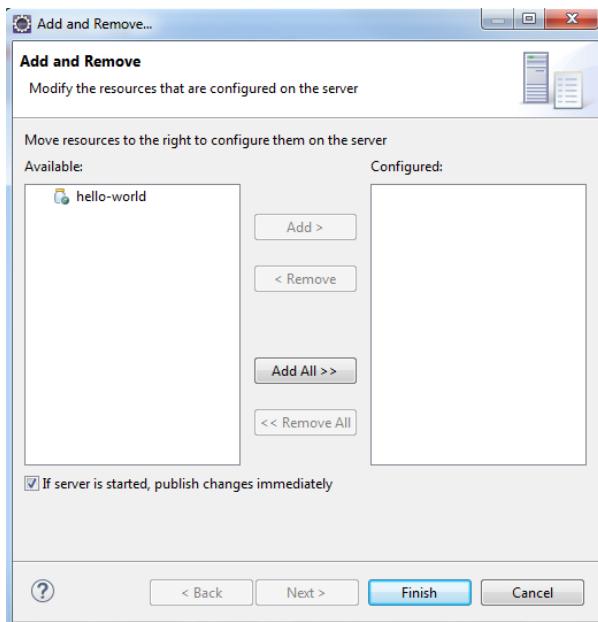
2.2.7 Add hello-world to Tomcat server

- eclipse

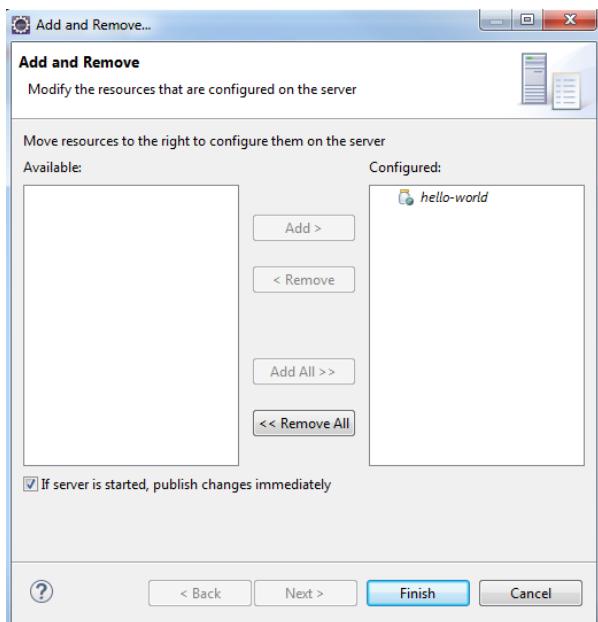
Right click Servers –>Tomcat v7.0 Server at localhost[Stopped, Republish]



Select Add and Remove...



Select hello-world from left panel, Click Add >, to add hello-world to the right panel.

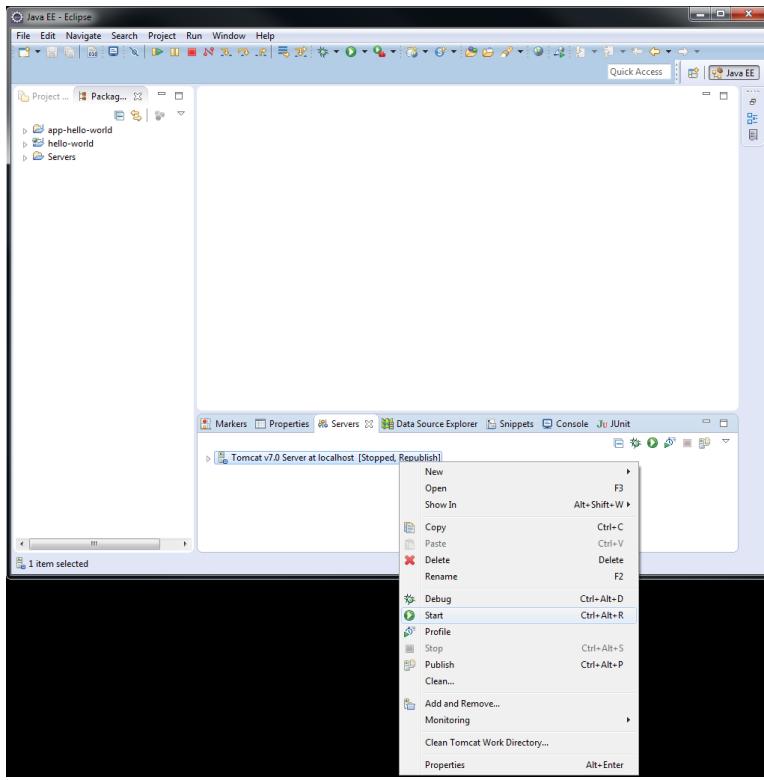


Click Finish

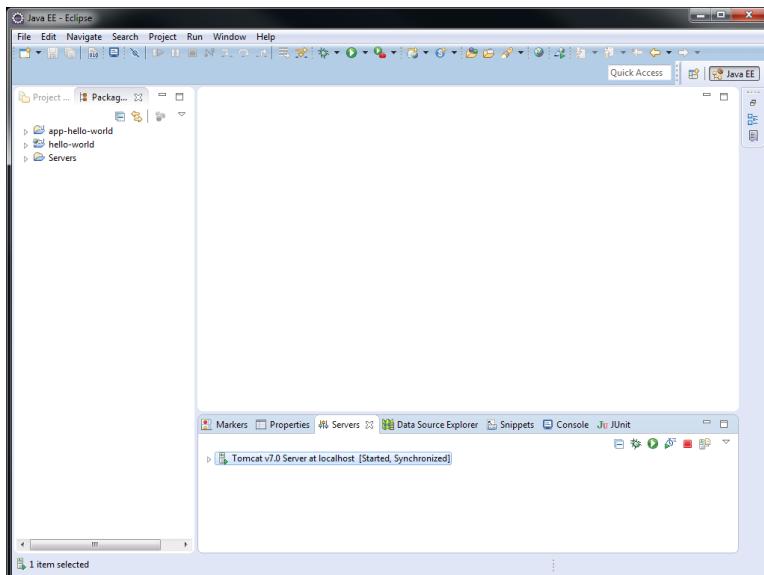
2.2.8 Run hello-world on Tomcat server

- eclipse
Right click Servers → Tomcat v7.0 Server at localhost[Stopped, Republish]

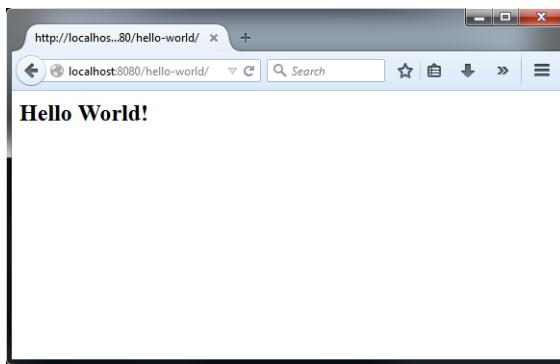
66 SIMPLE PROJECTS



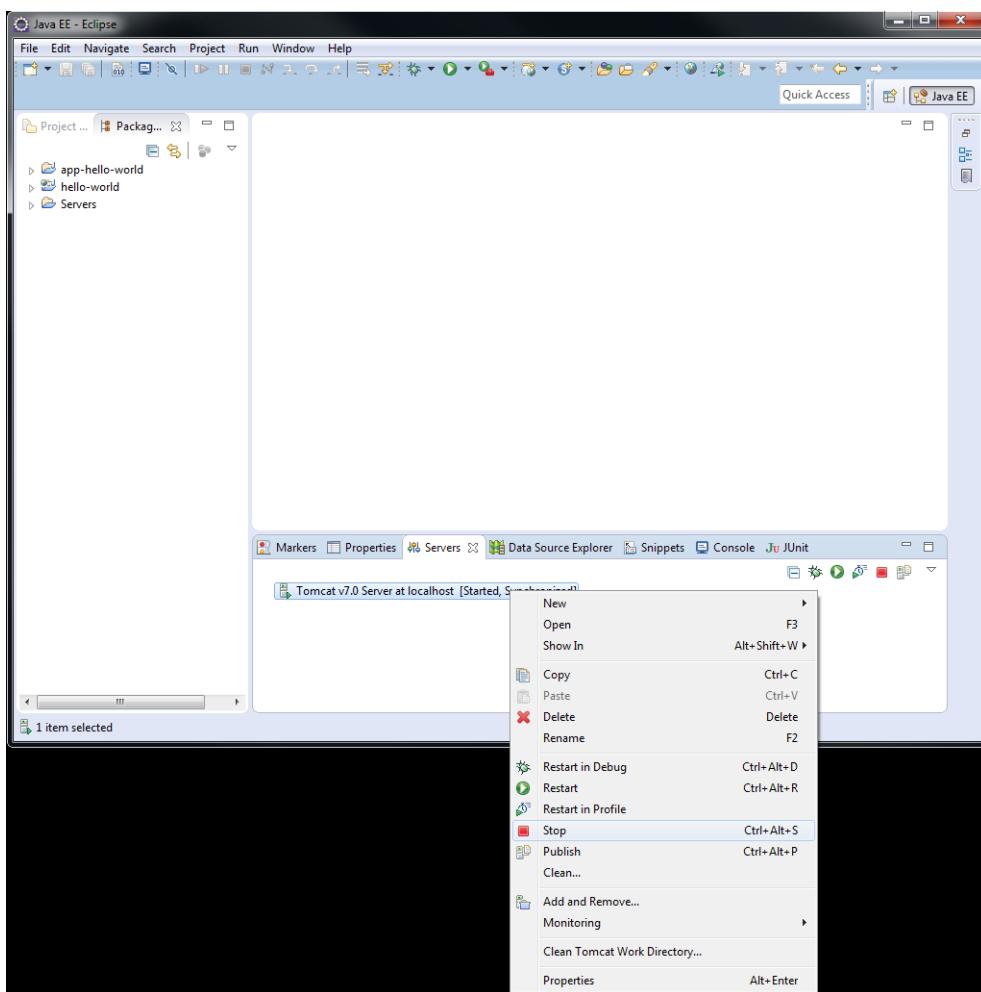
Select Start



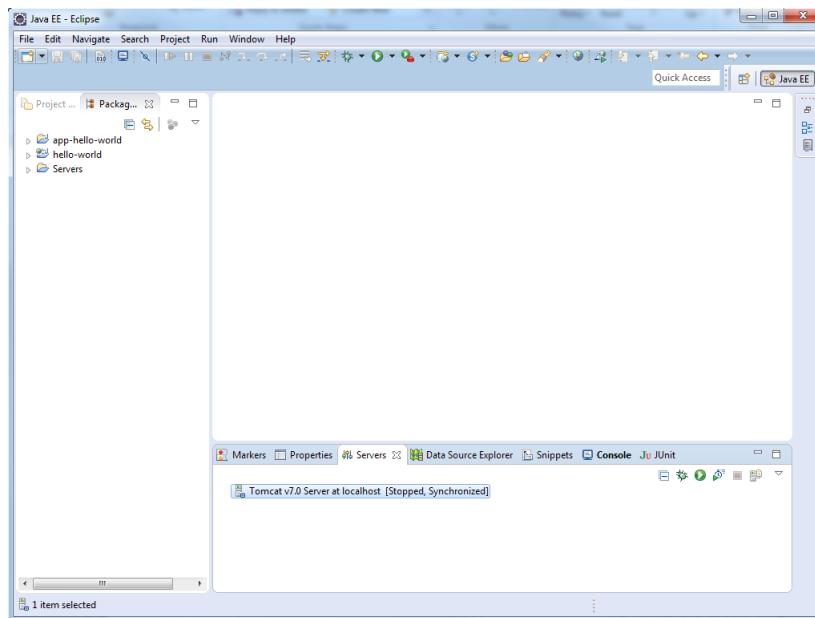
Open a browser, go to <http://localhost:8080/hello-world>



- Stop Tomcat Server



Right click Servers → Tomcat v7.0 Server at localhost[Stopped, Republish]
Select Stop



2.3 Simple spring maven project - app-hello-spring

2.3.1 Create project using command line

```
1 c:\workspaces\workspace\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=app-hello-spring
  ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

2.3.2 Modify pom.xml

- C:\workspaces\workspace\my-projects\app-hello-spring\pom.xml

Use properties
 Add spring dependencies
 Modify junit version
 Add maven-compiler-plugin, set JDK version to 1.7

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-spring</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-spring</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <java.version>1.7</java.version>
12    <junit.version>4.12</junit.version>
13    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
14    <spring.version>4.2.0.RELEASE</spring.version>
```

```

15  </properties>
16  <dependencies>
17      <dependency>
18          <groupId>org.springframework</groupId>
19          <artifactId>spring-context</artifactId>
20          <version>${spring.version}</version>
21      </dependency>
22      <dependency>
23          <groupId>org.springframework</groupId>
24          <artifactId>spring-core</artifactId>
25          <version>${spring.version}</version>
26      </dependency>
27      <dependency>
28          <groupId>junit</groupId>
29          <artifactId>junit</artifactId>
30          <version>${junit.version}</version>
31          <scope>test</scope>
32      </dependency>
33  </dependencies>
34  <build>
35      <finalName>app-hello-spring</finalName>
36      <plugins>
37          <plugin>
38              <groupId>org.apache.maven.plugins</groupId>
39              <artifactId>maven-compiler-plugin</artifactId>
40              <version>${maven.compiler.plugin.version}</version>
41              <configuration>
42                  <source>${java.version}</source>
43                  <target>${java.version}</target>
44              </configuration>
45          </plugin>
46      </plugins>
47  </build>
48</project>

```

2.3.3 Make this project an eclipse project

```
c:\workspaces\eclipse\my-projects\app-hello-spring>mvn eclipse:clean
→ eclipse:eclipse
```

Maven will download the Spring dependency libraries automatically and put it into your Maven's local repository (C:\opt\m2_repo).

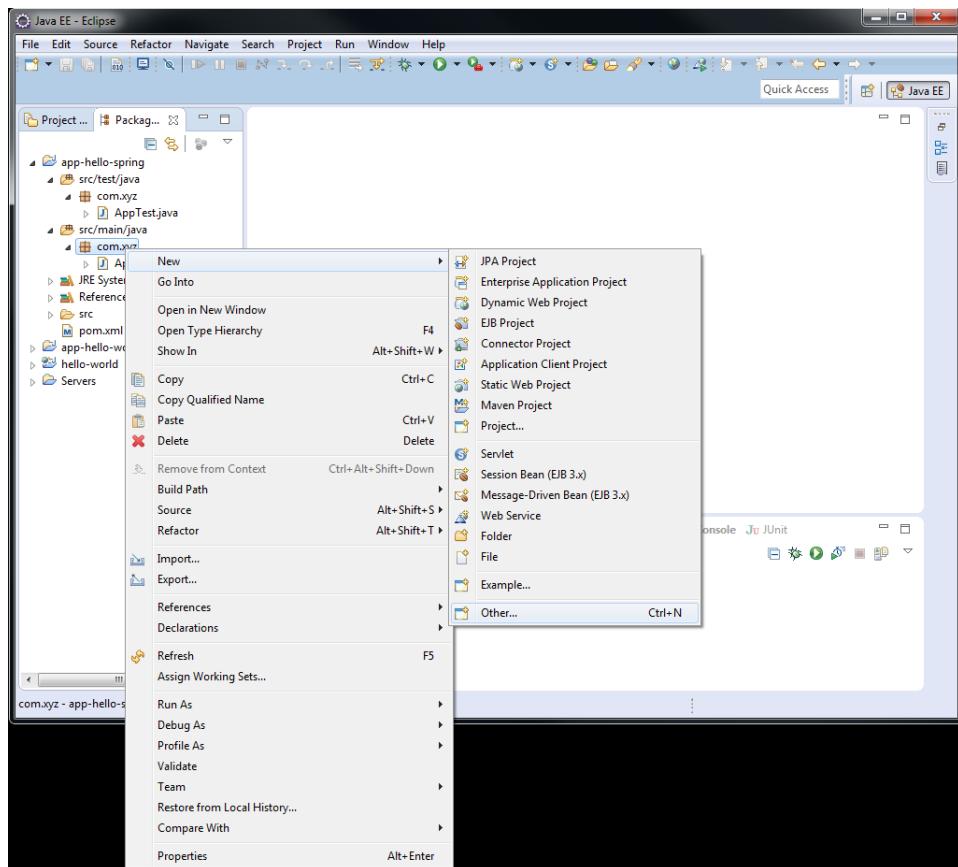
At the same time, Maven will add the downloaded libraries into Eclipse .classpath for dependency purpose

2.3.4 Import this project to eclipse

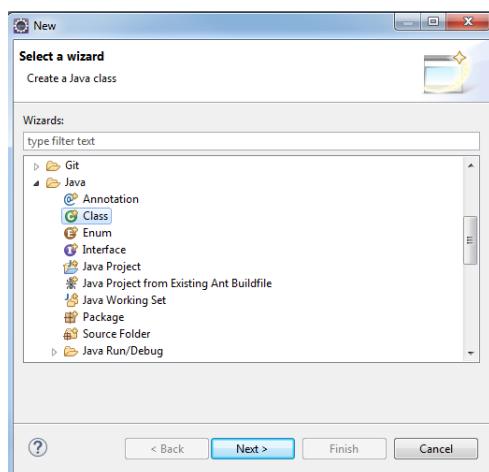
- Launch eclipse
- File → Import → General → Existing Projects into Workspace
- Browse to app-hello-spring, and import it.

2.3.5 Add a simple java bean

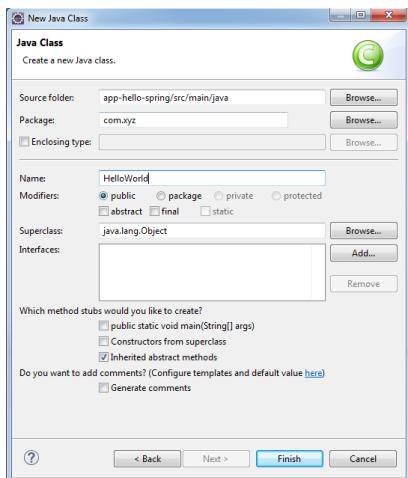
- eclipse
- Expand project to app-hello-spring → src/main/java → com.xyz



Right click com.xyz
New ->Other

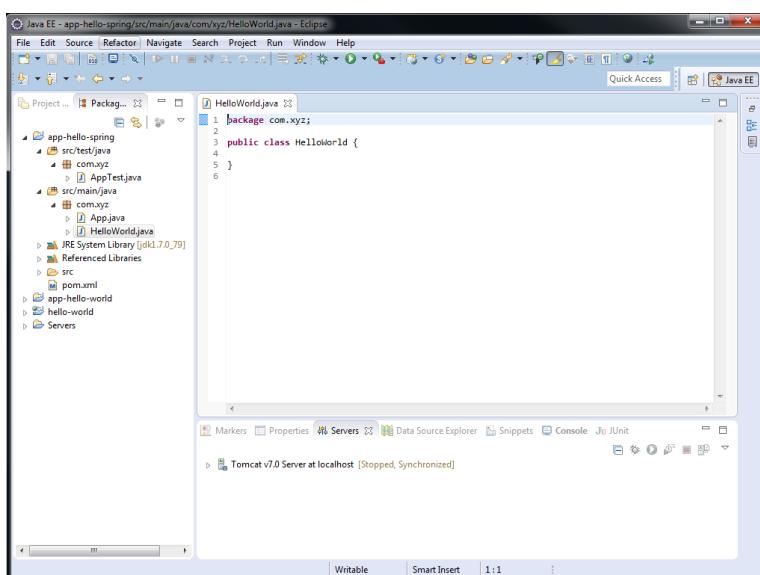


Select Java ->Class
Click Next



Type in Name: HelloWorld

Click Finish



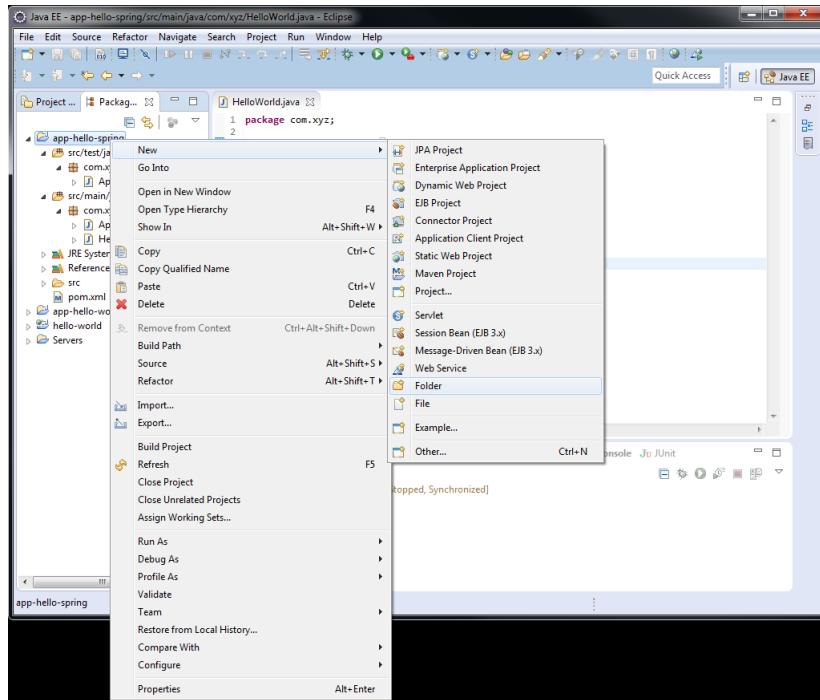
Then type in the code below:

```

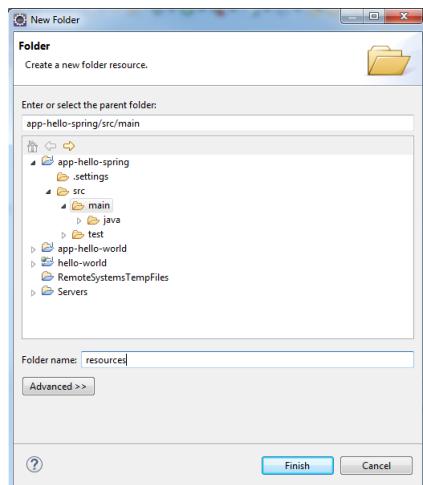
1 package com.xyz;
2
3 public class HelloWorld {
4     private String name;
5
6     public void setName(String name) {
7         this.name = name;
8     }
9
10    public void printHello() {
11        System.out.println("Spring 4 : Hello ! " + name);
12    }
13 }
```

2.3.6 Configure spring bean

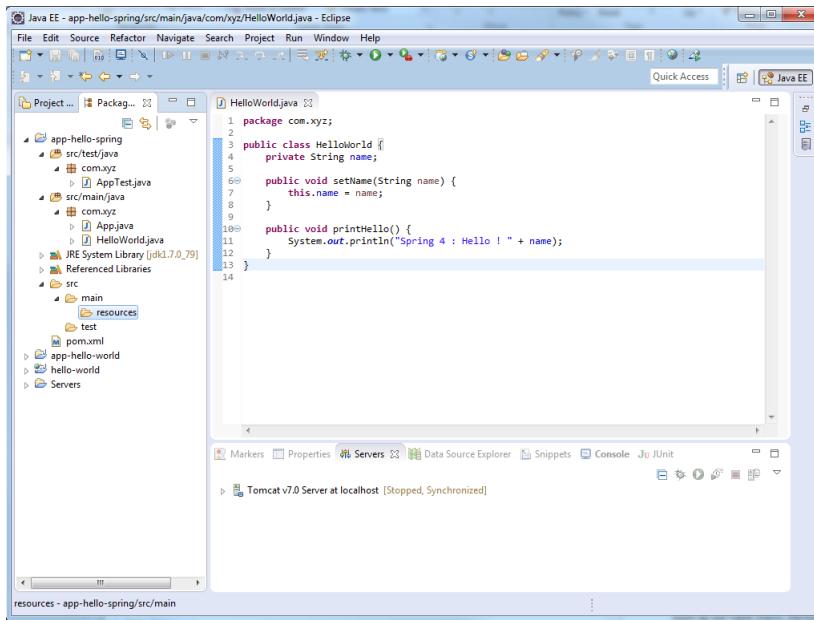
- Create resources directory



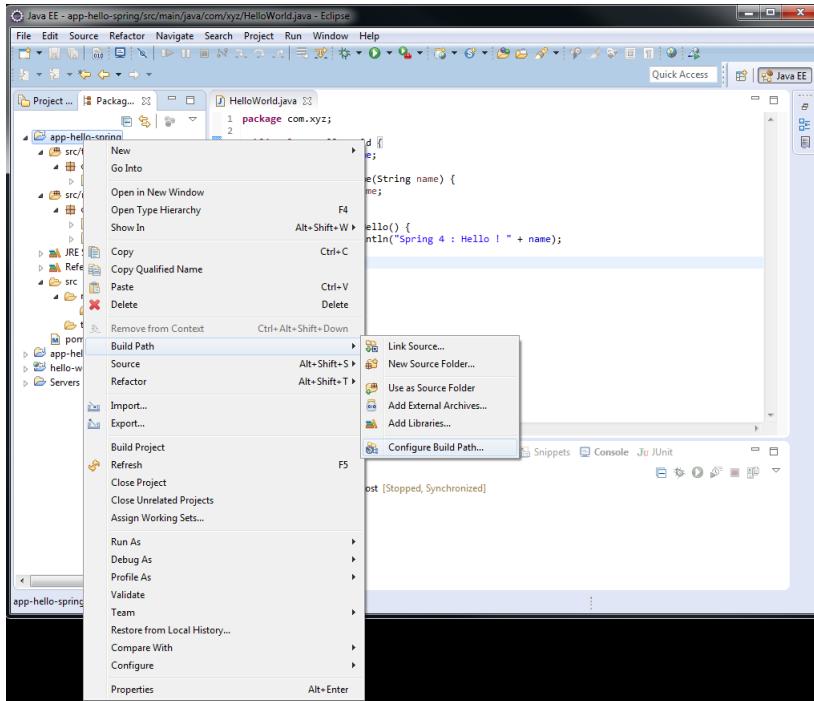
Right click on project app-hello-spring → New → Folder



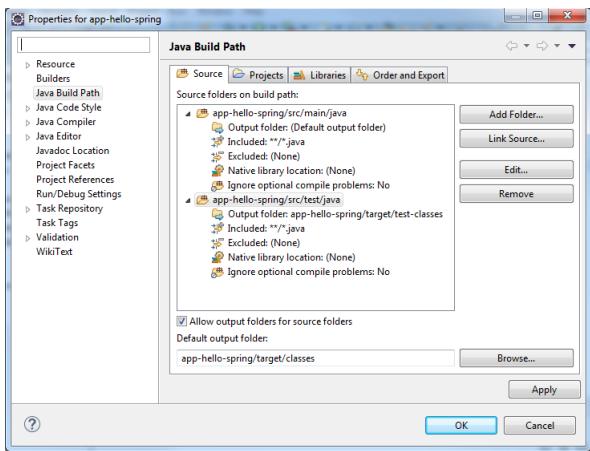
Select app-hello-spring → src → main
 Type in Folder name: resources
 Click Finish



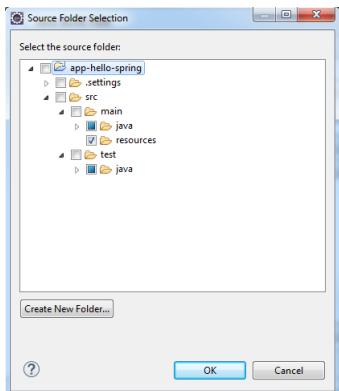
- Add resources directory to project build path



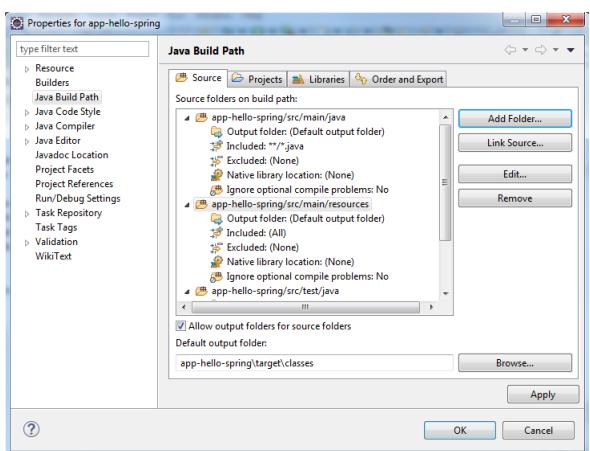
Right click on project app-hello-spring → Build Path → Configure Build Path



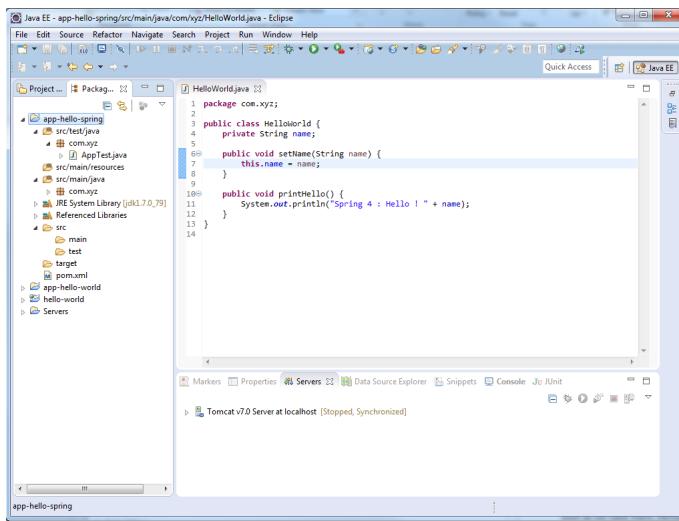
Click Source Tab → Add Folder



Check app-hello-spring → src → main → resources
Click OK



Click OK



▪ Configuration metadata

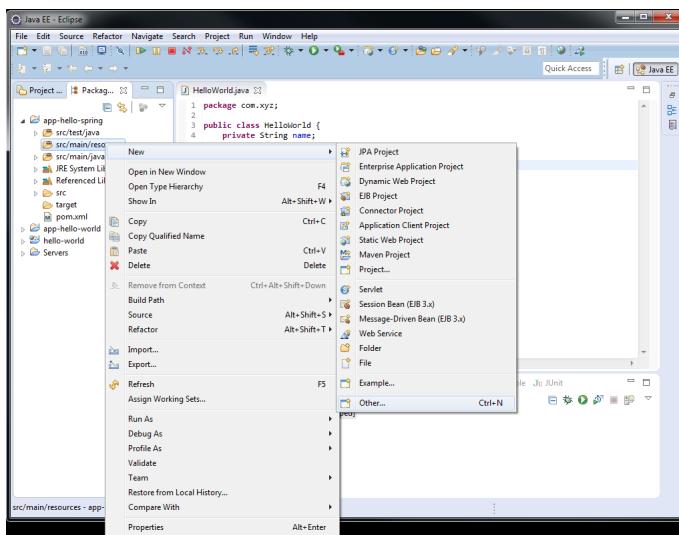
You can configure Spring metadata in three different ways:

XML-based configuration: Traditional configuration.

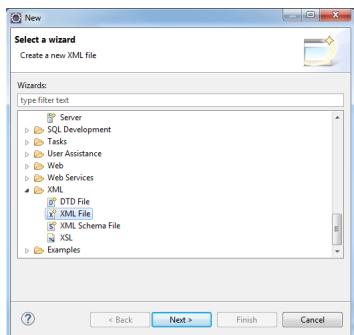
Annotation-based configuration: Spring 2.5 introduced support for annotation-based configuration metadata.

Java-based configuration: Starting with Spring 3.0, many features provided by the Spring JavaConfig project became part of the core Spring Framework. Thus you can define beans external to your application classes by using Java rather than XML files. To use these new features, see the @Configuration, @Bean, @Import and @Depend-
sOn annotations.

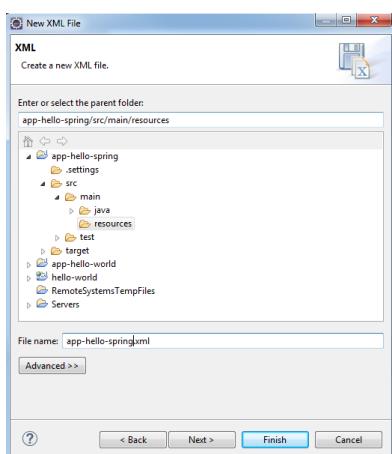
Let's Create XML-based configuration metadata:



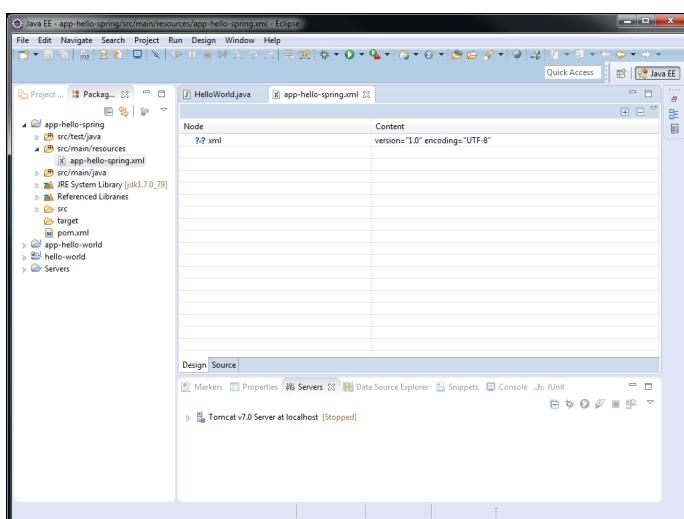
Right click on project src/main/resources ->New ->Other



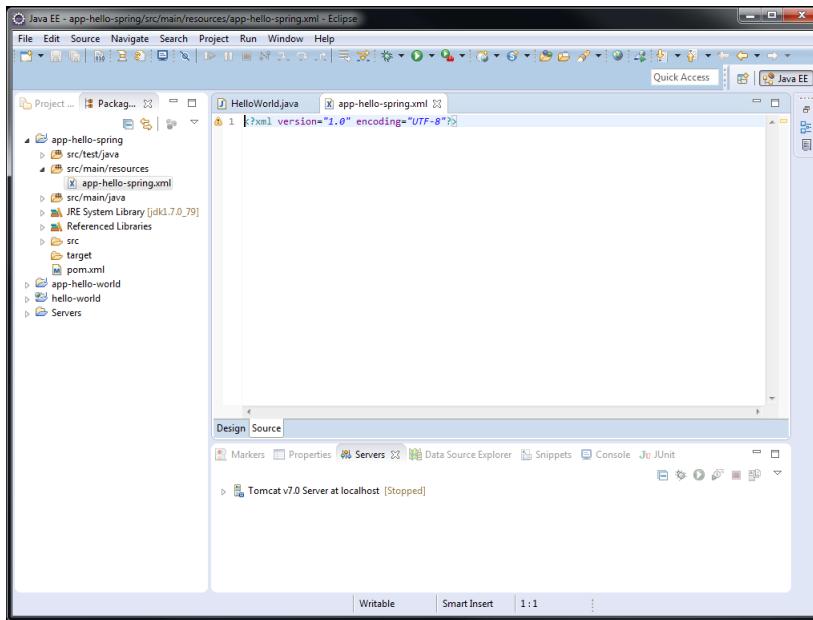
Select XML → XML File



Type in File name: app-hello-spring.xml
Click Finish



Click Source Tab



Type in code list below:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xsi:schemaLocation="http://www.springframework.org/schema/beans
5                           http://www.springframework.org/schema/beans/spring-beans.xsd">
6
7    <bean id="helloBean" class="com.xyz.HelloWorld">
8      <property name="name" value="John Smith" />
9    </bean>
10
11  </beans>
```

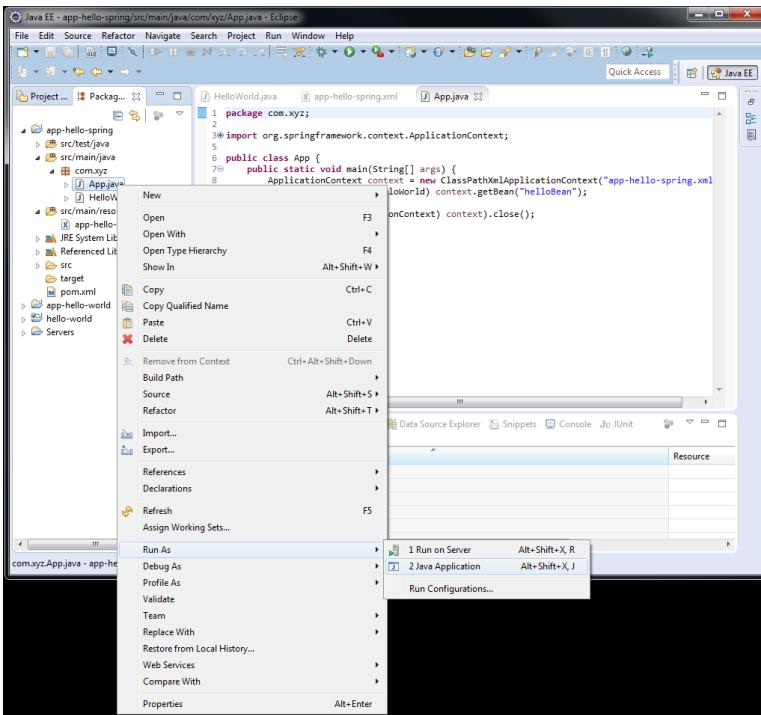
2.3.7 Instantiating a container

Instantiating a Spring IoC container is straightforward. The location path or paths supplied to an ApplicationContext constructor are actually resource strings that allow the container to load configuration metadata from a variety of external resources such as the local file system, from the Java CLASSPATH, and so on.

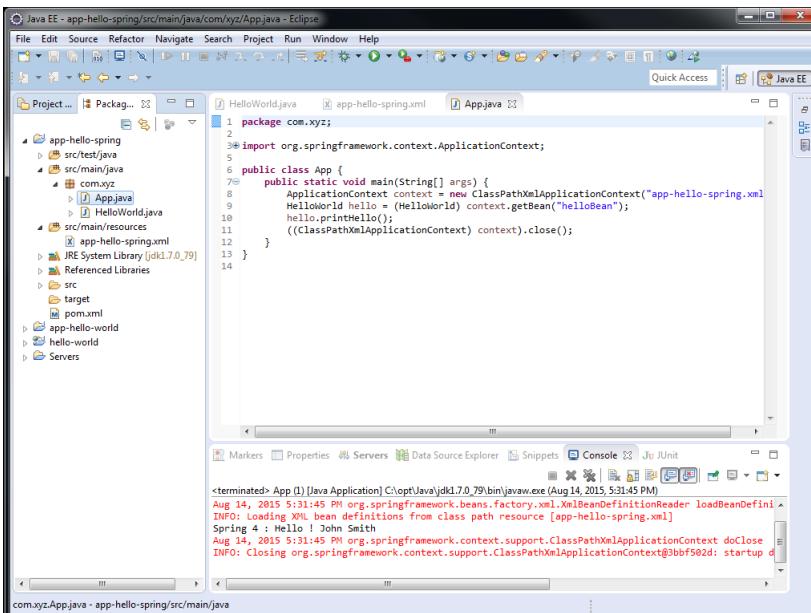
Type in the code list below to src/main/java ->com.xyz ->App.java

```
1 package com.xyz;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5
6 public class App {
7     public static void main(String[] args) {
8         ApplicationContext context = new
9             ClassPathXmlApplicationContext("app-hello-spring.xml");
10        HelloWorld hello = (HelloWorld) context.getBean("helloBean");
11        hello.printHello();
12        ((ClassPathXmlApplicationContext) context).close();
13    }
14 }
```

2.3.8 Run App



Right click on project src/main/java → com.xyz → App
Run As → Java Application



You will see in the console:
Spring 4 : Hello ! John Smith

2.4 Simple spring maven project - app-hello-spring-annotation

2.4.1 Create project using command line

```
1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=app-hello-spring-annotation
  ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

2.4.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\app-hello-spring-annotation\pom.xml

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-spring-annotation</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-spring-annotation</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <java.version>1.7</java.version>
12    <junit.version>4.12</junit.version>
13    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
14    <spring.version>4.2.0.RELEASE</spring.version>
15  </properties>
16  <dependencies>
17    <dependency>
18      <groupId>org.springframework</groupId>
19      <artifactId>spring-context</artifactId>
20      <version>${spring.version}</version>
21    </dependency>
22    <dependency>
23      <groupId>org.springframework</groupId>
24      <artifactId>spring-core</artifactId>
25      <version>${spring.version}</version>
26    </dependency>
27    <dependency>
28      <groupId>junit</groupId>
29      <artifactId>junit</artifactId>
30      <version>${junit.version}</version>
31      <scope>test</scope>
32    </dependency>
33  </dependencies>
34  <build>
35    <finalName>app-hello-spring-annotation</finalName>
36    <plugins>
37      <plugin>
38        <groupId>org.apache.maven.plugins</groupId>
39        <artifactId>maven-compiler-plugin</artifactId>
40        <version>${maven.compiler.plugin.version}</version>
41        <configuration>
42          <source>${java.version}</source>
43          <target>${java.version}</target>
44        </configuration>
45      </plugin>
46    </plugins>
47  </build>
48 </project>
```

2.4.3 Make this project an eclipse project

```
1 c:\workspaces\eclipse\my-projects\app-hello-spring-annotation>mvn eclipse:clean
→ eclipse:eclipse
```

2.4.4 Import this project to eclipse

File → Import → General → Existing Projects into Workspace
Browse to app-hello-spring-annotation, and import it.

2.4.5 Add a simple java bean

- src/main/java → com.xyz → HelloWorld.java

```
1 package com.xyz;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 public class HelloWorld {
6     @Autowired
7     private String name;
8
9     public void setName(String name) {
10         this.name = name;
11     }
12
13     public void printHello() {
14         System.out.println("Spring 4 Annotation : Hello ! " + name);
15     }
16 }
```

2.4.6 Configure spring bean

- Create resources directory and add it to Build Path
src/main/resources
- Configuration metadata
src/main/resources/app-hello-world-annotation.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xmlns:context="http://www.springframework.org/schema/context"
5         xsi:schemaLocation="http://www.springframework.org/schema/beans
6             http://www.springframework.org/schema/beans/spring-beans.xsd
7             http://www.springframework.org/schema/context
8             http://www.springframework.org/schema/context/spring-context.xsd">
9
10    <context:annotation-config />
11
12    <bean id="helloName" class="java.lang.String">
13        <constructor-arg value="John Smith" />
14    </bean>
15
16    <bean id="helloBean" class="com.xyz.HelloWorld">
17    </bean>
18
19 </beans>
```

2.4.7 Instantiating a container

- Type in the code list below to src/main/java → com.xyz → App.java

```

1 package com.xyz;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5
6 public class App {
7     public static void main(String[] args) {
8         ApplicationContext context = new
9             ClassPathXmlApplicationContext("app-hello-spring-annotation.xml");
10        HelloWorld hello = (HelloWorld) context.getBean("helloBean");
11        hello.printHello();
12    }
13 }
```

2.4.8 Run App

Right click on project src/main/java ->com.xyz ->App

Run As ->Java Application

You will see in the console:

Spring 4 Annotation : Hello ! John Smith

2.4.9 Using JSR 330 Standard Annotations

Starting with Spring 3.0, Spring offers support for JSR-330 standard annotations (Dependency Injection). Those annotations are scanned in the same way as the Spring annotations. You just need to have the relevant jars in your classpath.

If you are using Maven, the javax.inject artifact is available in the standard Maven repository (<http://repo1.maven.org/maven2/javax/inject/javax.inject/1/>). You can add the following dependency to your file pom.xml:

```

1 <dependency>
2   <groupId>javax.inject</groupId>
3   <artifactId>javax.inject</artifactId>
4   <version>1</version>
5 </dependency>
```

Instead of @Autowired, @javax.inject.Inject may be used

Instead of @Component, @javax.inject.Named may be used

2.5 Simple spring maven project - app-hello-spring-appconfig

2.5.1 Create project using command line

```

1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↵ -DartifactId=app-hello-spring-appconfig
  ↵ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

2.5.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\app-hello-spring-appconfig\pom.xml
-

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   ↵   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4     ↵   http://maven.apache.org/maven-v4_0_0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <groupId>com.xyz</groupId>
7   <artifactId>app-hello-spring-appconfig</artifactId>
8   <packaging>jar</packaging>
9   <version>1.0-SNAPSHOT</version>
10  <name>app-hello-spring-appconfig</name>
11  <url>http://maven.apache.org</url>
12  <properties>
13    <java.version>1.7</java.version>
14    <javax.inject.version>1</javax.inject.version>
15    <junit.version>4.12</junit.version>
16    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
17    <spring.version>4.2.0.RELEASE</spring.version>
18  </properties>
19  <dependencies>
20    <dependency>
21      <groupId>javax.inject</groupId>
22      <artifactId>javax.inject</artifactId>
23      <version>${javax.inject.version}</version>
24    </dependency>
25    <dependency>
26      <groupId>org.springframework</groupId>
27      <artifactId>spring-context</artifactId>
28      <version>${spring.version}</version>
29    </dependency>
30    <dependency>
31      <groupId>org.springframework</groupId>
32      <artifactId>spring-core</artifactId>
33      <version>${spring.version}</version>
34    </dependency>
35    <dependency>
36      <groupId>junit</groupId>
37      <artifactId>junit</artifactId>
38      <version>${junit.version}</version>
39      <scope>test</scope>
40    </dependency>
41  </dependencies>
42  <build>
43    <finalName>app-hello-spring-appconfig</finalName>
44    <plugins>
45      <plugin>
46        <groupId>org.apache.maven.plugins</groupId>
47        <artifactId>maven-compiler-plugin</artifactId>
48        <version>${maven.compiler.plugin.version}</version>
49        <configuration>
50          <source>${java.version}</source>
51          <target>${java.version}</target>
52        </configuration>
53      </plugin>
54    </plugins>
55  </build>
56</project>

```

2.5.3 Make this project an eclipse project

```

1 c:\workspaces\eclipse\my-projects\app-hello-spring-appconfig>mvn eclipse:clean
  ↵   eclipse:eclipse

```

2.5.4 Import this project to eclipse

File ->Import ->General ->Existing Projects into Workspace
 Browse to app-hello-spring-appconfig, and import it.

2.5.5 Add a simple java bean

- src/main/java -> com.xyz -> HelloWorld.java

```

1 package com.xyz;
2
3 import javax.inject.Inject;
4
5 public class HelloWorld {
6     @Inject
7     private String name;
8
9     public void setName(String name) {
10         this.name = name;
11     }
12
13     public void printHello() {
14         System.out.println("Spring 4 AppConfig : Hello ! " + name);
15     }
16 }
```

2.5.6 Configure spring bean - NO xml anymore

- src/main/java -> com.xyz -> AppConfig.java

```

1 package com.xyz;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.Configuration;
5
6 @Configuration
7 public class AppConfig {
8     @Bean
9     public String helloName() {
10         return new String("John Smith");
11     }
12
13     @Bean
14     public HelloWorld helloBean() {
15         return new HelloWorld();
16     }
17 }
```

2.5.7 Instantiating a container

- Type in the code list below to src/main/java -> com.xyz -> App.java

```

1 package com.xyz;
2
3 import org.springframework.context.ApplicationContext;
4 import
4   ↪ org.springframework.context.annotation.AnnotationConfigApplicationContext;
5
6 public class App {
7     public static void main(String[] args) {
8         ApplicationContext ctx = new
8           ↪ AnnotationConfigApplicationContext(AppConfig.class);
9         HelloWorld hello = (HelloWorld) ctx.getBean("helloBean");
10        hello.printHello();
11        ((AnnotationConfigApplicationContext) ctx).close();
12    }
13 }
```

2.5.8 Run App

Right click on project src/main/java ->com.xyz ->App

Run As ->Java Application

You will see in the console:

Spring 4 AppConfig : Hello ! John Smith

CHAPTER 3

SPRING MVC

3.1 Spring MVC - hello-spring

3.1.1 Create project using command line

```
1 c:\workspaces\workspace\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=hello-spring -DarchetypeArtifactId=maven-archetype-webapp
  ↳ -DinteractiveMode=false
```

3.1.2 Modify pom.xml

- C:\workspaces\workspace\my-projects\hello-spring\pom.xml

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>hello-spring</artifactId>
6   <packaging>war</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>hello-spring Maven Webapp</name>
9   <url>http://maven.apache.org</url>
```

[†]change c:\workspaces\workspace to /users/xyz/documents/workspace, then all the projects in this chapter will work in mac osx.

```

10  <properties>
11    <java.version>1.7</java.version>
12    <junit.version>4.12</junit.version>
13    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
14    <spring.version>4.2.0.RELEASE</spring.version>
15  </properties>
16  <dependencies>
17    <dependency>
18      <groupId>org.springframework</groupId>
19      <artifactId>spring-core</artifactId>
20      <version>${spring.version}</version>
21    </dependency>
22    <dependency>
23      <groupId>org.springframework</groupId>
24      <artifactId>spring-web</artifactId>
25      <version>${spring.version}</version>
26    </dependency>
27    <dependency>
28      <groupId>org.springframework</groupId>
29      <artifactId>spring-webmvc</artifactId>
30      <version>${spring.version}</version>
31    </dependency>
32    <dependency>
33      <groupId>junit</groupId>
34      <artifactId>junit</artifactId>
35      <version>${junit.version}</version>
36      <scope>test</scope>
37    </dependency>
38  </dependencies>
39  <build>
40    <finalName>hello-spring</finalName>
41    <plugins>
42      <plugin>
43        <groupId>org.apache.maven.plugins</groupId>
44        <artifactId>maven-compiler-plugin</artifactId>
45        <version>${maven.compiler.plugin.version}</version>
46        <configuration>
47          <source>${java.version}</source>
48          <target>${java.version}</target>
49        </configuration>
50      </plugin>
51    </plugins>
52  </build>
53</project>
```

3.1.3 Make this project an eclipse project

```

1 c:\workspaces\eclipse\my-projects\hello-spring>mvn eclipse:clean eclipse:eclipse
   ↵ -Dwtpversion=2.0

```

3.1.4 Import this project to eclipse

File → Import → General → Existing Projects into Workspace
 Browse to hello-spring, and import it.

3.1.5 Controller

- Create java directory and add it to Build Path
 src/main/java
- Create a new a java controller class
 src/main/java/com/xyz/controller/HelloController.java

```

1 package com.xyz.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.ModelMap;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestMethod;
7
8 @Controller
9 @RequestMapping("/welcome")
10 public class HelloController {
11
12     @RequestMapping(method = RequestMethod.GET)
13     public String printWelcome(ModelMap model) {
14         model.addAttribute("message", "Spring 4 MVC: Hello World");
15         return "hello";
16     }
17
18 }
```

3.1.6 JSP Views

- Create jsp directory
src/main/webapp/WEB-INF/jsp
- Create a new hello JSP file
src/main/webapp/WEB-INF/jsp/hello.jsp

```
<h1>Message : ${message}</h1>
```

3.1.7 Spring servlet configuration

- Create a new Spring servlet configuration file

Upon initialization of a DispatcherServlet, Spring MVC looks for a file named [servlet-name]-servlet.xml in the WEB-INF directory of your web application and creates the beans defined there, overriding the definitions of any beans defined with the same name in the global scope.

src/main/webapp/WEB-INF/hello-spring-servlet.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="
5       http://www.springframework.org/schema/beans
6       http://www.springframework.org/schema/beans/spring-beans.xsd
7       http://www.springframework.org/schema/context
8       http://www.springframework.org/schema/context/spring-context.xsd">
9
10    <context:component-scan base-package="com.xyz.controller" />
11
12    <bean id="viewResolver"
13      class="org.springframework.web.servlet.view.InternalResourceViewResolver">
14        <property name="prefix">
15          <value>/WEB-INF/jsp/</value>
16        </property>
17        <property name="suffix">
18          <value>.jsp</value>
19        </property>
20    </bean>
21
22 </beans>
```

3.1.8 Integrate web application with Spring - web.xml

- src/main/webapp/WEB-INF/web.xml

```

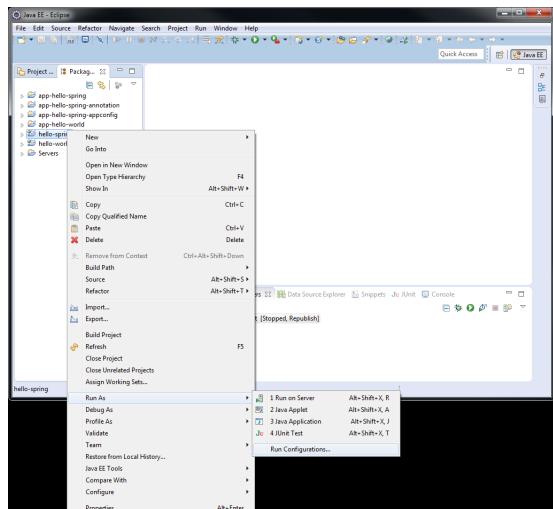
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" version="2.4">
5   <display-name>Hello Spring</display-name>
6
7   <servlet>
8     <servlet-name>hello-spring</servlet-name>
9
10    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
11    <load-on-startup>1</load-on-startup>
12  </servlet>
13
14  <servlet-mapping>
15    <servlet-name>hello-spring</servlet-name>
16    <url-pattern>/</url-pattern>
17  </servlet-mapping>
18
19 </web-app>

```

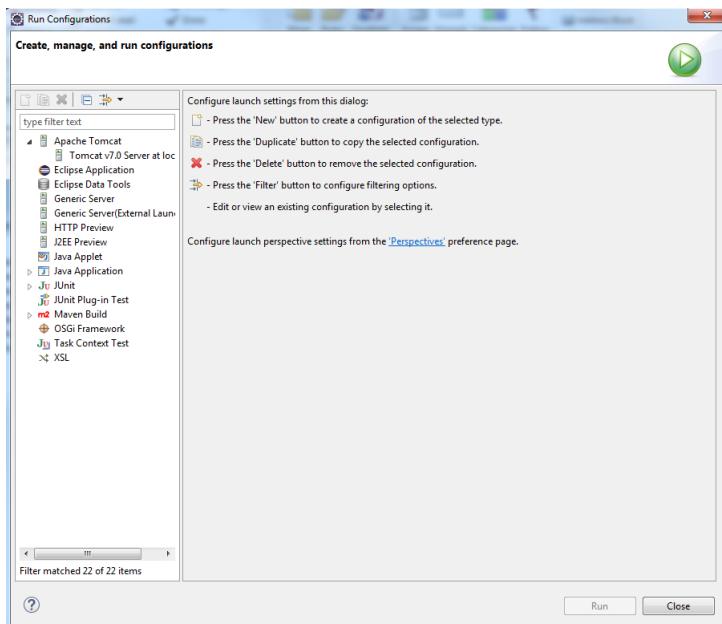
3.1.9 Build the project - command line

```
c:\workspaces\eclipse\my-projects\hello-spring>mvn install
```

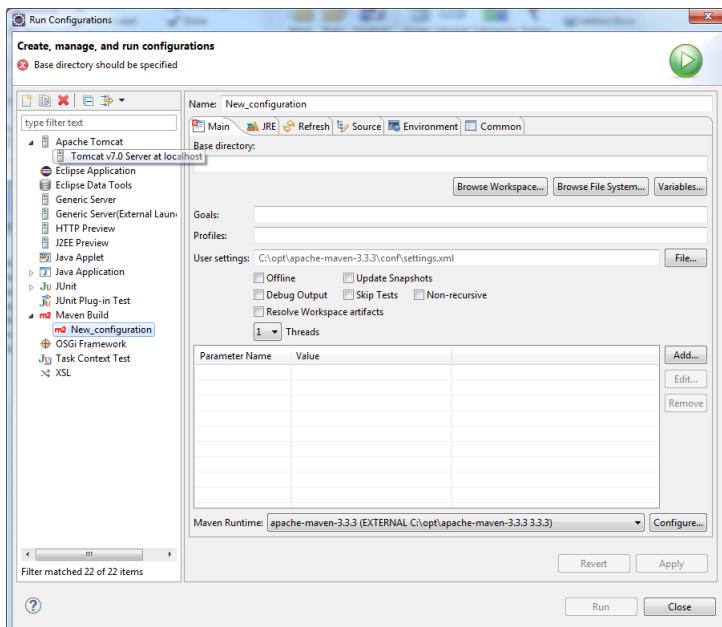
3.1.10 Build the project - eclipse



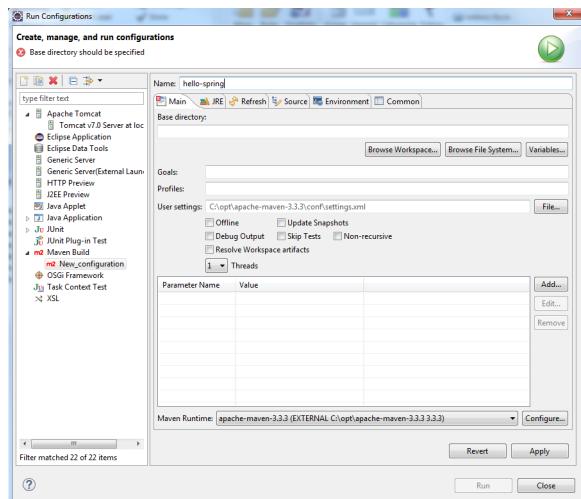
Right click on project ->Run As ->Run Configuration...



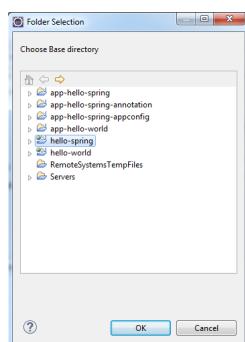
Select Maven Build, click the New icon on the top-left corner



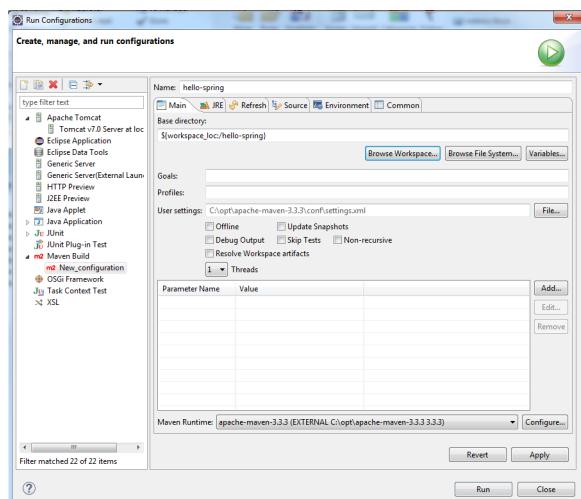
Type in Name: hello-spring



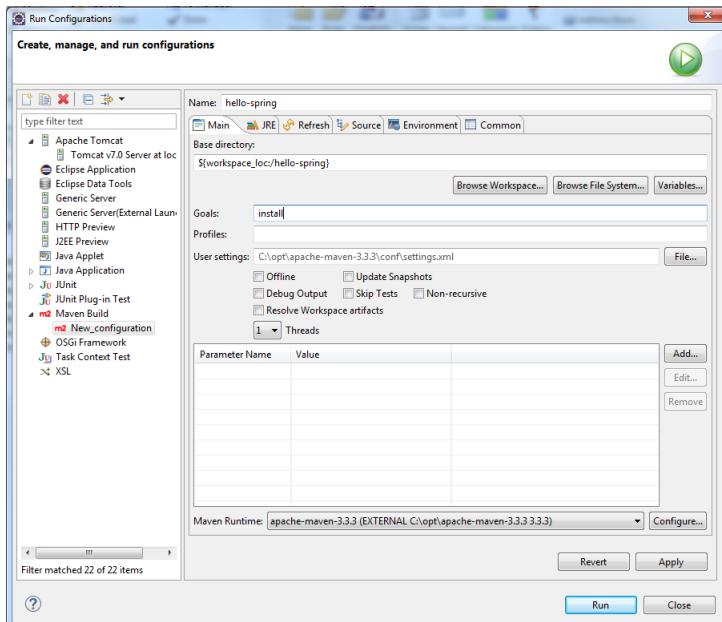
Click Browse Workspace...



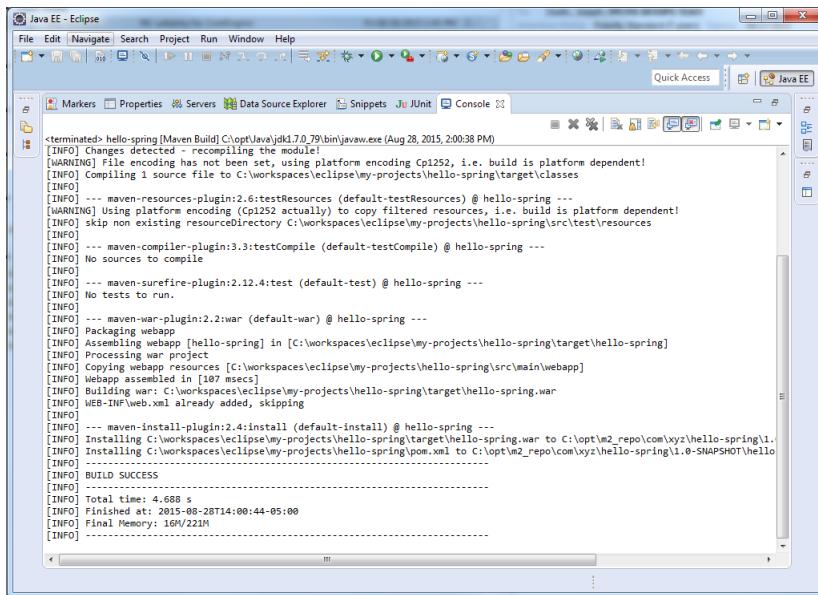
Select hello-spring



Type in Goals: install



Click Apply, Run, then maximize console window



3.1.11 Run

- Add project to Tomcat Server

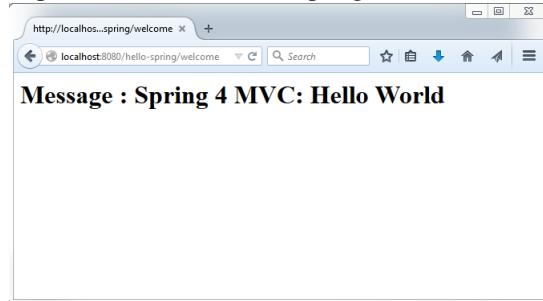
Go to Servers Tab

Right click Tomcat v7.0 Server at localhost [Stopped] → Add and Remove...

Move hello-spring from Available panel (left side) to Configured panel (right side)

- Start Tomcat Server

<http://localhost:8080/hello-spring/welcome>



3.2 Separate Spring configuration xml files

3.2.1 Create project using command line

```
1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=hello-spring-mvc -DarchetypeArtifactId=maven-archetype-webapp
  ↳ -DinteractiveMode=false
```

3.2.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\hello-spring-mvc\pom.xml

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>hello-spring-mvc</artifactId>
6   <packaging>war</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>hello-spring-mvc Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <java.version>1.7</java.version>
12    <junit.version>4.12</junit.version>
13    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
14    <spring.version>4.2.0.RELEASE</spring.version>
15  </properties>
16  <dependencies>
17    <dependency>
18      <groupId>org.springframework</groupId>
19      <artifactId>spring-core</artifactId>
20      <version>${spring.version}</version>
21    </dependency>
22    <dependency>
23      <groupId>org.springframework</groupId>
24      <artifactId>spring-web</artifactId>
25      <version>${spring.version}</version>
26    </dependency>
27    <dependency>
28      <groupId>org.springframework</groupId>
29      <artifactId>spring-webmvc</artifactId>
30      <version>${spring.version}</version>
31    </dependency>
32    <dependency>
33      <groupId>junit</groupId>
```

```

34      <artifactId>junit</artifactId>
35      <version>${junit.version}</version>
36      <scope>test</scope>
37  </dependency>
38 </dependencies>
39 <build>
40   <finalName>hello-spring-mvc</finalName>
41   <plugins>
42     <plugin>
43       <groupId>org.apache.maven.plugins</groupId>
44       <artifactId>maven-compiler-plugin</artifactId>
45       <version>${maven.compiler.plugin.version}</version>
46       <configuration>
47         <source>${java.version}</source>
48         <target>${java.version}</target>
49       </configuration>
50     </plugin>
51   </plugins>
52 </build>
53 </project>

```

3.2.3 Make this project an eclipse project

```

1 c:\workspaces\eclipse\my-projects\hello-spring-mvc>mvn eclipse:clean
→ eclipse:eclipse -Dwtpversion=2.0

```

3.2.4 Import this project to eclipse

File ->Import ->General ->Existing Projects into Workspace
Browse to hello-spring-mvc, and import it.

3.2.5 Controller

- Create java directory and add it to Build Path
src/main/java
- Create a new a java controller class
src/main/java/com/xyz/controller/HelloController.java

```

1 package com.xyz.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.ModelMap;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestMethod;
7
8 @Controller
9 @RequestMapping("/welcome")
10 public class HelloController {
11
12     @RequestMapping(method = RequestMethod.GET)
13     public String printWelcome(ModelMap model) {
14         model.addAttribute("message", "Spring 4 MVC: Hello World");
15         return "hello";
16     }
17
18 }

```

3.2.6 JSP Views

- Create jsp directory
src/main/webapp/WEB-INF/jsp
- Create a new hello JSP file
src/main/webapp/WEB-INF/jsp/hello.jsp

```
<h1>Message : ${message}</h1>
```

3.2.7 Spring resources configuration

- Create a new directory spring under src/main/resources, create a new Spring resources configuration file in this directory
src/main/resources/spring/hello-spring-mvc-context.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/context
5       http://www.springframework.org/schema/context/spring-context.xsd
6       http://www.springframework.org/schema/mvc
7         http://www.springframework.org/schema/mvc/spring-mvc.xsd">
8
9   <mvc:annotation-driven />
10  <context:component-scan base-package="com.xyz.controller" />
11
12 </beans>
```

3.2.8 Spring servlet configuration

- Create a new Spring servlet configuration file
src/main/webapp/WEB-INF/hello-spring-mvc-servlet.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/context
5       http://www.springframework.org/schema/context/spring-context.xsd
6       http://www.springframework.org/schema/beans
7         http://www.springframework.org/schema/beans/spring-beans.xsd
8       http://www.springframework.org/schema/context
9         http://www.springframework.org/schema/context/spring-context.xsd">
10
11  <bean id="viewResolver"
12    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
13    <property name="prefix">
14      <value>/WEB-INF/jsp/</value>
15    </property>
16    <property name="suffix">
17      <value>.jsp</value>
18    </property>
19  </bean>
20
21 </beans>
```

3.2.9 Integrate web application with Spring - web.xml

- It is wise sometimes to split all the Configuration information across multiple Configuration Files.

In such a case we have to depend on a Listener Servlet called Context Loader represented by org.springframework.web.context.ContextLoaderListener.

By default, this Context Listener will try to look for the Configuration File by name 'applicationContext.xml' in the '/WEB-INF' directory.

But with the help of the parameter 'contextConfigLocation' the default location can be overridden. Even multiple Configuration Files each containing separate piece of Information is also possible.

For example:

```
<param-value>/WEB-INF/contacts.xml,/WEB-INF/resources.xml</param-value>
```

- src/main/webapp/WEB-INF/web.xml

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
   ↪ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
   ↪ http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
4   version="2.4">
5
6   <display-name>Hello Spring MVC</display-name>
7
8   <context-param>
9     <param-name>contextConfigLocation</param-name>
10    <param-value>classpath:spring/hello-spring-mvc-context.xml</param-value>
11  </context-param>
12
13  <listener>
14    <!--> <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
15 </listener>
16
17  <servlet>
18    <servlet-name>hello-spring-mvc</servlet-name>
19
20      <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
21      <load-on-startup>1</load-on-startup>
22  </servlet>
23
24  <servlet-mapping>
25    <servlet-name>hello-spring-mvc</servlet-name>
26    <url-pattern>/</url-pattern>
27  </servlet-mapping>
28
29 </web-app>

```

3.2.10 Build the project

```
c:\workspaces\workspace\my-projects\hello-spring-mvc>mvn install
```

3.2.11 Run

- Add project to Tomcat Server

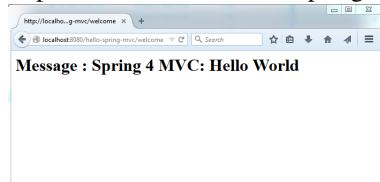
Go to Servers Tab

Right click Tomcat v7.0 Server at localhost [Stopped] → Add and Remove...

Move hello-spring-mvc from Available panel (left side) to Configured panel (right)

side)

- Start Tomcat Server
<http://localhost:8080/hello-spring-mvc/welcome>



3.2.12 Add a controller which returns null view name to project

- If a controller returns a null view name, or declares a void return type, Spring will attempt to infer the view name from the request URL
- Create a new java controller class
`src/main/java/com/xyz/controller/HelloVoidController.java`

```

1 package com.xyz.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.ModelMap;
5 import org.springframework.web.bind.annotation.RequestMapping;
6
7 @Controller
8 public class HelloVoidController {
9
10     @RequestMapping("/hello")
11     public void printWelcome(ModelMap model) {
12         model.addAttribute("message", "Spring 4 MVC: Hello World - Void");
13     }
14
15 }
```

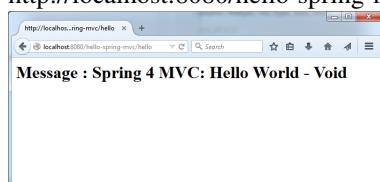
- Spring will try to find hello.jsp

3.2.13 Build the project

```
c:\workspaces\eclipse\my-projects\hello-spring-mvc>mvn install
```

3.2.14 Run

- Start Tomcat Server
<http://localhost:8080/hello-spring-mvc/hello>



CHAPTER 4

MORE SPRING

4.1 Spring JSON - hello-spring-json

4.1.1 Create project using command line

```
1 c:\workspaces\workspace>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=hello-spring-json -DarchetypeArtifactId=maven-archetype-webapp
  ↳ -DinteractiveMode=false
```

4.1.2 Modify pom.xml

- C:\workspaces\workspace\my-projects\hello-spring-json\pom.xml

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>hello-spring-json</artifactId>
6   <packaging>war</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>hello-spring-json Maven Webapp</name>
9   <url>http://maven.apache.org</url>
```

[†]change c:\workspaces\workspace to /users/xyz/documents/workspace, then all the projects in this chapter will work in mac osx.

```

10 <properties>
11   <jackson-jaxrs-base.version>2.6.1</jackson-jaxrs-base.version>
12   <java.version>1.7</java.version>
13   <junit.version>4.12</junit.version>
14   <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
15   <spring.version>4.2.0.RELEASE</spring.version>
16 </properties>
17 <dependencies>
18   <dependency>
19     <groupId>org.springframework</groupId>
20     <artifactId>spring-core</artifactId>
21     <version>${spring.version}</version>
22   </dependency>
23   <dependency>
24     <groupId>org.springframework</groupId>
25     <artifactId>spring-web</artifactId>
26     <version>${spring.version}</version>
27   </dependency>
28   <dependency>
29     <groupId>org.springframework</groupId>
30     <artifactId>spring-webmvc</artifactId>
31     <version>${spring.version}</version>
32   </dependency>
33   <dependency>
34     <groupId>com.fasterxml.jackson.jaxrs</groupId>
35     <artifactId>jackson-jaxrs-base</artifactId>
36     <version>2.6.1</version>
37   </dependency>
38   <dependency>
39     <groupId>junit</groupId>
40     <artifactId>junit</artifactId>
41     <version>${junit.version}</version>
42     <scope>test</scope>
43   </dependency>
44 </dependencies>
45 <build>
46   <finalName>hello-spring-json</finalName>
47   <plugins>
48     <plugin>
49       <groupId>org.apache.maven.plugins</groupId>
50       <artifactId>maven-compiler-plugin</artifactId>
51       <version>${maven.compiler.plugin.version}</version>
52       <configuration>
53         <source>${java.version}</source>
54         <target>${java.version}</target>
55       </configuration>
56     </plugin>
57   </plugins>
58 </build>
59 </project>
```

4.1.3 Make this project an eclipse project

```
1 c:\workspaces\workspace\my-projects\hello-spring-json>mvn eclipse:clean
  ↵  eclipse:eclipse -Dwtpversion=2.0
```

4.1.4 Import this project to eclipse

File → Import → General → Existing Projects into Workspace
 Browse to hello-spring-json, and import it.

4.1.5 Model

- Create java directory and add it to Build Path
src/main/java
- Create a new a java model class
src/main/java/com/xyz/model/Author.java

```

1 package com.xyz.model;
2
3 public class Author {
4     String name;
5     String books[];
6
7     public String getName() {
8         return name;
9     }
10
11    public void setName(String name) {
12        this.name = name;
13    }
14
15    public String[] getBooks() {
16        return books;
17    }
18
19    public void setBooks(String[] books) {
20        this.books = books;
21    }
22
23 }
```

4.1.6 Controller

- Create a new a java controller class
src/main/java/com/xyz/controller/JsonController.java

```

1 package com.xyz.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.PathVariable;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestMethod;
7 import org.springframework.web.bind.annotation.ResponseBody;
8
9 import com.xyz.model.Author;
10
11 @Controller
12 @RequestMapping("/xyz/authors")
13 public class JsonController {
14
15     @RequestMapping(value = "{name}", method = RequestMethod.GET)
16     public @ResponseBody Author getShopInJSON(@PathVariable String name) {
17         Author author = new Author();
18         author.setName(name);
19         author.setBooks(new String[] { "Hamlet", "Romeo and Juliet" });
20         return author;
21     }
22
23 }
```

4.1.7 Spring servlet configuration

- Create a new Spring servlet configuration file

src/main/webapp/WEB-INF/hello-spring-json-servlet.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     <!-- xmlns:context="http://www.springframework.org/schema/context"
5       xmlns:mvc="http://www.springframework.org/schema/mvc"
6
7       xsi:schemaLocation="
8         http://www.springframework.org/schema/beans
9           http://www.springframework.org/schema/beans/spring-beans.xsd
10        http://www.springframework.org/schema/context
11          http://www.springframework.org/schema/context/spring-context.xsd
12        http://www.springframework.org/schema/mvc
13          http://www.springframework.org/schema/mvc/spring-mvc.xsd">
14
15   <mvc:annotation-driven>
16     <mvc:message-converters>
17       <bean
18         class="org.springframework.http.converter.StringHttpMessageConverter" />
19       <bean
20         class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter"
21         />
22     </mvc:message-converters>
23   </mvc:annotation-driven>
24
25   <context:component-scan base-package="com.xyz.controller" />
26
27 </beans>
```

4.1.8 Integrate web application with Spring - web.xml

- src/main/webapp/WEB-INF/web.xml

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
3   <!-- xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
5       http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
6   version="2.4">
7
8   <display-name>Hello Spring JSON</display-name>
9
10  <servlet>
11    <servlet-name>hello-spring-json</servlet-name>
12
13    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
14    <load-on-startup>1</load-on-startup>
15  </servlet>
16
17  <servlet-mapping>
18    <servlet-name>hello-spring-json</servlet-name>
19    <url-pattern>/rest/*</url-pattern>
20  </servlet-mapping>
21
22 </web-app>
```

4.1.9 Build the project

```
c:\workspaces\workspace\my-projects\hello-spring-json>mvn install
```

4.1.10 Run

- Add project to Tomcat Server

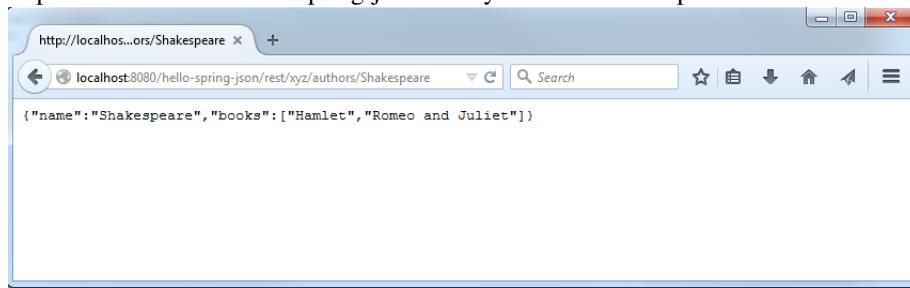
Go to Servers Tab

Right click Tomcat v7.0 Server at localhost [Stopped] → Add and Remove...

Move hello-spring-json from Available panel (left side) to Configured panel (right side)

- Start Tomcat Server

<http://localhost:8080/hello-spring-json/rest/xyz/authors/Shakespeare>



4.2 Spring Security - hello-spring-security

4.2.1 Create project using command line

```
1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=hello-spring-security
  ↳ -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false
```

4.2.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\hello-spring-security\pom.xml

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>hello-spring-security</artifactId>
6   <packaging>war</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>hello-spring-security Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <java.version>1.7</java.version>
12    <junit.version>4.12</junit.version>
13    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
14    <spring.version>4.2.0.RELEASE</spring.version>
15    <spring.security.version>4.0.2.RELEASE</spring.security.version>
16  </properties>
17  <dependencies>
18    <dependency>
19      <groupId>org.springframework</groupId>
20      <artifactId>spring-core</artifactId>
21      <version>${spring.version}</version>
22    </dependency>
23    <dependency>
```

```

24      <groupId>org.springframework</groupId>
25      <artifactId>spring-web</artifactId>
26      <version>${spring.version}</version>
27  </dependency>
28  <dependency>
29      <groupId>org.springframework</groupId>
30      <artifactId>spring-webmvc</artifactId>
31      <version>${spring.version}</version>
32  </dependency>
33  <dependency>
34      <groupId>org.springframework.security</groupId>
35      <artifactId>spring-security-config</artifactId>
36      <version>${spring.security.version}</version>
37  </dependency>
38  <dependency>
39      <groupId>org.springframework.security</groupId>
40      <artifactId>spring-security-core</artifactId>
41      <version>${spring.security.version}</version>
42  </dependency>
43  <dependency>
44      <groupId>org.springframework.security</groupId>
45      <artifactId>spring-security-web</artifactId>
46      <version>${spring.security.version}</version>
47  </dependency>
48  <dependency>
49      <groupId>junit</groupId>
50      <artifactId>junit</artifactId>
51      <version>${junit.version}</version>
52      <scope>test</scope>
53  </dependency>
54 </dependencies>
55 <build>
56     <finalName>hello-spring-security</finalName>
57     <plugins>
58         <plugin>
59             <groupId>org.apache.maven.plugins</groupId>
60             <artifactId>maven-compiler-plugin</artifactId>
61             <version>${maven.compiler.plugin.version}</version>
62             <configuration>
63                 <source>${java.version}</source>
64                 <target>${java.version}</target>
65             </configuration>
66         </plugin>
67     </plugins>
68 </build>
69 </project>

```

4.2.3 Make this project an eclipse project

```

1 c:\workspaces\workspace\my-projects\hello-spring-security>mvn eclipse:clean
  ↳ eclipse:eclipse -Dwtpversion=2.0

```

4.2.4 Import this project to eclipse

File ->Import ->General ->Existing Projects into Workspace
 Browse to hello-spring-security, and import it.

4.2.5 Controller

- Create java directory and add it to Build Path
 src/main/java

- Create a new a java model class
src/main/java/com/xyz/controller/HelloController.java

```

1 package com.xyz.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.ModelMap;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestMethod;
7
8 @Controller
9 @RequestMapping("/welcome")
10 public class HelloController {
11
12     @RequestMapping(method = RequestMethod.GET)
13     public String printWelcome(ModelMap model) {
14         model.addAttribute("message", "Spring Security Hello World");
15         return "hello";
16     }
17 }
18 }
```

4.2.6 JSP Views

- Create jsp directory
src/main/webapp/WEB-INF/jsp
- Create a new hello JSP file
src/main/webapp/WEB-INF/jsp/hello.jsp

```
<h1>Message : ${message}</h1>
```

4.2.7 Spring Security Namespace Configuration

- To start using the security namespace in your application context, you need to have the spring-security-config jar on your classpath. Then all you need to do is add the schema declaration to your application context file:

```

1 <beans xmlns="http://www.springframework.org/schema/beans"
2   xmlns:security="http://www.springframework.org/schema/security"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://www.springframework.org/schema/beans
5       http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
6       http://www.springframework.org/schema/security
7       http://www.springframework.org/schema/security/spring-security.xsd">
8 ...
9 </beans>
```

- In many of the examples you will see (and in the sample) applications, we will often use "security" as the default namespace rather than "beans", which means we can omit the prefix on all the security namespace elements, making the content easier to read. You may also want to do this if you have your application context divided up into separate files and have most of your security configuration in one of them. Your security application context file would then start like this

```

1 <beans:beans xmlns="http://www.springframework.org/schema/security"
2   xmlns:beans="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://www.springframework.org/schema/beans
```

```

5      http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
6      http://www.springframework.org/schema/security
7      http://www.springframework.org/schema/security/spring-security.xsd">
8      ...
9  </beans:beans>
```

Well assume this syntax is being used from now on in this book.

- Create a new Spring configuration file
src/main/webapp/WEB-INF/hello-spring-security-context.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans:beans xmlns="http://www.springframework.org/schema/security"
3    xmlns:beans="http://www.springframework.org/schema/beans"
4    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5    xsi:schemaLocation="http://www.springframework.org/schema/beans
6      http://www.springframework.org/schema/beans/spring-beans.xsd
7      http://www.springframework.org/schema/security
8      http://www.springframework.org/schema/security/spring-security.xsd">
9
10 <http auto-config="true">
11   <intercept-url pattern="/welcome*" access="hasRole('USER')"/>
12 </http>
13 <authentication-manager>
14   <authentication-provider>
15     <user-service>
16       <user name="john" password="password" authorities="ROLE_USER"/>
17     </user-service>
18   </authentication-provider>
19 </authentication-manager>
20
21 </beans:beans>
```

4.2.8 Spring servlet configuration

- Create a new Spring servlet configuration file
src/main/webapp/WEB-INF/hello-spring-security-servlet.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4    xmlns:context="http://www.springframework.org/schema/context"
5    xsi:schemaLocation="http://www.springframework.org/schema/beans
6      http://www.springframework.org/schema/beans/spring-beans.xsd
7      http://www.springframework.org/schema/context
8      http://www.springframework.org/schema/context/spring-context.xsd">
9
10 <context:component-scan base-package="com.xyz.controller"/>
11
12 <bean id="viewResolver"
13   class="org.springframework.web.servlet.view.InternalResourceViewResolver">
14   <property name="prefix">
15     <value>/WEB-INF/jsp/</value>
16   </property>
17   <property name="suffix">
18     <value>.jsp</value>
19   </property>
20 </bean>
21 </beans>
```

4.2.9 Integrate web application with Spring - web.xml

- src/main/webapp/WEB-INF/web.xml

```

1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
5          http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
6      version="2.4">
7
8      <display-name>Hello Spring Security</display-name>
9
10     <context-param>
11         <param-name>contextConfigLocation</param-name>
12         <param-value>/WEB-INF/hello-spring-security-context.xml,
13             /WEB-INF/hello-spring-security-servlet.xml</param-value>
14     </context-param>
15
16     <listener>
17         <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
18     </listener>
19
20     <filter>
21         <filter-name>springSecurityFilterChain</filter-name>
22
23         <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
24     </filter>
25
26     <filter-mapping>
27         <filter-name>springSecurityFilterChain</filter-name>
28         <url-pattern>/*</url-pattern>
29     </filter-mapping>
30
31     <servlet>
32         <servlet-name>hello-spring-security</servlet-name>
33
34         <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
35         <load-on-startup>1</load-on-startup>
36     </servlet>
37
38 </web-app>

```

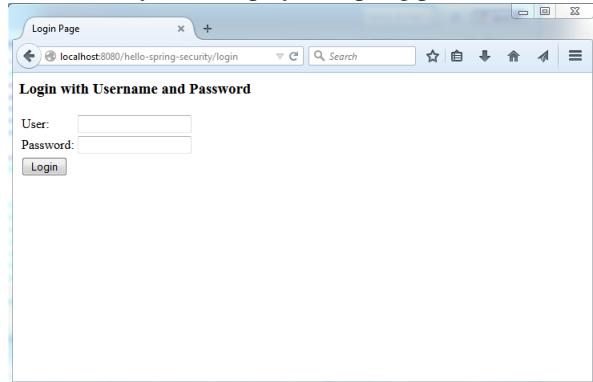
4.2.10 Build the project

```
c:\workspaces\eclipse\my-projects\hello-spring-security>mvn install
```

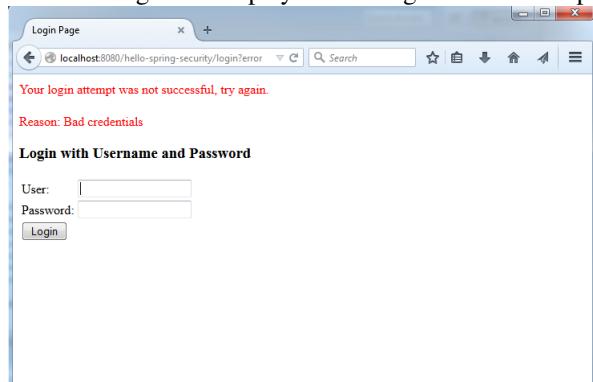
4.2.11 Run

- Add project to Tomcat Server
Go to Servers Tab
Right click Tomcat v7.0 Server at localhost [Stopped] ->Add and Remove...
Move hello-spring-security from Available panel (left side) to Configured panel (right side)
- Start Tomcat Server
<http://localhost:8080/hello-spring-security/welcome>
Access <http://localhost:8080/hello-spring-security/welcome>, Spring Security will in-

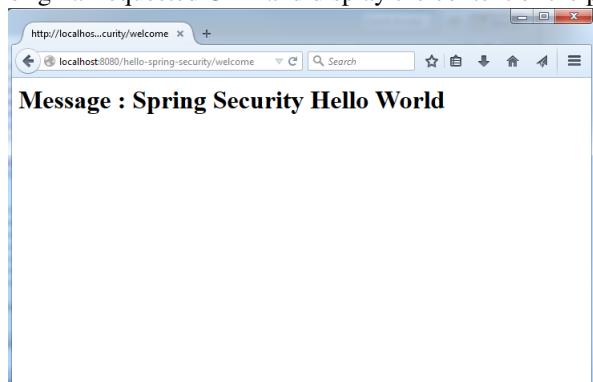
tercept the request and redirect to `http://localhost:8080/hello-spring-security/spring_security_login` automatically. And display the Spring predefined authentication form.



Error messages are displayed if wrong username and password are provided



If correct username and password are provided, Spring security will redirect to the original requested URL and display the content of the page.



4.3 Spring Batch - app-hello-spring-batch

4.3.1 Create project using command line

```

1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↵ -DartifactId=app-hello-spring-batch
  ↵ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

```

4.3.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\app-hello-spring-batch\pom.xml

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↵ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  ↵ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-spring-batch</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-spring-batch</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <java.version>1.7</java.version>
12    <junit.version>4.12</junit.version>
13    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
14    <spring.batch.version>3.0.5.RELEASE</spring.batch.version>
15  </properties>
16  <dependencies>
17    <dependency>
18      <groupId>org.springframework.batch</groupId>
19      <artifactId>spring-batch-core</artifactId>
20      <version>${spring.batch.version}</version>
21    </dependency>
22    <dependency>
23      <groupId>junit</groupId>
24      <artifactId>junit</artifactId>
25      <version>${junit.version}</version>
26      <scope>test</scope>
27    </dependency>
28  </dependencies>
29  <build>
30    <finalName>app-hello-spring-batch</finalName>
31    <plugins>
32      <plugin>
33        <groupId>org.apache.maven.plugins</groupId>
34        <artifactId>maven-compiler-plugin</artifactId>
35        <version>${maven.compiler.plugin.version}</version>
36        <configuration>
37          <source>${java.version}</source>
38          <target>${java.version}</target>
39        </configuration>
40      </plugin>
41    </plugins>
42  </build>
43</project>

```

4.3.3 Make this project an eclipse project

```

1 c:\workspaces\ eclipse\my-projects\app-hello-spring-batch>mvn eclipse:clean
  ↵ eclipse:eclipse

```

4.3.4 Import this project to eclipse

File ->Import ->General ->Existing Projects into Workspace
 Browse to app-hello-spring-batch, and import it.

4.3.5 Tasklet

- Create a new a java model class

src/main/java/com/xyz/batch/PrintTasklet.java

```

1 package com.xyz.batch;
2
3 import org.springframework.batch.core.StepContribution;
4 import org.springframework.batch.core.scope.context.ChunkContext;
5 import org.springframework.batch.core.step.tasklet.Tasklet;
6 import org.springframework.batch.repeat.RepeatStatus;
7
8 public class PrintTasklet implements Tasklet {
9     private String message;
10
11     public void setMessage(String message) {
12         this.message = message;
13     }
14
15     public RepeatStatus execute(StepContribution arg0, ChunkContext arg1) throws
16             Exception {
17         System.out.print(message);
18         return RepeatStatus.FINISHED;
19     }
20 }
```

4.3.6 Configure spring bean

- Create resources directory and add it to Build Path
src/main/resources

- Configuration metadata

src/main/resources/applicationContext.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/beans
5         http://www.springframework.org/schema/beans/spring-beans.xsd">
6
7     <bean id="jobRepository"
8         class="org.springframework.batch.core.repository.support.SimpleJobRepository">
9         <constructor-arg>
10            <bean
11                class="org.springframework.batch.core.repository.dao.MapJobInstanceDao"
12                />
13        </constructor-arg>
14        <constructor-arg>
15            <bean
16                class="org.springframework.batch.core.repository.dao.MapJobExecutionDao"
17                />
18        </constructor-arg>
19        <constructor-arg>
20            <bean
21                class="org.springframework.batch.core.repository.dao.MapStepExecutionDao"
22                />
23        </constructor-arg>
24    </bean>
```

```

23 <bean id="transactionManager"
24   class="org.springframework.batch.support.transaction.ResourcelessTransactionManager"
25   />
26 <bean id="jobLauncher"
27   class="org.springframework.batch.core.launch.support.SimpleJobLauncher">
28   <property name="jobRepository" ref="jobRepository" />
29 </bean>
30
31 </beans>
```

4.3.7 Configure job xml

- Configuration hello-world job xml
src/main/resources/helloWorldJob.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     <!-- xmlns:batch="http://www.springframework.org/schema/batch"
5       xsi:schemaLocation="http://www.springframework.org/schema/beans
6         http://www.springframework.org/schema/beans/spring-beans.xsd
7         http://www.springframework.org/schema/batch
8         http://www.springframework.org/schema/batch/spring-batch.xsd"-->
9
10 <import resource="applicationContext.xml" />
11 <bean id="hello" class="com.xyz.batch.PrintTasklet">
12   <property name="message" value="Hello" />
13 </bean>
14
15 <bean id="space" class="com.xyz.batch.PrintTasklet">
16   <property name="message" value=" " />
17 </bean>
18
19 <bean id="world" class="com.xyz.batch.PrintTasklet">
20   <property name="message" value=" World!" />
21 </bean>
22
23 <bean id="taskletStep" abstract="true"
24   class="org.springframework.batch.core.step.tasklet.TaskletStep">
25   <property name="jobRepository" ref="jobRepository" />
26   <property name="transactionManager" ref="transactionManager" />
27 </bean>
28
29 <bean id="helloWorldJob" class="org.springframework.batch.core.job.SimpleJob">
30   <property name="name" value="helloWorldJob" />
31   <property name="steps">
32     <list>
33       <bean parent="taskletStep">
34         <property name="tasklet" ref="hello" />
35       </bean>
36       <bean parent="taskletStep">
37         <property name="tasklet" ref="space" />
38       </bean>
39       <bean parent="taskletStep">
40         <property name="tasklet" ref="world" />
41       </bean>
42     </list>
43   </property>
44   <property name="jobRepository" ref="jobRepository" />
45 </bean>
46
47 </beans>
```

4.3.8 Build the project

```
c:\workspaces\workspace\my-projects\app-hello-spring-batch>mvn package
```

4.3.9 Run App in command line

```
1 c:\workspaces\workspace\my-projects\app-hello-spring-batch>mvn exec:java
  ↳ -Dexec.mainClass=org.springframework.batch.core.launch.support.CommandLineJobRunner
  ↳ -Dexec.args="helloWorldJob.xml helloWorldJob"
```

or

```
1 c:\workspaces\workspace\my-projects\app-hello-spring-batch>mvn exec:java
  ↳ -Dhttp.proxyHost=http.proxy.xyz.com -Dhttp.proxyPort=8000
  ↳ -Dexec.mainClass=org.springframework.batch.core.launch.support.CommandLineJobRunner
  ↳ -Dexec.args="helloWorldJob.xml helloWorldJob"
```

You will see your screen list this:

```
1 c:\workspaces\workspace\my-projects\app-hello-spring-batch>mvn exec:java
  ↳ -Dexec.mainClass=org.springframework.batch.core.launch.support.CommandLineJobRunner
  ↳ -Dexec.args="helloWorldJob.xml helloWorldJob"
2 [INFO] Scanning for projects...
3 [INFO]
4 [INFO] -----
5 [INFO] Building app-hello-spring-batch 1.0-SNAPSHOT
6 [INFO] -----
7 [INFO]
8 [INFO] --- exec-maven-plugin:1.4.0:java (default-cli) @ app-hello-spring-batch
  ↳ ---
9 Aug 27, 2015 3:45:24 PM
  ↳ org.springframework.context.support.ClassPathXmlApplicationContext
  ↳ prepareRefresh
10 INFO: Refreshing
  ↳ org.springframework.context.support.ClassPathXmlApplicationContext@28bcd128:
  ↳ startup date [Thu Aug 27 15:45:24 CDT 2015]; root of context hierarchy
11 Aug 27, 2015 3:45:24 PM
  ↳ org.springframework.beans.factory.xml.XmlBeanDefinitionReader
  ↳ loadBeanDefinitions
12 INFO: Loading XML bean definitions from class path resource [helloWorldJob.xml]
13 Aug 27, 2015 3:45:25 PM
  ↳ org.springframework.beans.factory.xml.XmlBeanDefinitionReader
  ↳ loadBeanDefinitions
14 INFO: Loading XML bean definitions from class path resource
  ↳ [applicationContext.xml]
15 Aug 27, 2015 3:45:25 PM
  ↳ org.springframework.batch.core.launch.support.SimpleJobLauncher
  ↳ afterPropertiesSet
16 INFO: No TaskExecutor has been set, defaulting to synchronous executor.
17 Aug 27, 2015 3:45:25 PM
  ↳ org.springframework.batch.core.launch.support.SimpleJobLauncher run
18 INFO: Job: [SimpleJob: [name=helloWorldJob]] launched with the following
  ↳ parameters: []
19 Aug 27, 2015 3:45:25 PM org.springframework.batch.core.job.SimpleStepHandler
  ↳ handleStep
20 INFO: Executing step: [taskletStep$child#2dcb0bb]
HelloAug 27, 2015 3:45:25 PM
  ↳ org.springframework.batch.core.job.SimpleStepHandler handleStep
22 INFO: Executing step: [taskletStep$child#66d6c674]
23 Aug 27, 2015 3:45:25 PM org.springframework.batch.core.job.SimpleStepHandler
  ↳ handleStep
24 INFO: Executing step: [taskletStep$child#1e7fle6b]
25 World!Aug 27, 2015 3:45:25 PM
  ↳ org.springframework.batch.core.launch.support.SimpleJobLauncher run
26 INFO: Job: [SimpleJob: [name=helloWorldJob]] completed with the following
  ↳ parameters: [] and the following status: [COMPLETED]
```

```

27 Aug 27, 2015 3:45:25 PM
   <-- org.springframework.context.support.ClassPathXmlApplicationContext doClose
28 INFO: Closing
   <-- org.springframework.context.support.ClassPathXmlApplicationContext@28bcd128:
   <-- startup date [Thu Aug 27 15:45:24 CDT 2015]; root of context hierarchy

```

This is hard to read, because it mixed with system output with our result output, so we'll direct output to a file:

```

1 c:\workspaces\workspace\my-projects\app-hello-spring-batch>mvn exec:java
   <-- -Dhttp.proxyHost=http.proxy.xyz.com -Dhttp.proxyPort=8000
   <-- -Dexec.mainClass=org.springframework.batch.core.launch.support.CommandLineJobRunner
   <-- -Dexec.args="helloWorldJob.xml helloWorldJob" > result.txt

```

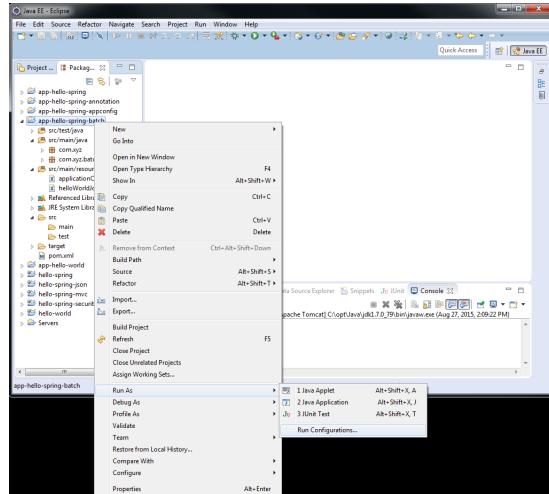
Result:

```

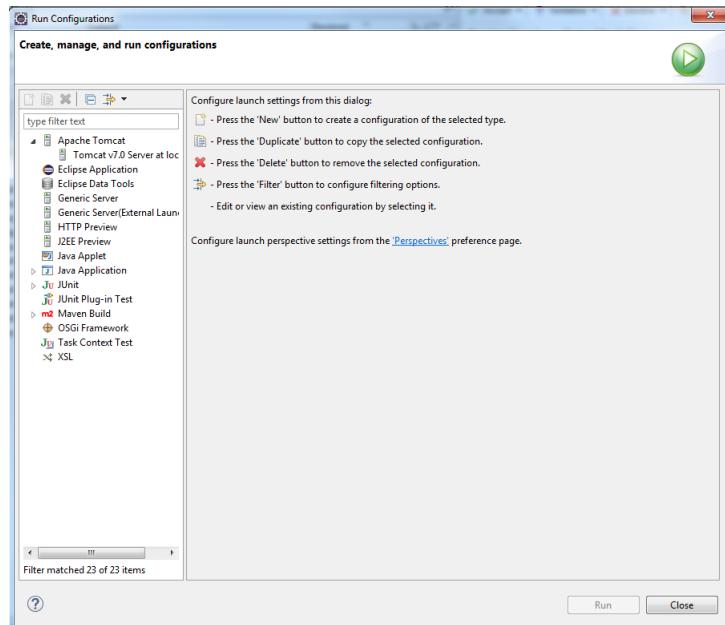
1 c:\workspaces\workspace\my-projects\app-hello-spring-batch>type result.txt
2 [INFO] Scanning for projects...
3 [INFO]
4 [INFO] -----
5 [INFO] Building app-hello-spring-batch 1.0-SNAPSHOT
6 [INFO] -----
7 [INFO]
8 [INFO] --- exec-maven-plugin:1.4.0:java (default-cli) @ app-hello-spring-batch
   <-- ---
9 Hello World!

```

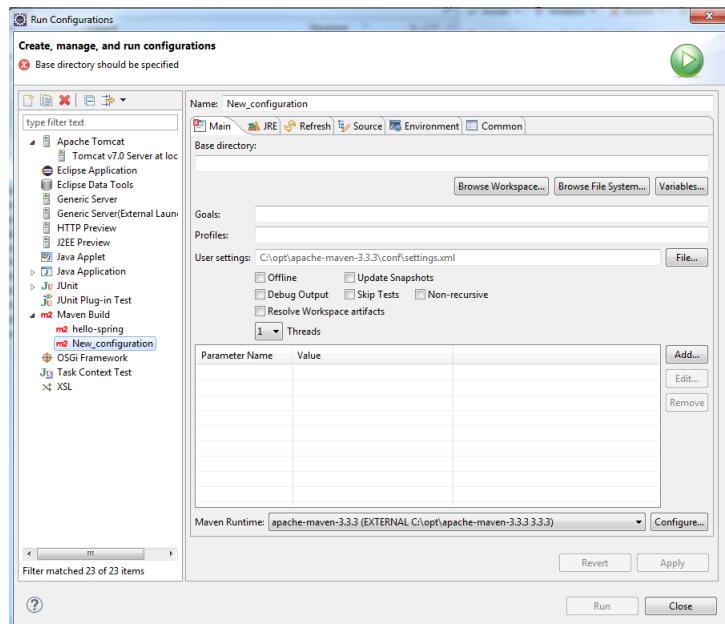
4.3.10 Run App - in eclipse



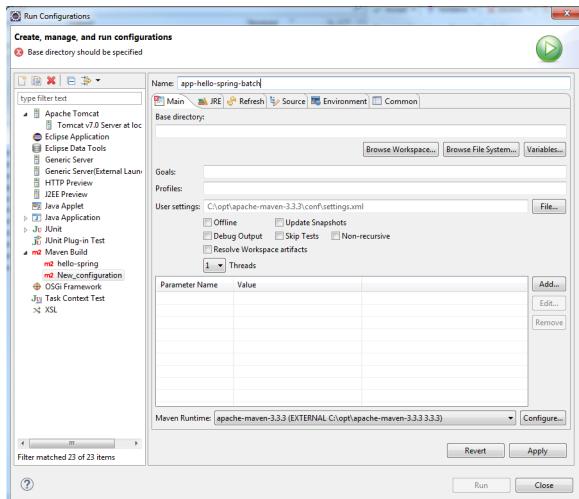
Right click on project ->Run As ->Run Configuration...



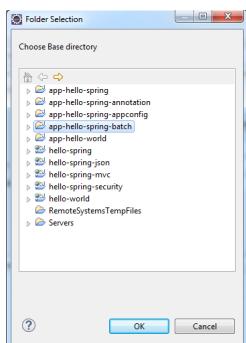
Select Maven Build, click the New icon on the top-left corner



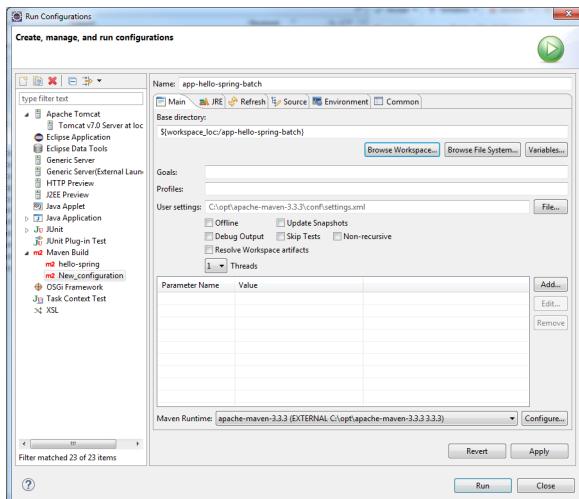
Type in Name: app-hello-spring-batch



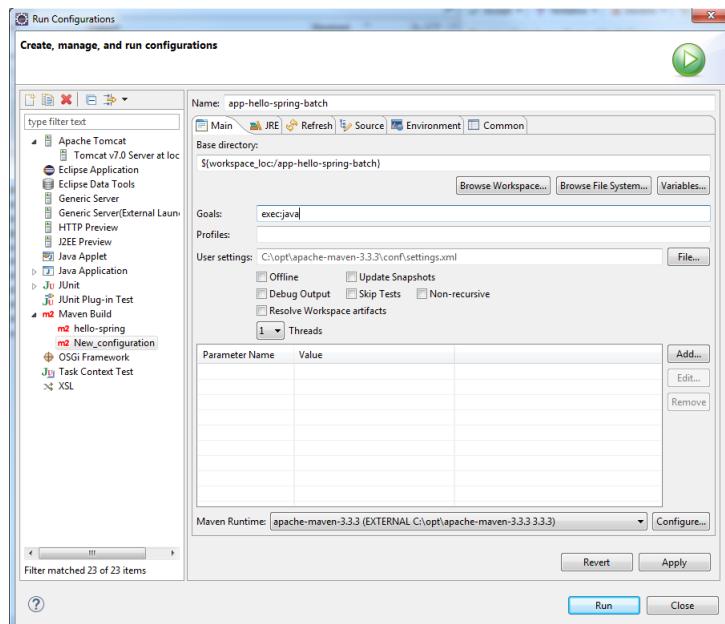
Click Browse Workspace...



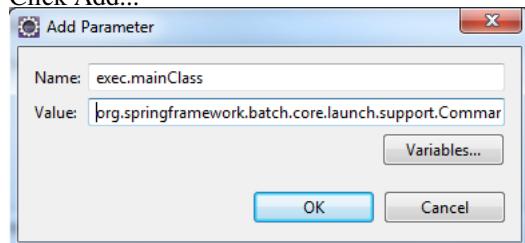
Select app-hello-spring-batch



Type in Goals: exec:java

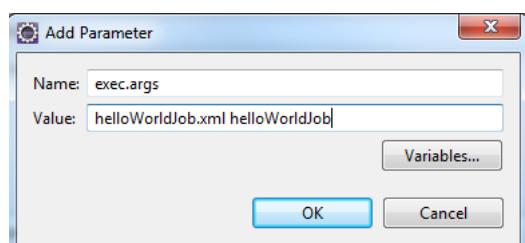


Click Add...



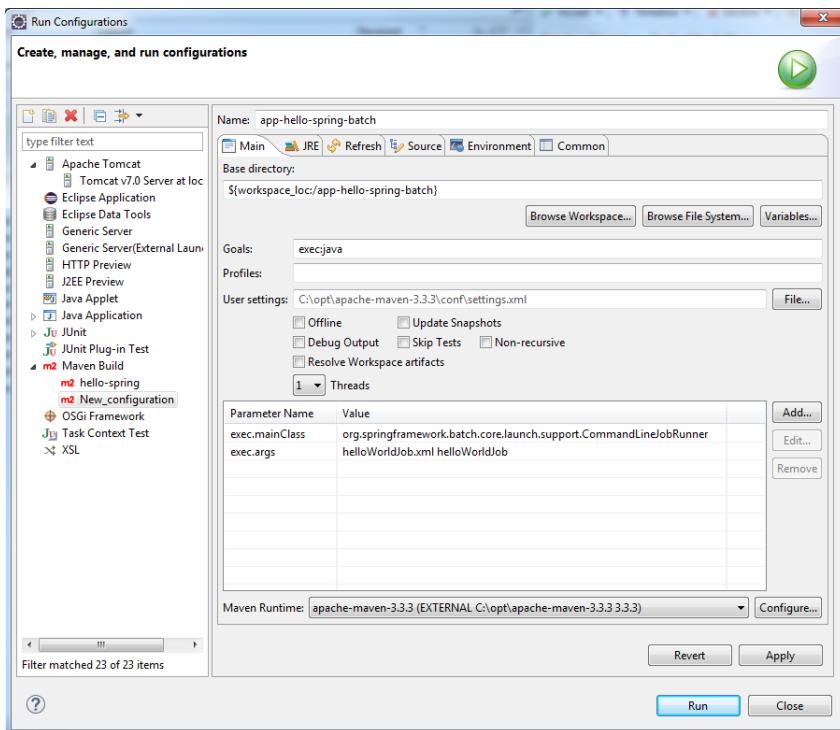
Name: exec.mainClass

Value: org.springframework.batch.core.launch.support.CommandLineJobRunner

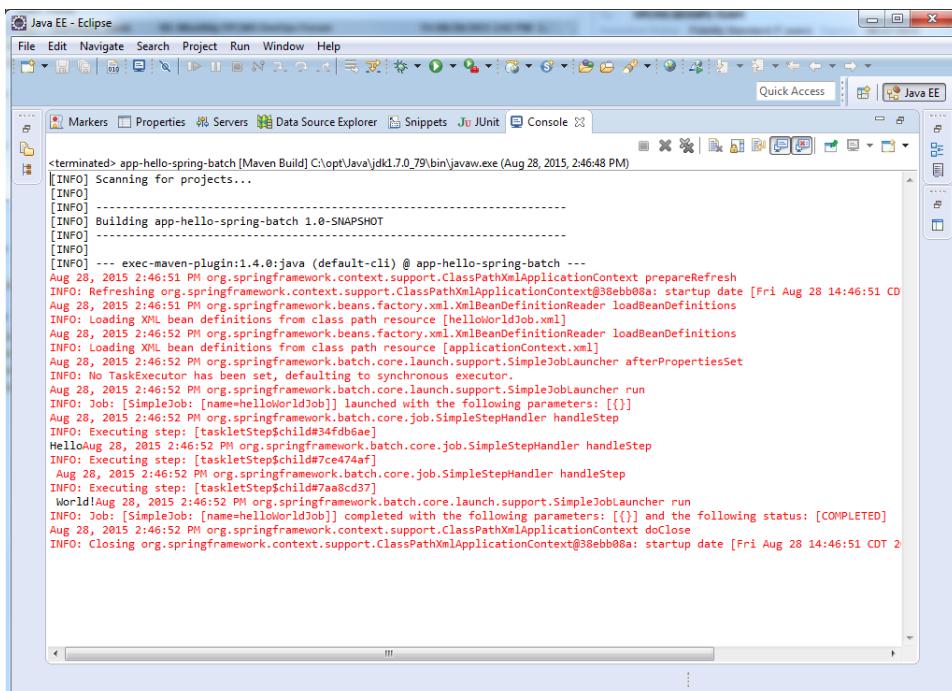


Name: exec.args

Value: helloWorldJob.xml helloWorldJob



Click Apply, Run, then maximize console window



PART II

VIRTUAL MACHINE

CHAPTER 5

VIRTUAL MACHINE

In computing, a virtual machine (VM) is an emulation of a particular computer system. Virtual machines operate based on the computer architecture and functions of a real or hypothetical computer, and their implementations may involve specialized hardware, software, or a combination of both.

5.1 Hardware

First you need to go to your machine's BIOS settings, Enable Virtualization Technology, and / or VT-d Feature.

5.2 Oracle VM VirtualBox

VirtualBox is a cross-platform virtualization application. What does that mean? For one thing, it installs on your existing Intel or AMD-based computers, whether they are running Windows, Mac, Linux or Solaris operating systems. Secondly, it extends the capabilities of your existing computer so that it can run multiple operating systems (inside multiple virtual machines) at the same time. So, for example, you can run Windows and Linux on your Mac, run Windows Server 2008 on your Linux server, run Linux on your Windows PC, and so on, all alongside your existing applications. You can install and run as many virtual machines as you like – the only practical limits are disk space and memory.

5.2.1 Download Oracle VM VirtualBox

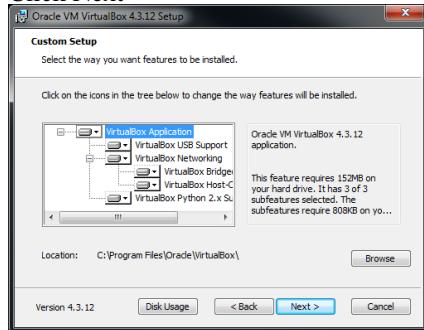
- Some newer version of Oracle VM VirtualBox cannot start if you do not have windows administrator rights. So we use an older version 4.3.12
- Download VirtualBox 4.3.12 for Windows hosts (VirtualBox-4.3.12-93733-Win.exe) from:
<https://www.virtualbox.org/wiki/Downloads>
 Go to VirtualBox older builds ->VirtualBox 4.3 ->VirtualBox 4.3 older builds ->VirtualBox 4.3.12 (released May 16th 2014) ->Windows hosts x86/AMD64
https://www.virtualbox.org/wiki/Download_Old_Builds_4_3_pre24
<http://download.virtualbox.org/virtualbox/4.3.12/>

5.2.2 Install Oracle VM VirtualBox

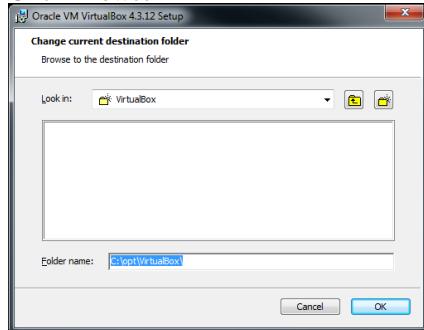
- Right click VirtualBox-4.3.12-93733-Win.exe, Run as administrator



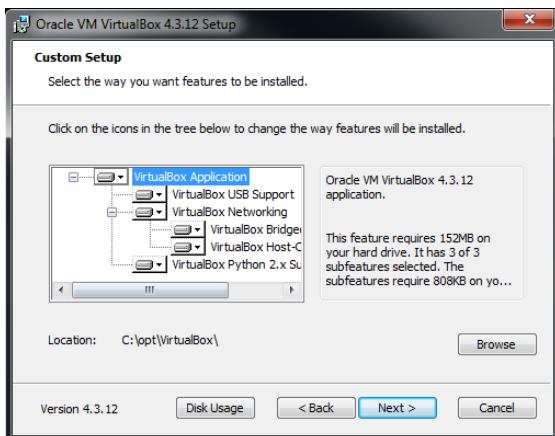
Click Next



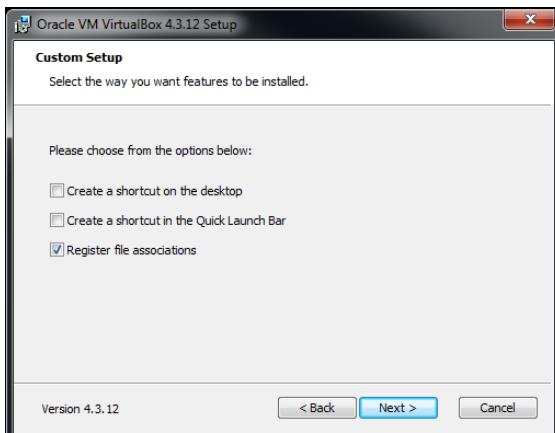
Click Browse



Change Folder name to: C:\opt\Oracle\VirtualBox\ , click OK



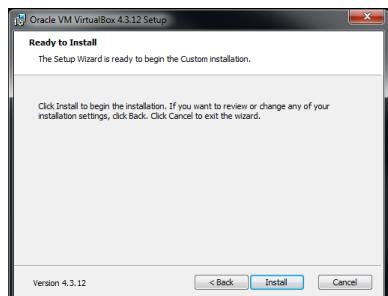
Click Next



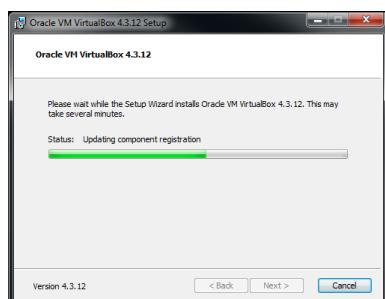
Uncheck Create a shortcut on the desktop and Create a shortcut in the Quick Launch Bar, click Next



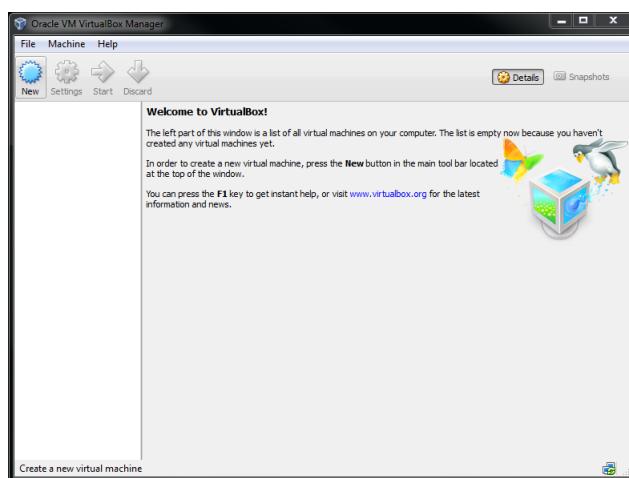
Click Yes



Click Install

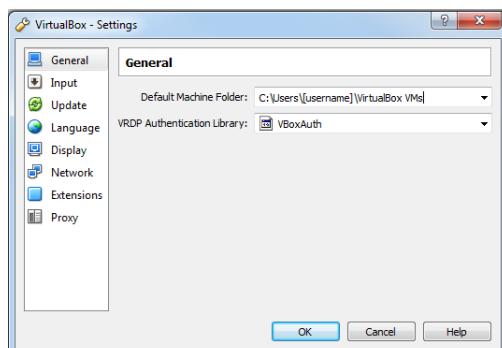
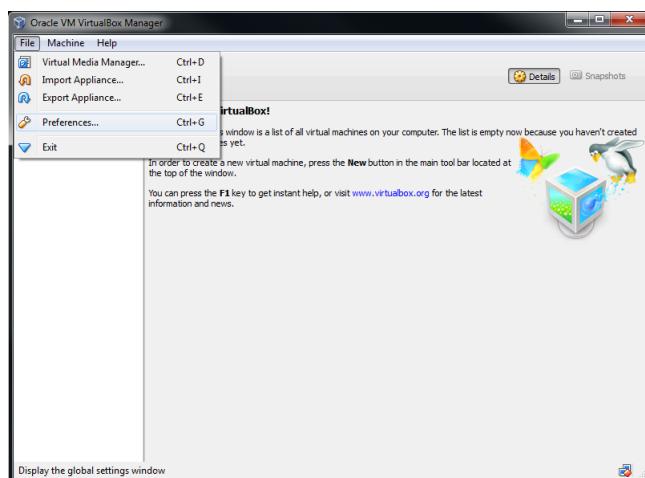


Click Finish

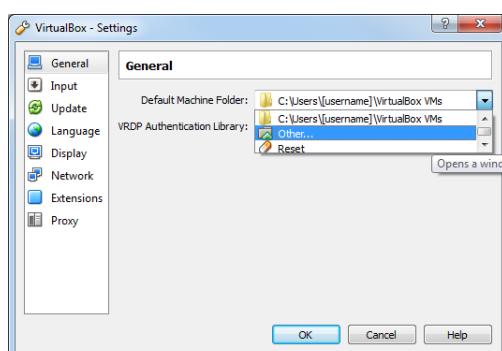


5.2.3 Configure Oracle VM VirtualBox

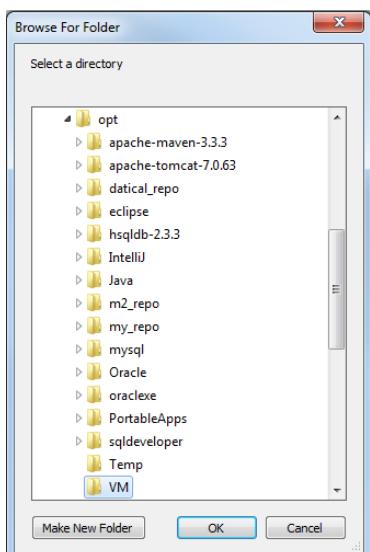
- Create a folder C:\opt\VM first
- Go to File ->Preferences...



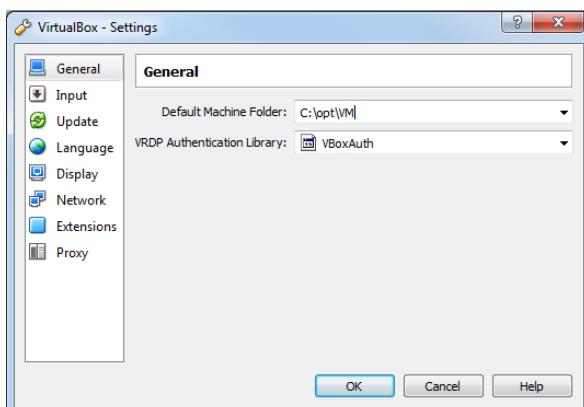
Default Machine Folder is C:\Users\[username]\VirtualBox VMs
Click the down arrow on the right side of Default Machine Folder



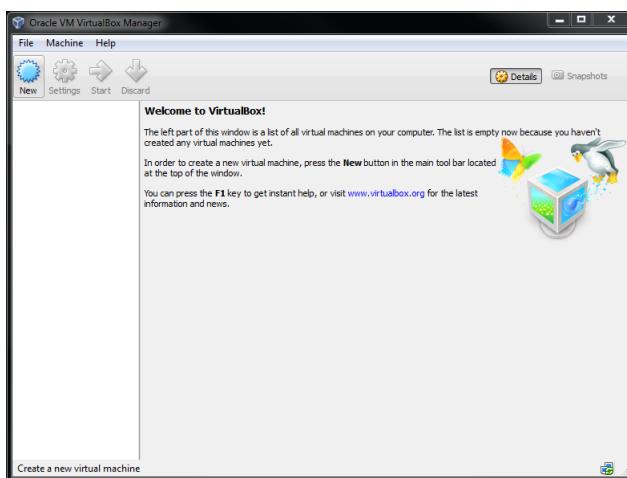
Select Other...



Change Folder name to: C:\opt\VM, click OK



Click OK



5.3 Oracle VM VirtualBox - Mac

5.3.1 Download Oracle VM VirtualBox

- Download VirtualBox 5.0.24 for OS X hosts(VirtualBox-5.0.24-108355-OSX.dmg) and Extension Pack(Oracle_VM_VirtualBox_Extension_Pack-5.0.24-108355.vox-extpack) from:
<https://www.virtualbox.org/wiki/Downloads>

5.3.2 Install Oracle VM VirtualBox

Double click VirtualBox-5.0.24-108355-OSX.dmg file to lanuch it



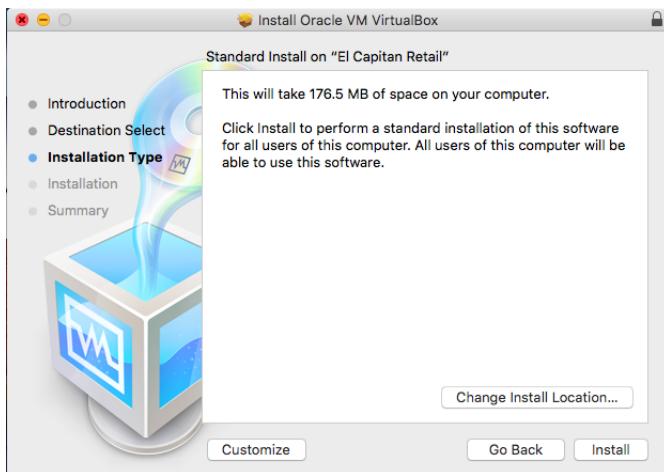
Double click the package icon to launch the Install app



Click Continue

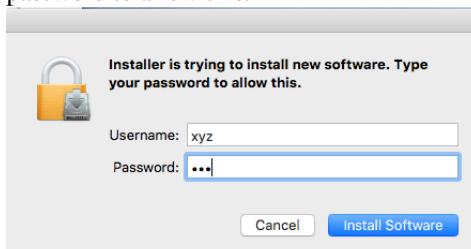


Click Continue



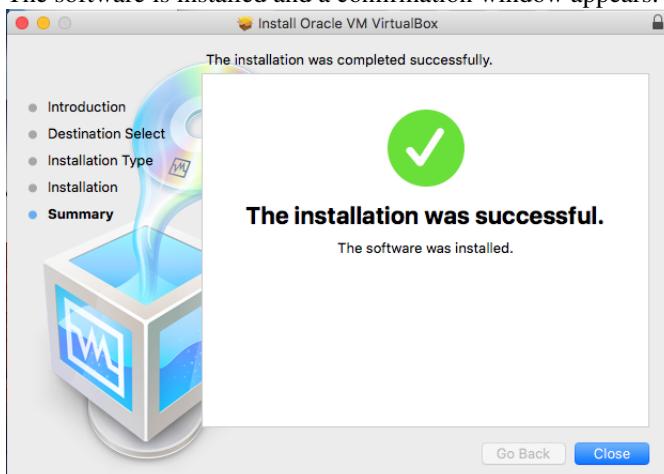
Click Install

A window appears that says "Installer is trying to install new software. Type your password to allow this."



Enter the Administrator login and password and click Install Software.

The software is installed and a confirmation window appears.

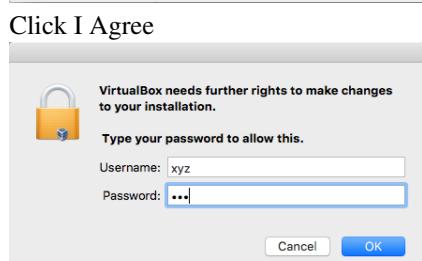
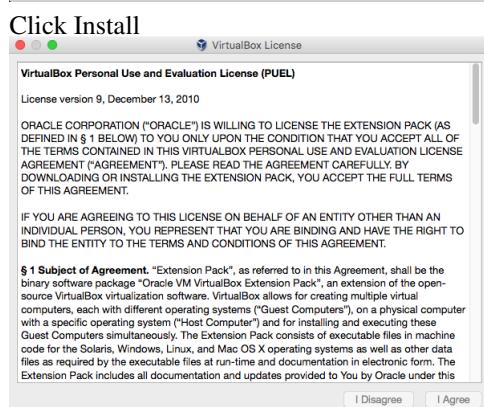
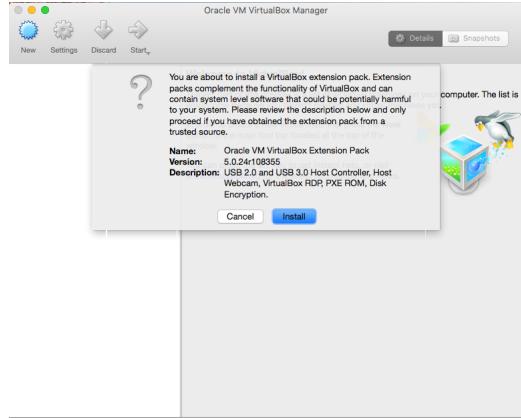


Click Close

After the software is installed, delete the .dmg file if you want to save disk space.

5.3.3 Install Oracle VM VirtualBox Extension Pack

Double click Oracle_VM_VirtualBox_Extension_Pack-5.0.24-108355.vox-extrpack file to launch it



Enter the Administrator login and password and click OK



Click OK

After the software is installed, delete the extonsion pack file if you want to save disk space.

5.3.4 Configure Oracle VM VirtualBox

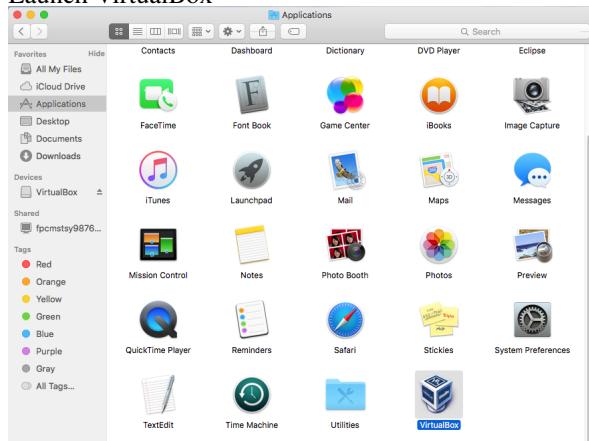
- Create a folder /opt/VM first

```

1 $ sudo mkdir -p /opt/VM
2
3 $ sudo chown -R xyz /opt/VM
4
5 $ sudo chgrp -R staff /opt/VM

```

- Launch VirtualBox



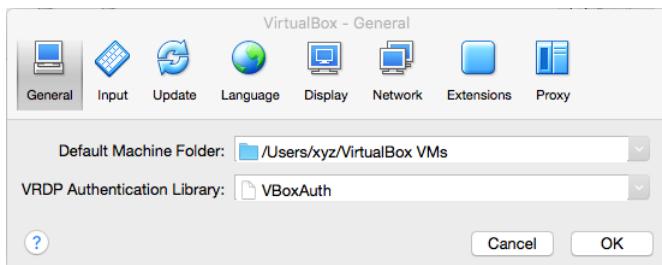
Double click the Applications/VirtualBox icon.



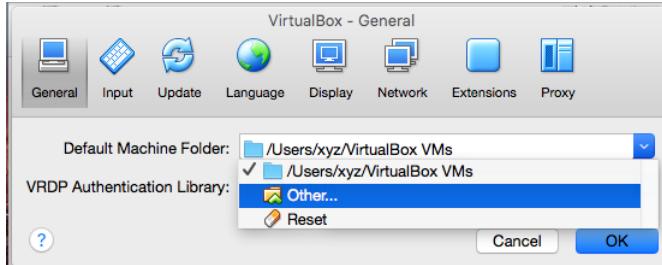
VirtualBox started.

- Go to VirtualBox → Preferences...

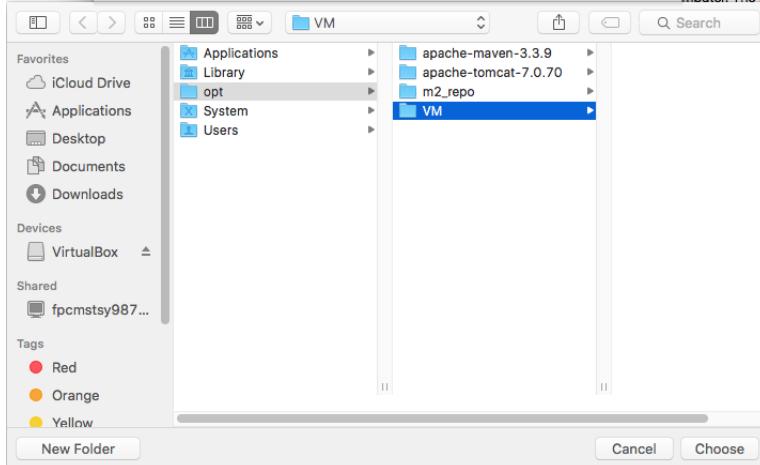




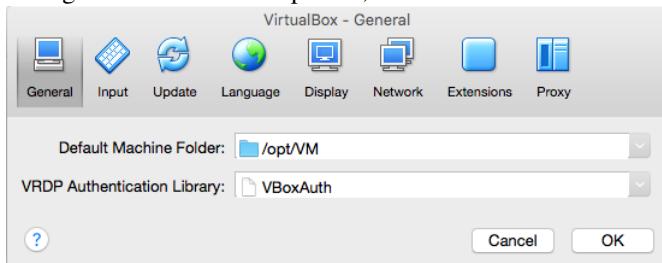
Default Machine Folder is /Users/[username]/VirtualBox VMs



Click the down arrow on the right side of Default Machine Folder
Select Other...



Change Folder name to: /opt/VM, click Choose



Click OK

CHAPTER 6

LINUX ON VIRTUAL MACHINE

6.1 Linux

We are going to install Oracle Linux Operating System

6.1.1 Register for Download

- Download Oracle Linux 7 Update 1 for x86_64 (64 bit) (V74844-01.iso) from:
<http://www.oracle.com/technetwork/indexes/downloads/index.html>
Click Sign in/Register <https://login.oracle.com/mysso/signon.jsp>
Click Create Account
Fill in the details
Click Create Account

6.1.2 Download Oracle Linux - Windows

- Download Oracle Linux 7 Update 1 for x86_64 (64 bit) (V74844-01.iso) from:
<http://www.oracle.com/technetwork/indexes/downloads/index.html>
Servers and Storage Systems → Oracle Linux and Oracle Enterprise Kernel
Login to Oracle Software Delivery Cloud
Search by Release, type in Linux

The screenshot shows the Oracle Software Delivery Cloud interface. At the top, there's a search bar and navigation links for 'FAQ', 'hongsong.zhou@jpmorgan.com', 'English', and 'Sign Out'. Below the header, a message says 'Need Help? Contact Software Delivery Customer Service'. A note instructs users to add items to their download queue by entering a product or release name and selecting a platform. The main area is titled 'Download Queue' and contains a table with one row:

Selected Item	Platform
Release: Oracle Linux 7.1	x86 64 bit

At the bottom of the queue table, there are 'Remove All' and 'Continue' buttons. A note below the table states: 'You must agree to Oracle's trial license terms before downloading products that you do not have a current valid license to use.' The footer includes links for 'About Oracle', 'Legal Notices', 'Terms of Use', 'Your Privacy Rights', and social media icons.

Click Continue

The screenshot shows the same Oracle Software Delivery Cloud interface as the previous one, but the 'Release' column in the 'Download Queue' table is checked. The table now displays the selected item: 'Release: Oracle Linux 7.1 for x86 64 bit, 4 files' and 'Selected Item: Oracle Linux 7.1'. Other columns show 'Applicable Terms & Restrictions: Oracle Standard Terms and Restrictions', 'Size: 9.8 GB', and 'Published Date: Mar 16, 2015'. The 'Continue' button is visible at the bottom right of the queue table.

Select Oracle Linux 7.1 for x86 64 bit, 4 files, click Continue

The screenshot shows a modal dialog box for accepting a license agreement. At the top left is an information icon with the text 'Please read the following license agreements carefully.' Below this is a 'Print' button. The main content area contains the following text:

S. MISCELLANEOUS
Upon 45 days written notice, Oracle may audit your use of the Programs. You agree to cooperate with Oracle's audit and provide reasonable assistance and access to information. You agree to pay within 30 days of written notification any unpaid fees. If you do not pay, Oracle can end your technical support, licenses and/or this Trial License.

Oracle Employees: Under no circumstances are Oracle Employees authorized to download software for the purpose of distributing it to customers. Oracle products are available to employees for internal use or demonstration purposes only. In keeping with Oracle's trade compliance obligations under U.S. and applicable multilateral law, failure to comply with this policy could result in disciplinary action up to and including termination.

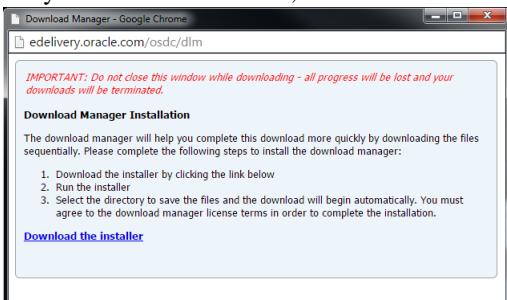
Last updated 21 November 2014

At the bottom of the dialog, there is a link 'Scroll to read the full license agreement' and a checkbox labeled 'I have reviewed and accept the terms of the Commercial License, Special Programs License, and/or Trial License.' Below the checkbox are 'Cancel' and 'Continue' buttons.

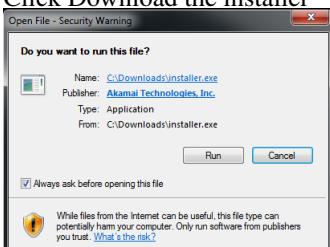
Click Continue



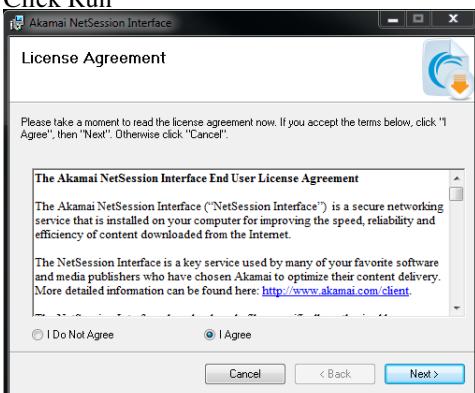
Only check V74844-01.iso, click Download



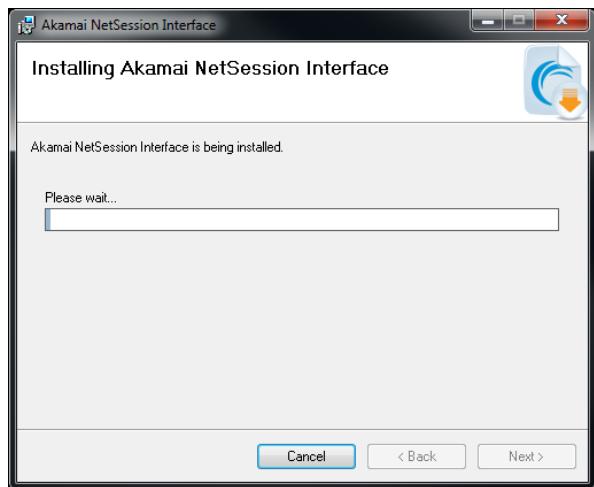
Click Download the installer



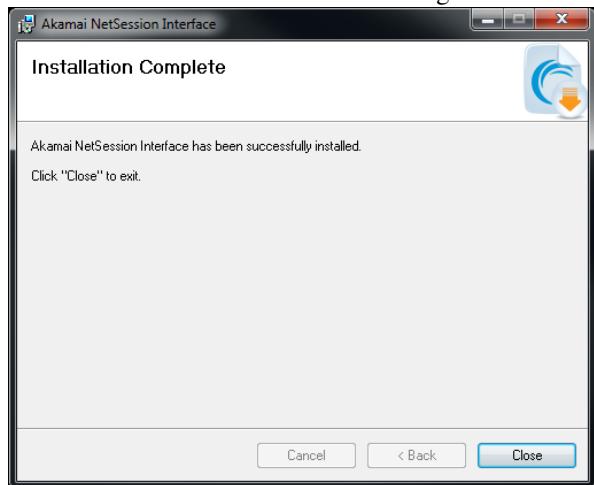
Click Run



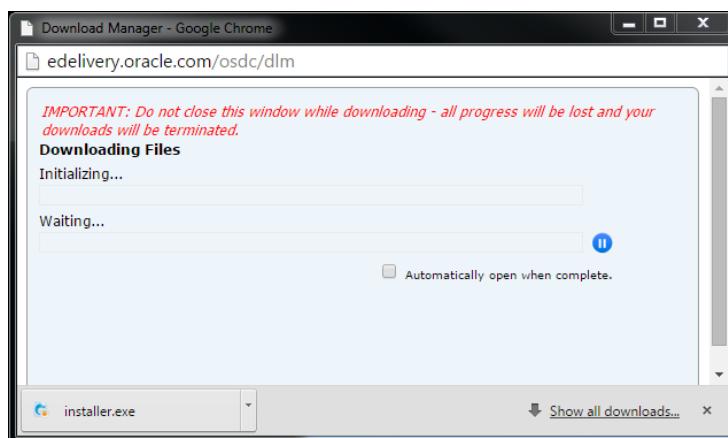
Click I Agree, then click Next



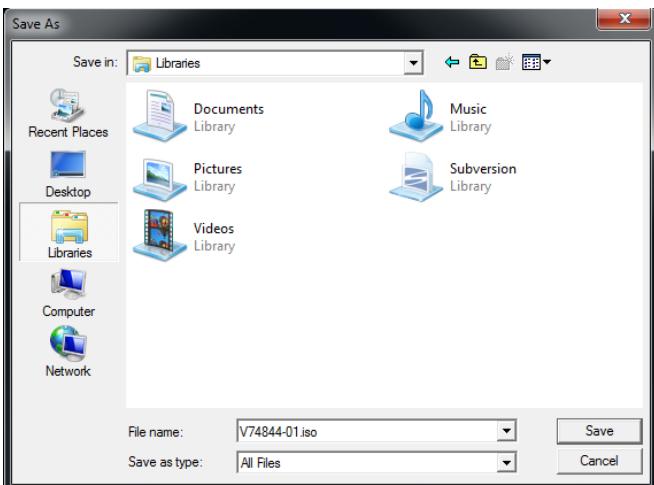
Akamai NetSession Interface is installing...



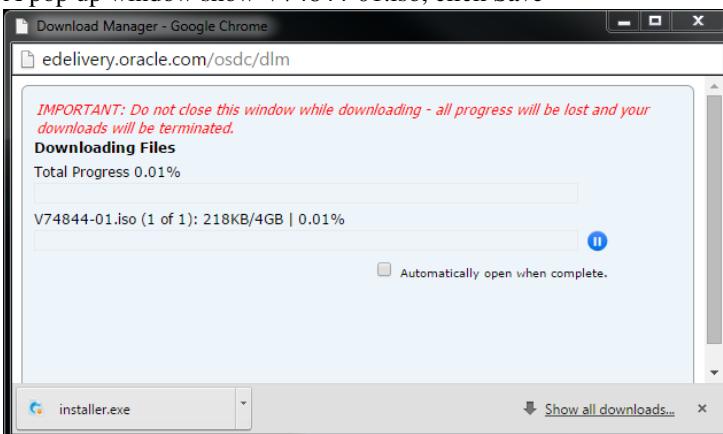
Click Close



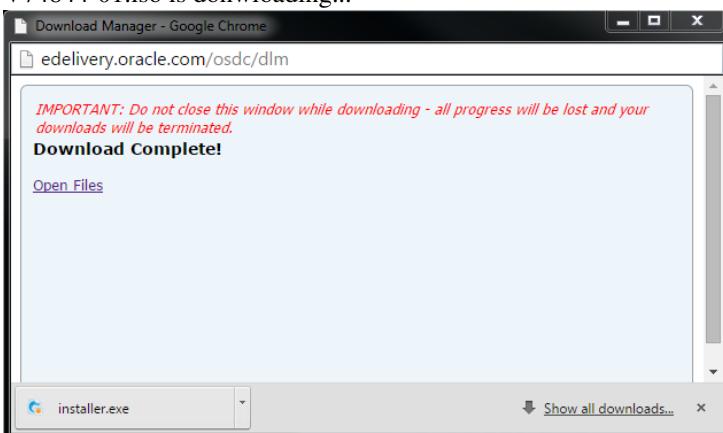
In the mean time, Download Manager page start Initializing...



A pop up window show V74844-01.iso, click Save



V74844-01.iso is downloading...



Download Completed

6.1.3 Download Oracle Linux - Mac

- Download Oracle Linux 7 Update 1 for x86_64 (64 bit) (V74844-01.iso) from:

<http://www.oracle.com/technetwork/indexes/downloads/index.html>

Servers and Storage Systems → Oracle Linux and Oracle Enterprise Kernel

Login to Oracle Software Delivery Cloud

Search by Release, type in Linux

The screenshot shows a search results page for 'Linux'. At the top, there's a navigation bar with 'ORACLE CLOUD', 'Oracle Software Delivery Cloud', 'FAQ', 'Hongxiong.Zhou@jpmorgan.com', 'English', and 'Sign Out'. Below the navigation is a search bar with placeholder text 'If more than one release is available, you may select an alternate release by clicking on the "Select Alternate Release..." link.' A table titled 'Download Queue' lists a single item: 'Oracle Linux 7.1 for x86_64 bit; 4 files'. The table has columns for 'Selected Item', 'Applicable Terms & Restrictions', 'Size', and 'Published Date'. The item details are: Oracle Linux 7.1, Oracle Standard Terms and Restrictions, 9.8 GB, and Mar 16, 2015. At the bottom of the page are buttons for '< Return to Search' and 'Continue'.

Select Oracle Linux 7.1 for x86 64 bit, 4 files, click Continue

The screenshot shows a 'File Download' dialog box. It contains instructions: 'You may download files:' with bullet points 'Using the download manager - Select the checkboxes next to the desired files, then click 'Download'' and 'Individually - Click the file name to download'. Below this is a list of files under 'Oracle Linux (7.1) for x86 64 bit':

File Name	Description	Size
<input checked="" type="checkbox"/> V74844-01.iso	Oracle Linux 7 Update 1 for x86_64 (64 bit)	4.0 GB
<input type="checkbox"/> V74845-01.iso	Oracle Linux Release 7 Update 1 Boot iso image for x86_64 (64 bit)	341.0 MB
<input type="checkbox"/> V74846-01.iso	Oracle Linux Release 7 Update 1 source DVD 1	3.3 GB
<input type="checkbox"/> V74847-01.iso	Oracle Linux Release 7 Update 1 source DVD 2	2.1 GB

At the bottom of the dialog, it says 'Total 4 files', 'About 11 hrs (at 256KB/sec)', 'Total Size 9.8 GB', and includes links for 'WGET Options', 'View Digest Details', '< Back', 'Restore', and a large 'Download' button.

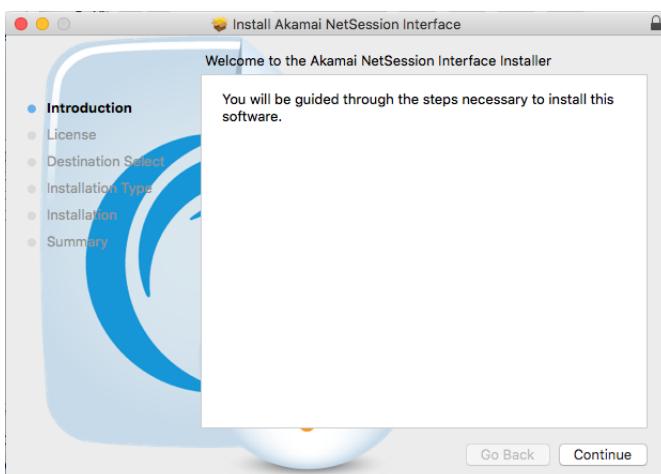
Only check V74844-01.iso, click Download

The screenshot shows a download progress window for 'edelivery.oracle.com'. It features a progress bar at the top with three colored dots (red, yellow, green). Below the progress bar is the URL 'edelivery.oracle.com'. A red warning message reads: 'IMPORTANT: Do not close this window while downloading - all progress will be lost and your downloads will be terminated.' A section titled 'Download Manager Installation' explains that the download manager will help complete the download faster. It lists three steps: 1. Download the installer, 2. Run the installer, 3. Select the directory to save the files. A blue link 'Download the installer' is visible at the bottom.

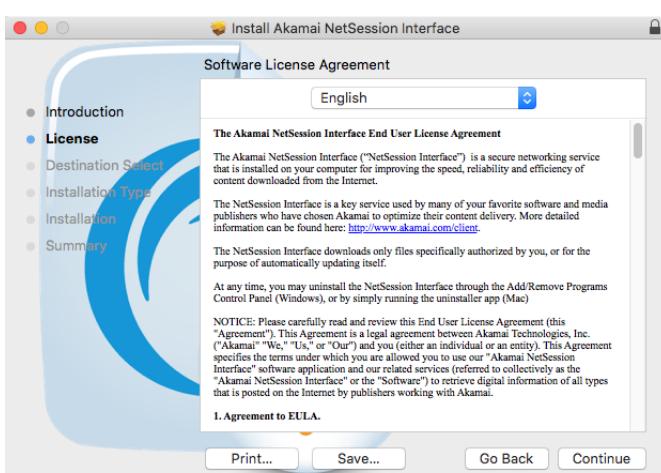
Click Download the installer



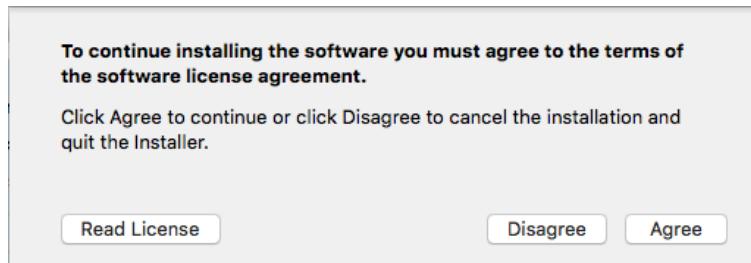
Double click Akamai NetSession Interface.pkg



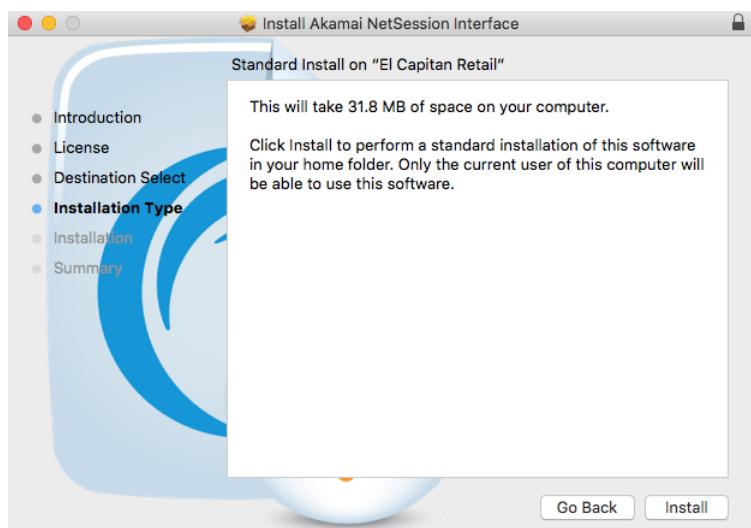
Click Continue



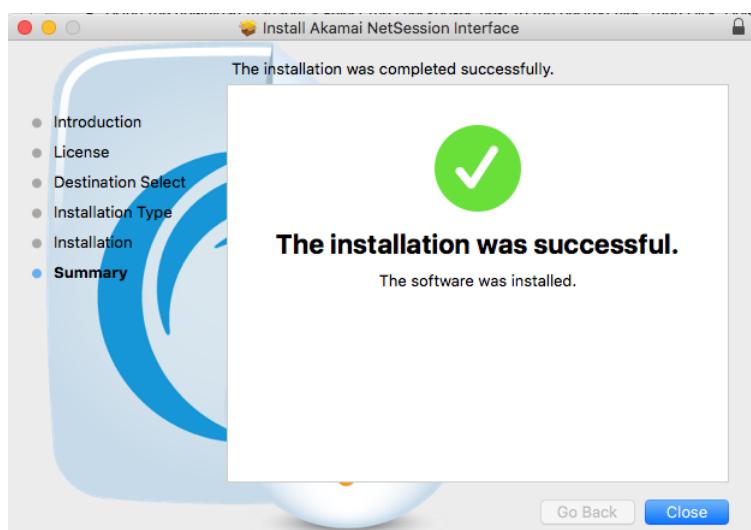
Click Continue



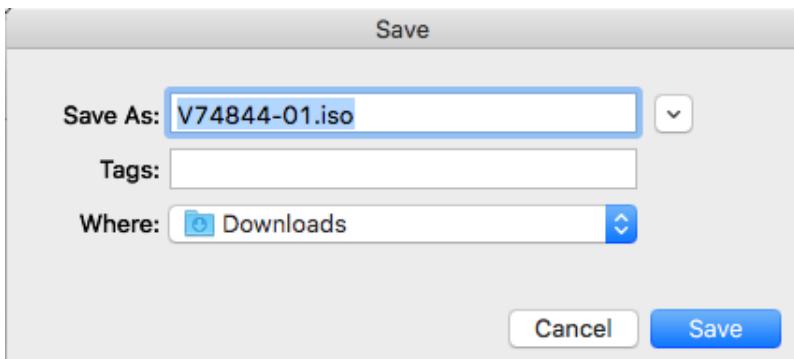
Click Agree



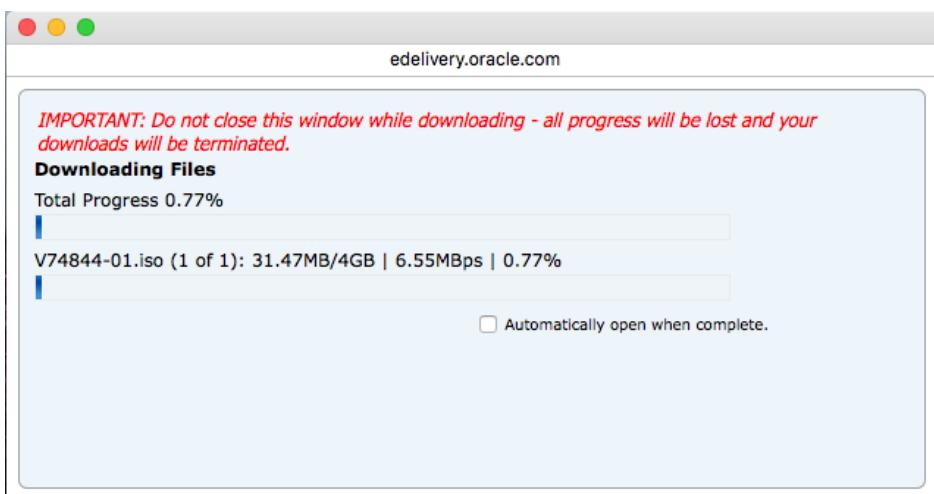
Click Install



Click Close



Click Save



V74844-01.iso is downloading...



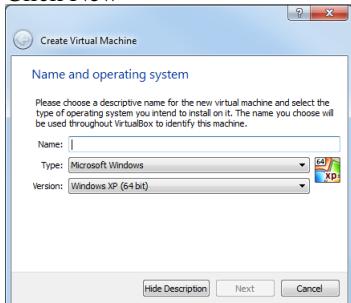
Download Completed

6.1.4 Install Oracle Linux - Create VM

- These steps apply to both Windows and Mac
- Launch Oracle VM VirtualBox



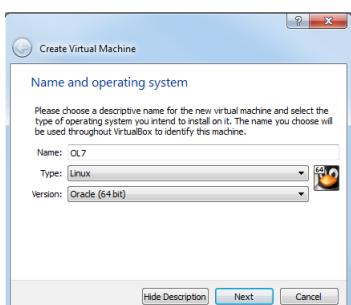
Click New



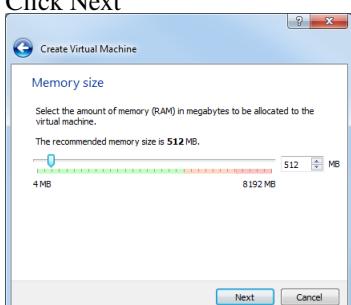
Name: OL7

Type: Linux

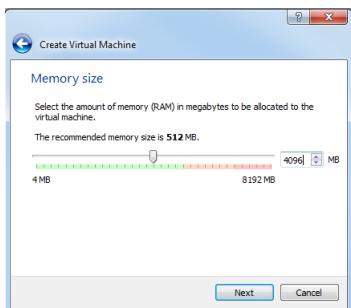
Version: Oracle (64-bit)



Click Next



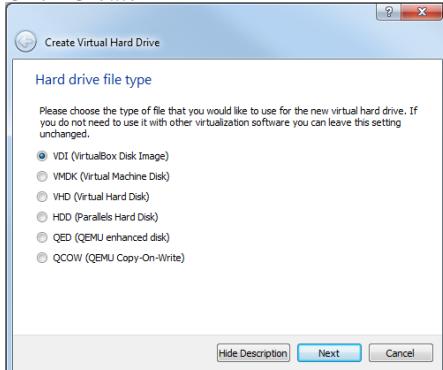
Change Memory size to 4G - 4096 MB



Click Next



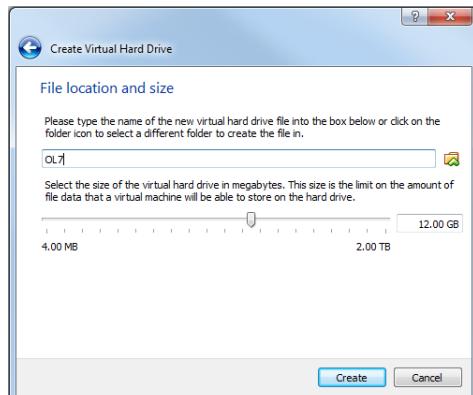
Click Create



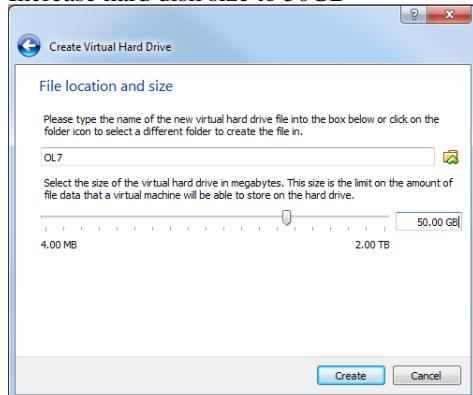
Click Next



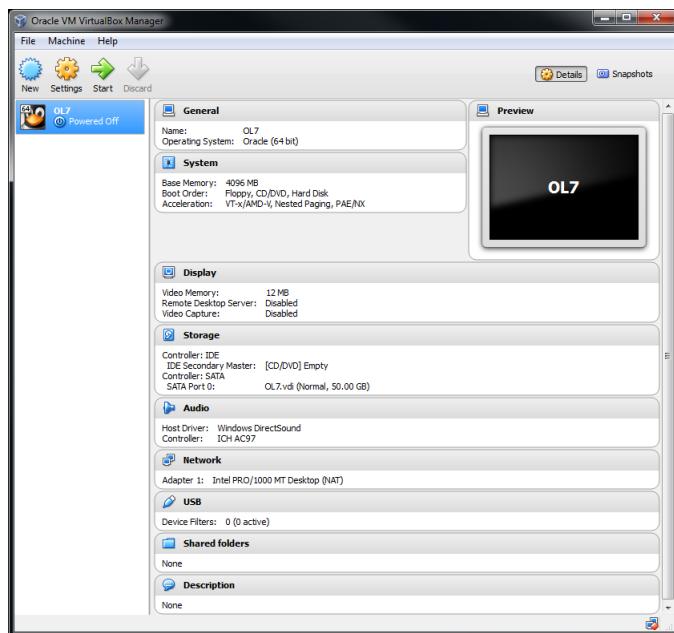
Click Next



Increase hard disk size to 50GB

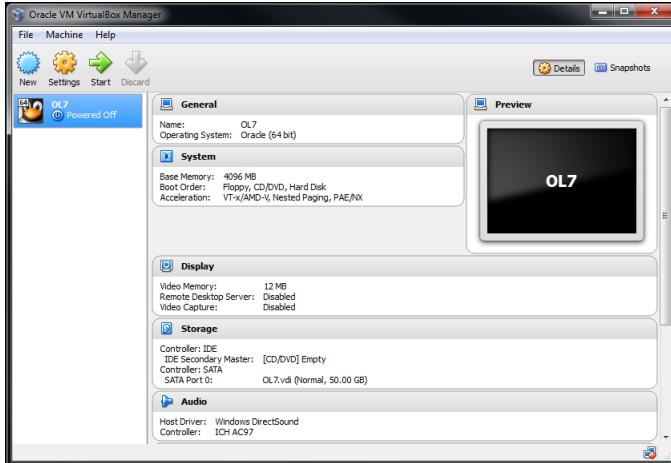


Click Create

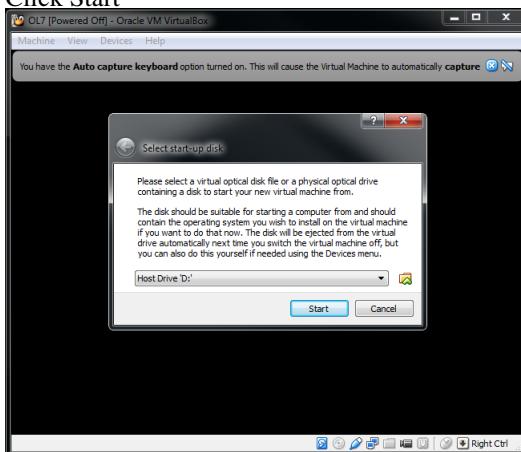


6.1.5 Install Oracle Linux - From Oracle iso

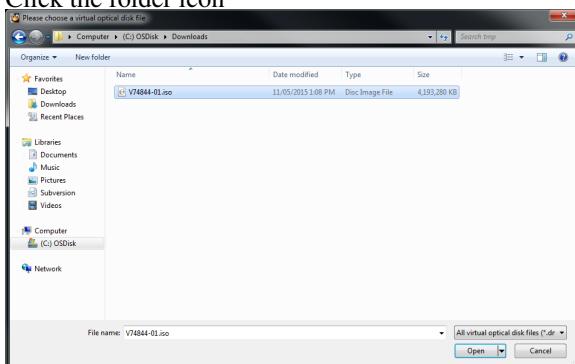
- These steps apply to both Windows and Mac
- Launch Oracle VM VirtualBox



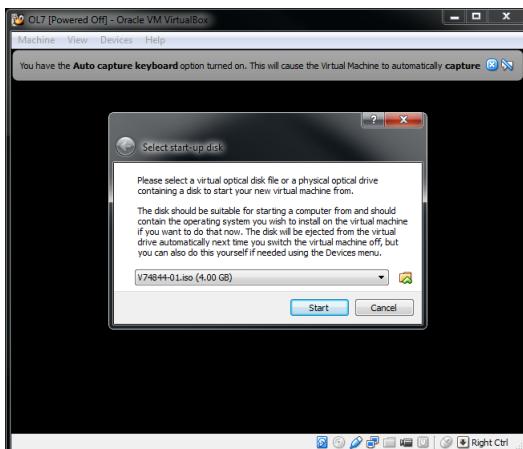
Click Start



Click the folder icon



Browse to Oracle Linux 7 iso file V74844-01.iso, click Open



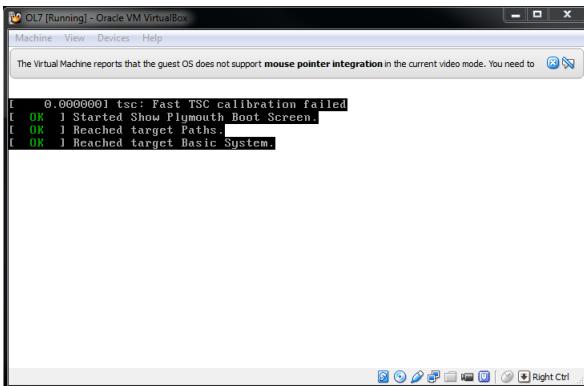
Click Start



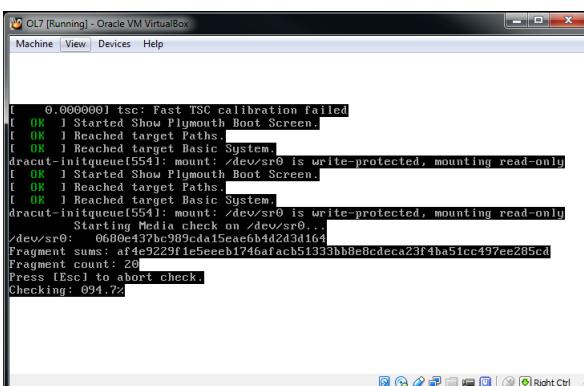
Click the X mark on the top right corner to close the message bar



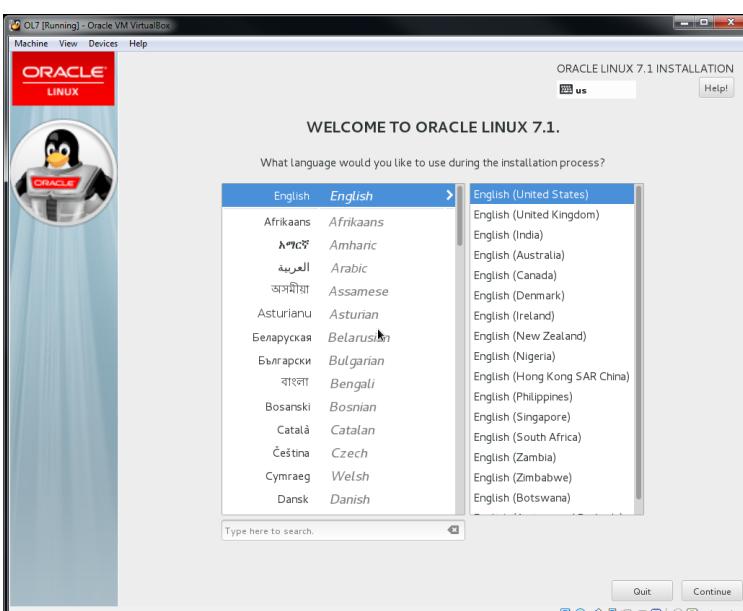
Wait for 60 seconds, it will automatically boot



Click the X mark on the top right corner to close the message bar

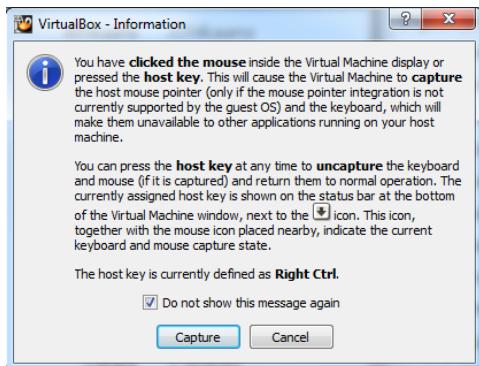


Oracle Linux booting...

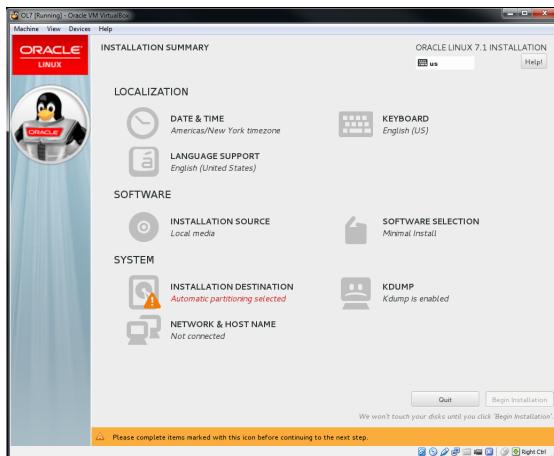


Click Continue

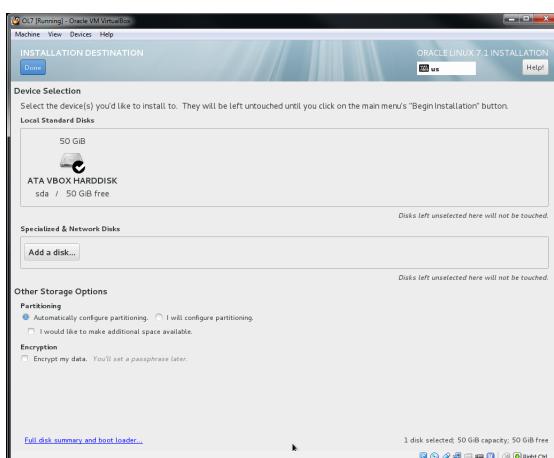
From now on, the mouse movement is within VM box, if you want to move mouse outside of VM box, click Right Ctrl and move mouse out of the VM box.



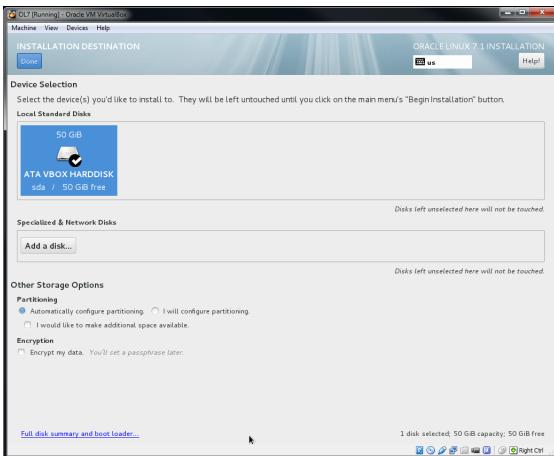
If you see this message box, check Do not show this message again, then click Capture



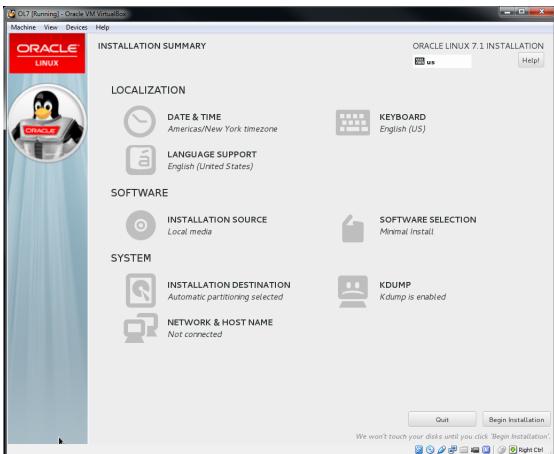
Click INSTALLATION DESTINATION



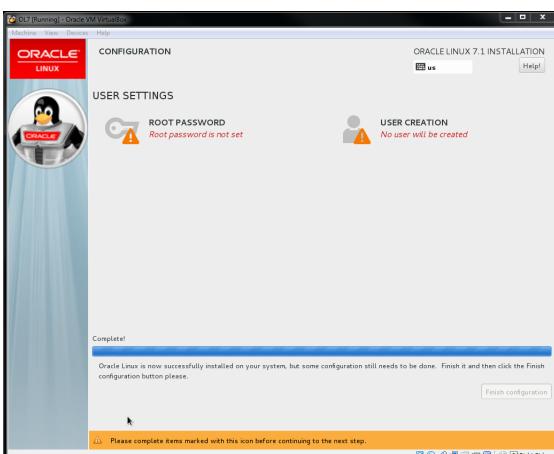
Double Click the harddisk icon



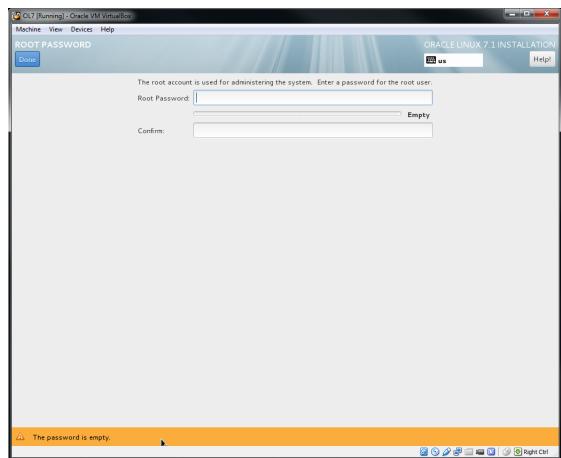
Click Done on the top left corner



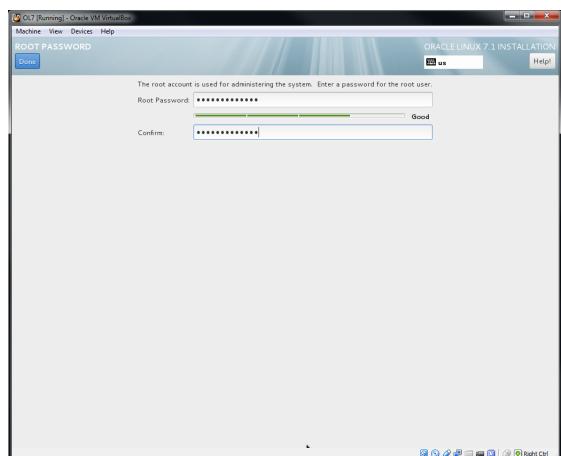
Click Begin Installation



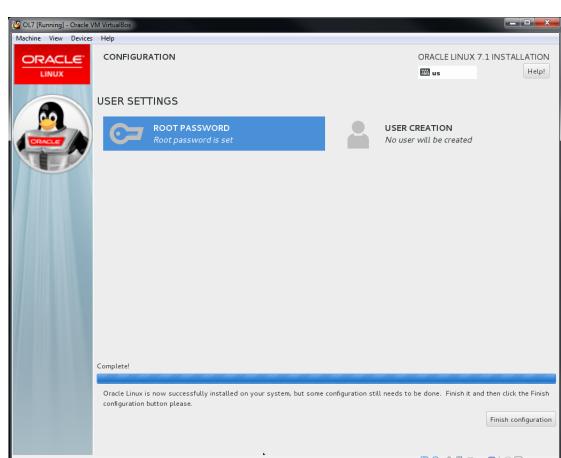
Click ROOT PASSWORD



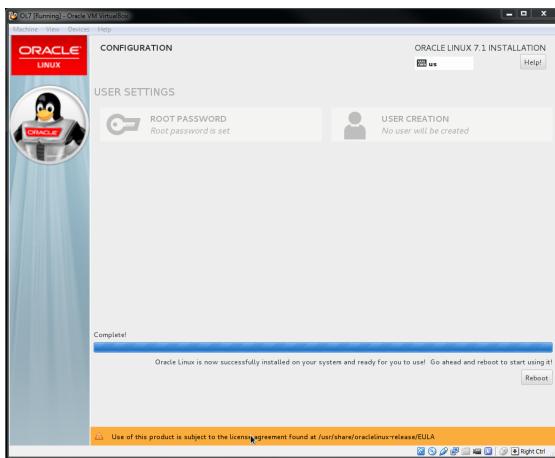
Type in Root Password



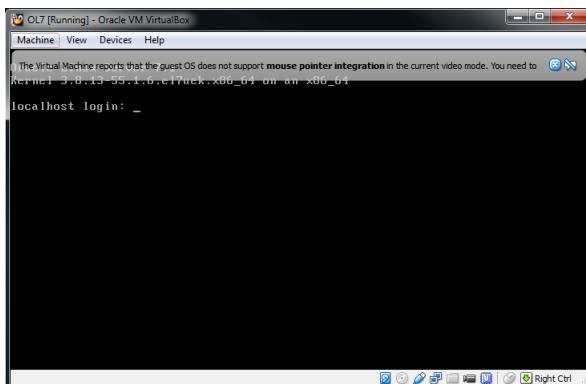
Click Done on the top left corner



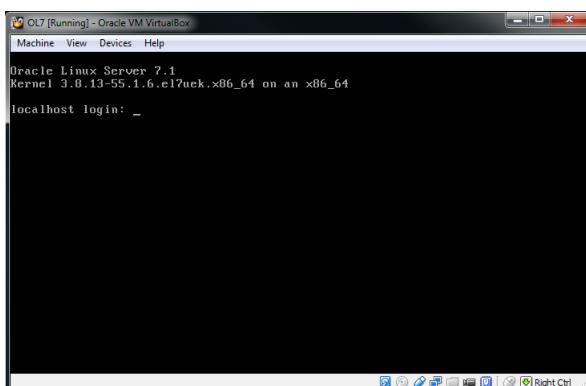
Click Finish configuration



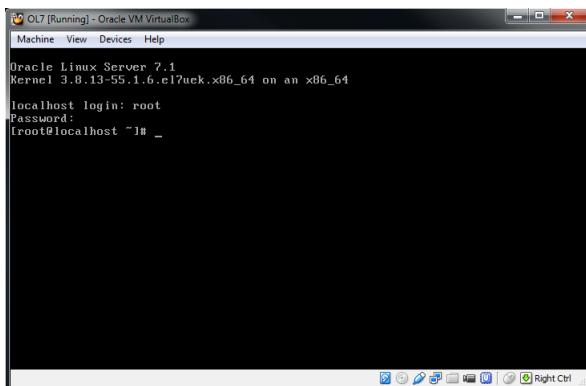
Click Reboot



Click the X mark on the top right corner to close the message bar



Type in your login id (root) and password

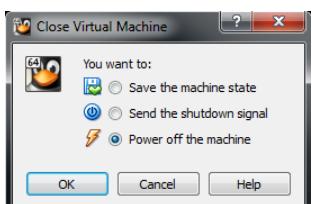


Login successful.

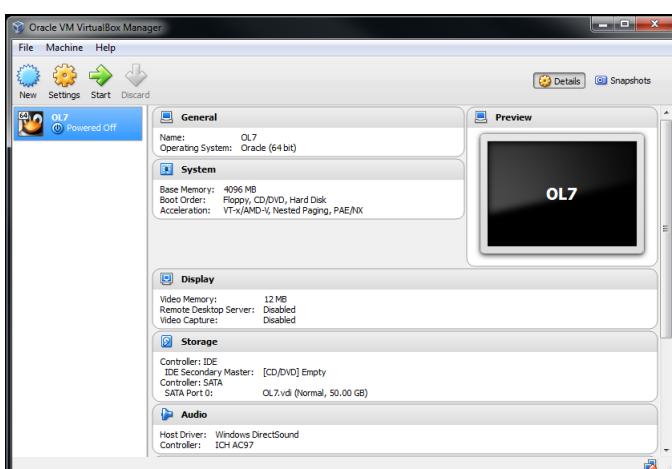
- Click the X mark on the top right corner to close the window



You will have choices to Save the machine state or Power off the machine.



Click Power off the machine



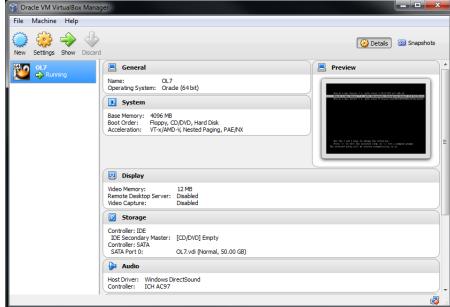
VM OL7 Powered off

6.2 Install Linux Guest Additions

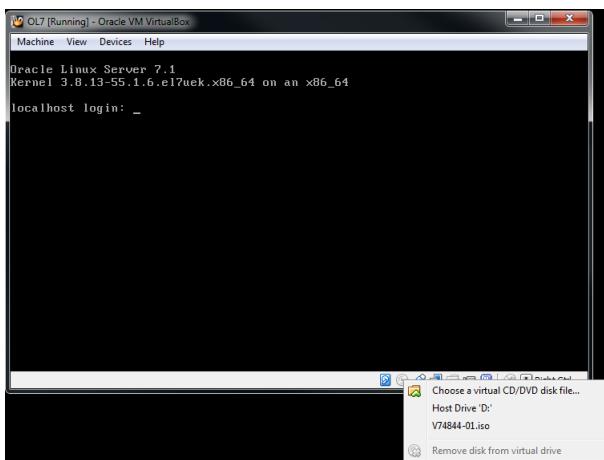
These steps apply to both Windows and Mac

6.2.1 Use CD-ROM in VM Window

Start OL7



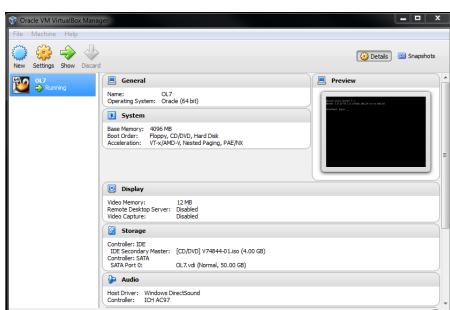
Here you can see Storage [CD/DVD] Empty.



Right click the CD-ROM icon at the right bottom of the VM window.

Choose a virtual CD/DVD disk file..., browse to the Oracle Linux iso file (V74844-01.iso).

Or, just select the V74844-01.iso file if it list on the menu.



Now you can see Storage [CD/DVD] V74844-01.iso (4.00 GB)

6.2.2 Mount CD-ROM in OL7

Login to OL7 using root, mount cdrom

```

1 Oracle Linux Server 7.1
2 Kernel 3.8.13-55.1.6.el7uek.x86_64 on an x86_64
3
4 Hint: Num Lock on
5
6 localhost login: root
7 Password:
8 Last login: Mon Nov  9 12:29:44 on ttym1
9 [root@localhost ~]# cd /media
10 [root@localhost media]# mkdir cdrom
11 [root@localhost media]# mount -t iso9660 /dev/cdrom /media/cdrom
12 mount: /dev/sr0 is write-protected, mounting read-only
13 [root@localhost media]#

```

6.2.3 Change yum repo to use local CD-ROM

```

1 [root@localhost ~]# cd /etc/yum.repos.d
2 [root@localhost yum.repos.d]# cp public-yum-ol7.repo ol7dvd.repo
3 [root@localhost yum.repos.d]# mv public-yum-ol7.repo
   ↳ public-yum-ol7.repo.original
4 [root@localhost yum.repos.d]# vi ol7dvd.repo

```

Modify yum repo:

Original:

```

1 [ol7_latest]
2 name=Oracle Linux $releasever Latest ($basearch)
3 baseurl=http://public-yum.oracle.com/repo/OracleLinux/OL7/latest/$basearch/
4 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
5 gpgcheck=1
6 enabled=1
7
8 [ol7_u0_base]
9 .....

```

Delete all sections, only leave the first one, here is the modified file:

```

1 [ol7_latest_cdrom]
2 name=Oracle Linux $releasever Latest ($basearch)
3 baseurl=file:///media/cdrom
4 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
5 gpgcheck=0
6 enabled=1

```

6.2.4 Prepare for Guest Additions

- **yum update**

```

1 [root@localhost ~]# yum update
2 ol7_latest
3 (1/2): ol7_latest/group_gz | 3.6 kB     00:00
4 (2/2): ol7_latest/group_gz | 134 kB    00:00
5 No packages marked for update
6 [root@localhost ~]#

```

- **yum install gcc**

```

1 [root@localhost ~]# yum install gcc
2 Package gcc-4.8.3-9.el7.x86_64 already installed and latest version
3 Nothing to do
4 [root@localhost ~]#

```

▪ **yum install kernel-devel**

```

1 [root@localhost ~]# yum install kernel-devel

```

▪ **yum install kernel-uek-devel**

```

1 [root@localhost ~]# yum install kernel-uek-devel

```

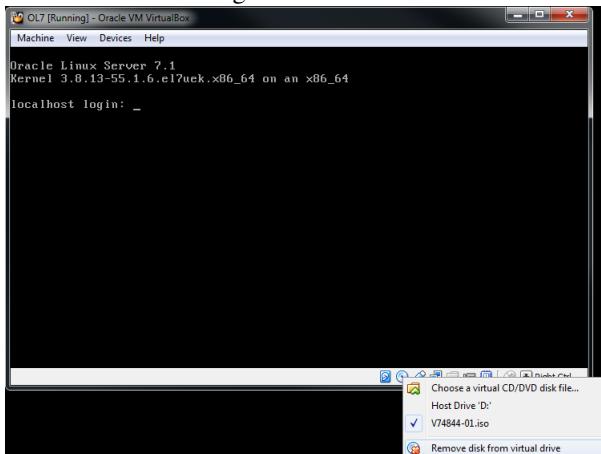
▪ **yum install bzip2**

```

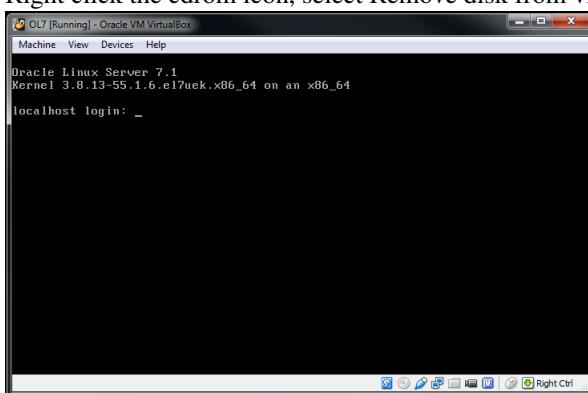
1 [root@localhost ~]# yum install bzip2

```

▪ **Remove cdrom from guest window.**

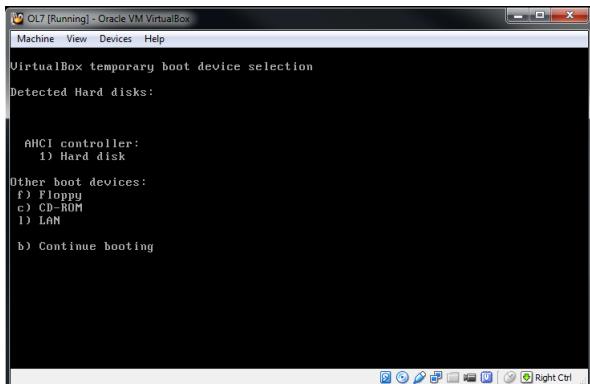


Right click the cdrom icon, select Remove disk from virtual drive.



You can see the cdrom icon is grey.

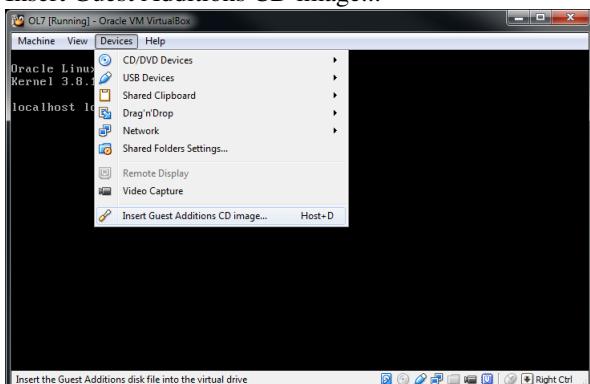
- Reboot your guest system in order to activate the updates.
- If you want to reboot your guest system without umount cdrom, and you want to boot from Hard disk, then during reboot, press F12



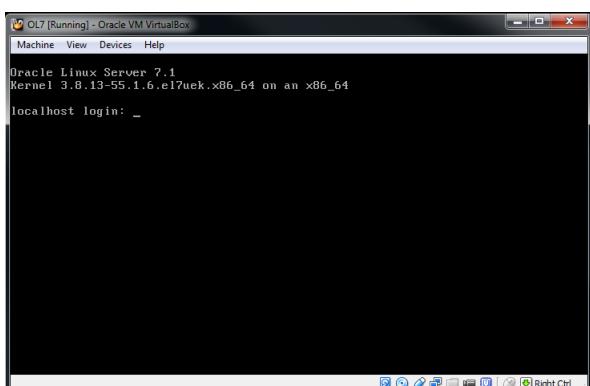
Select 1 to boot from Hard disk.

6.2.5 Install Guest Additions

- Insert Guest Additions CD image...



Go to Devices → Insert Guest Additions CD image...



You can see the cdrom icon shows Guest Additions CD inserted.

- mount cdrom

```
1 [root@localhost ~]# mount -t iso9660 /dev/cdrom /media/cdrom
2 mount: /dev/sr0 is write-protected, mounting read-only
3 [root@localhost ~]#
```

- install guest additions

```

1 [root@localhost ~]# cd /media/cdrom
2 [root@localhost cdrom]# ./VBoxLinuxAdditions.run

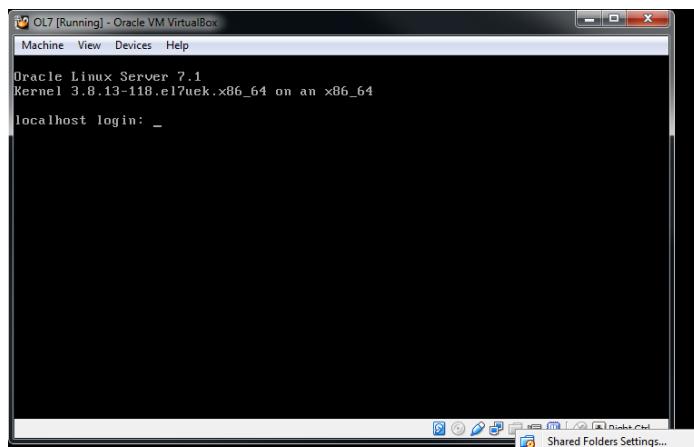
```

- Reboot your guest system.
- You can Remove disk from virtual drive by right click the cdrom icon on the bottom of the VM window.

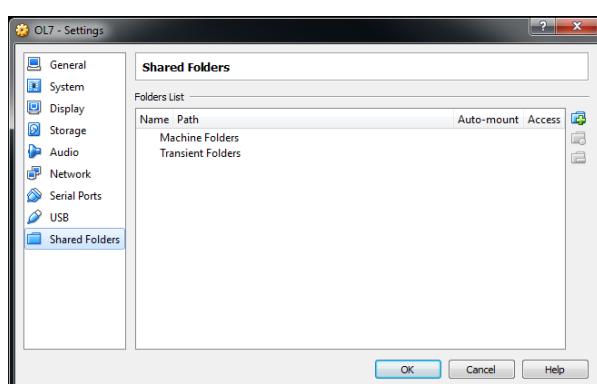
6.3 Shared folders

With the "shared folders" feature of VirtualBox, you can access files of your host system from within the guest system. This is similar how you would use network shares in Windows networks – except that shared folders do not need require networking, only the Guest Additions.

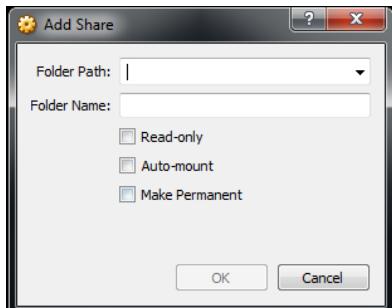
6.3.1 Shared folders - Windows



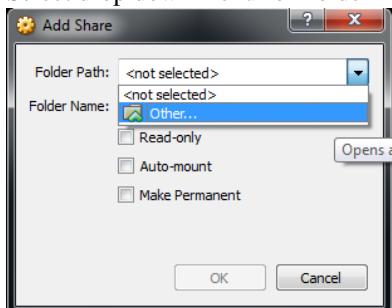
Right click guest window Shared folder icon on the bottom right corner. Select Shared Folders Settings...



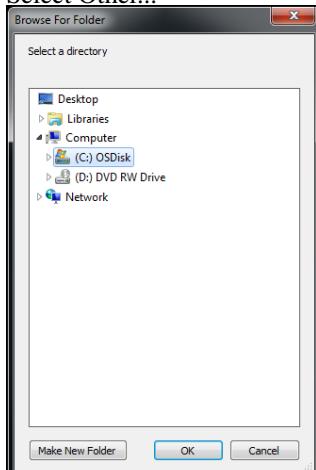
Click the + sign on the right.



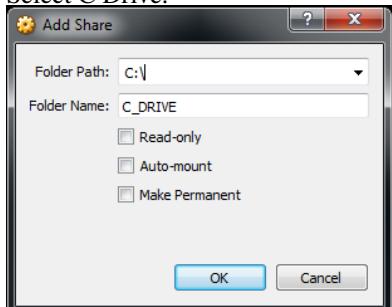
Select drop down menu for Folder Path.



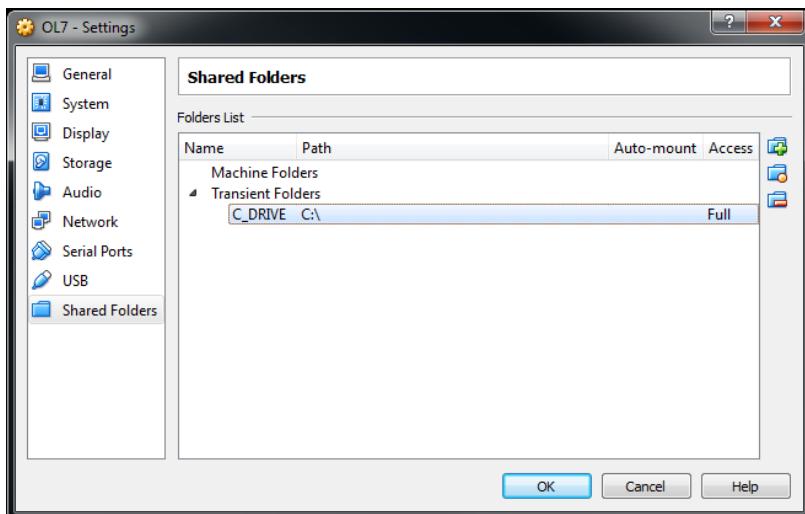
Select Other...



Select C Drive.



Click OK.



Click OK.

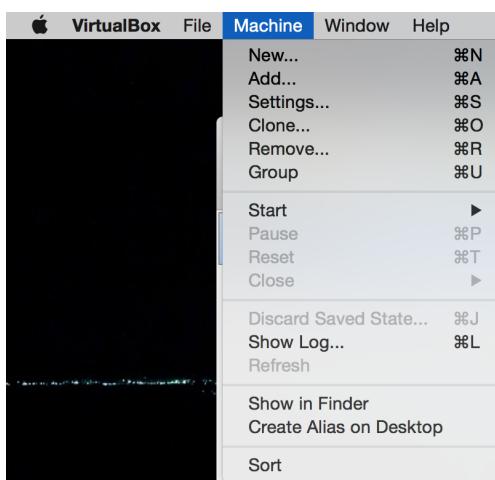
- mount C Drive.

```

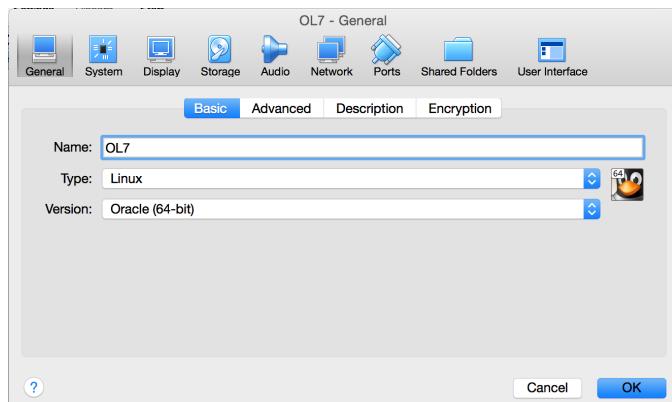
1 [root@localhost ~]# cd /media
2 [root@localhost media]# mkdir share
3 [root@localhost media]# mount -t vboxsf C_DRIVE /media/share
4 [root@localhost media]# cd share
5 [root@localhost share]# ls -al
6 total 14249044
7 lwxrwxrwx. 1 root root      0 Jul 14  2009 Documents and Settings
8 drwxrwxrwx. 1 root root    12288 Nov 10 10:07 ProgramData
9 dr-xr-xr-x. 1 root root     8192 Jun 11 09:08 Program Files
10 drwxrwxrwx. 1 root root    20480 Oct 22 10:01 Program Files (x86)
11 dr-xr-xr-x. 1 root root     4096 Apr 10  2015 Users
12 drwxrwxrwx. 1 root root    40960 Oct 30 14:43 Windows
13 ...
14 [root@localhost share]#

```

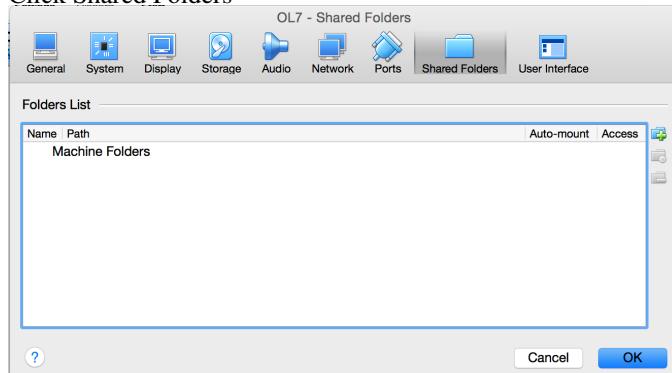
6.3.2 Shared folders - Mac



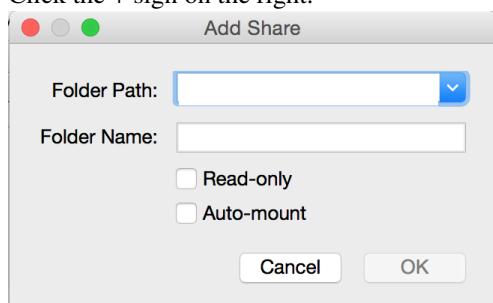
Click top menu Machine ->Settings...



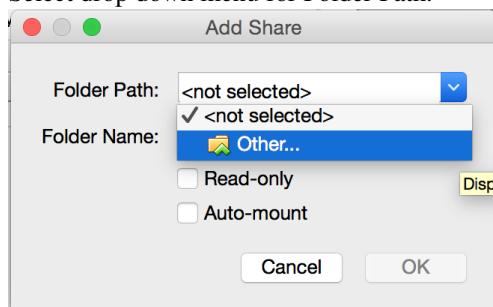
Click Shared Folders



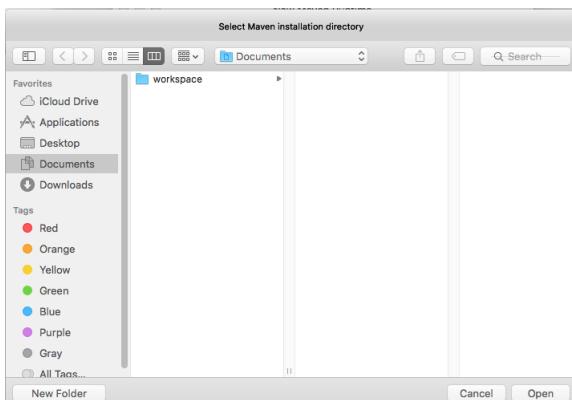
Click the + sign on the right.



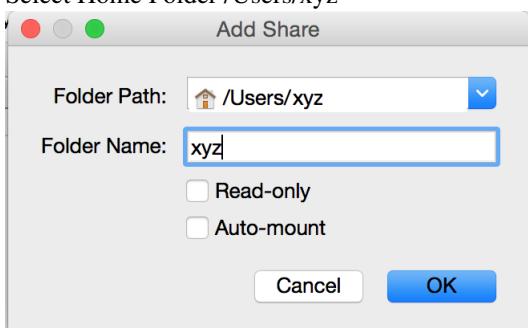
Select drop down menu for Folder Path.



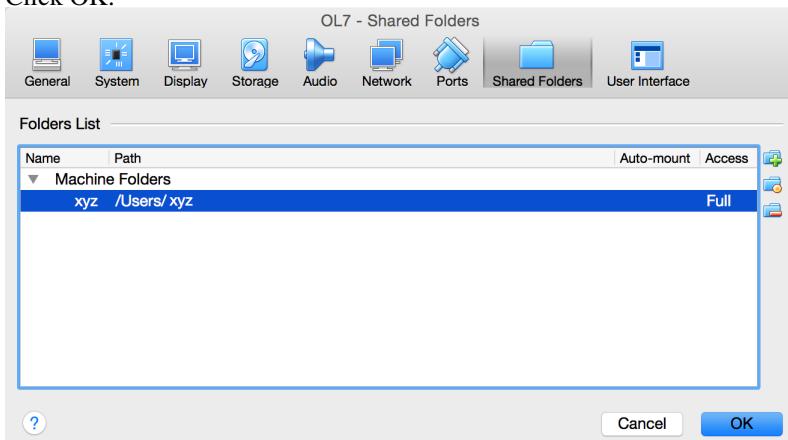
Select Other...



Select Home Folder /Users/xyz



Click OK.



Click OK.

- mount xyz.

```

1 [root@localhost ~]# cd /media
2 [root@localhost media]# mkdir share
3 [root@localhost media]# mount -t vboxsf xyz /media/share
4 [root@localhost media]# cd share
5 [root@localhost share]# ls -al
6 ...
7 [root@localhost share]#

```

6.4 Network

- These steps apply to both Windows and Mac
- Change back original yum repo

```

1 [root@localhost ~]# cd /etc/yum.repos.d
2 [root@localhost yum.repos.d]# mv public-yum-ol7.repo.original
   ↳ public-yum-ol7.repo

```

- Disable ol7dvd.repo

```

1 [root@localhost yum.repos.d]# vi ol7dvd.repo
2 [ol7_latest_cdrom]
3 name=Oracle Linux $releasever Latest ($basearch)
4 baseurl=file:///media/cdrom
5 gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
6 gpgcheck=0
7 enabled=0

```

- Setup proxy

```

1 [root@localhost ~]# export http_proxy=http://http.proxy.xyz.com:8000
2 [root@localhost ~]#

```

- Check network status

```

1 [root@localhost ~]# nmcli device status
2 DEVICE  TYPE      STATE      CONNECTION
3 enp0s3  ethernet  disconnected  --
4 lo      loopback  unmanaged   --
5
6 [root@localhost ~]#

```

- Enable ethernet

```

1 [root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-enp0s3
2 TYPE=Ethernet
3 BOOTPROTO=dhcp
4 DEFROUTE=yes
5 PEERDNS=yes
6 PEERROUTES=yes
7 IPV4_FAILURE_FATAL=no
8 IPV6INIT=yes
9 IPV6_AUTOCONF=yes
10 IPV6_DEFROUTE=yes
11 IPV6_PEERDNS=yes
12 IPV6_PEERROUTES=yes
13 IPV6_FAILURE_FATAL=no
14 NAME=enp0s3
15 UUID=dd481d7b-4872-4c14-8ab6-5b5fec5fac59
16 DEVICE=enp0s3
17 ONBOOT=no          ---> change to yes
18
19 [root@localhost ~]#

```

- Reboot

- Check network status

```

1 [root@localhost ~]# nmcli device status
2 DEVICE  TYPE      STATE      CONNECTION
3 enp0s3  ethernet  connected   enp0s3
4 lo      loopback  unmanaged   --
5
6 [root@localhost ~]#

```

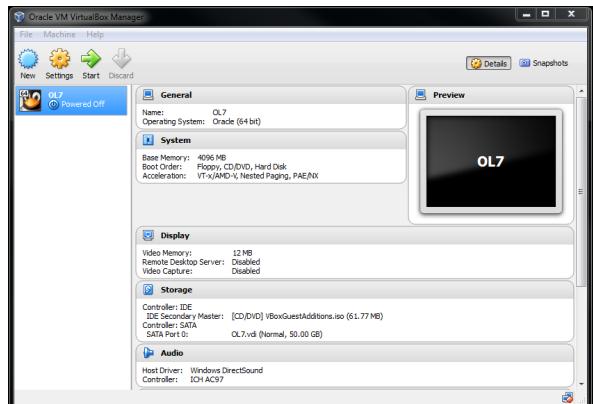
CHAPTER 7

CONNECT TO VIRTUAL MACHINE

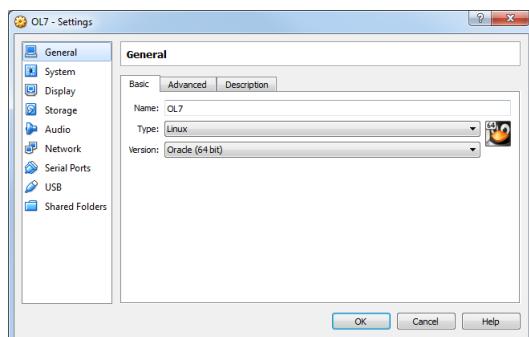
7.1 Connect to VM

7.1.1 VM Network Setting

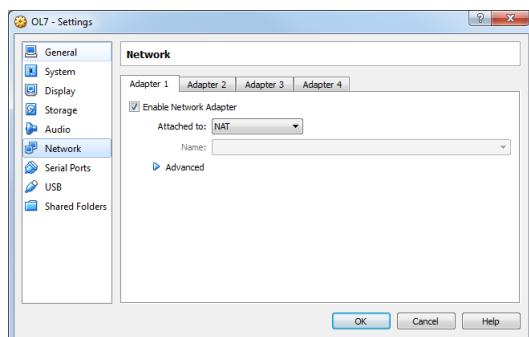
- These steps apply to both Windows and Mac
- VirtualBox will create a private network (10.0.2.x) which will be connected to your host network using NAT. (Unless configured otherwise.)
- This means that you cannot directly access any host of the private network from the host network. To do so, you need some port forwarding.
- Stop your VirtualBox machine



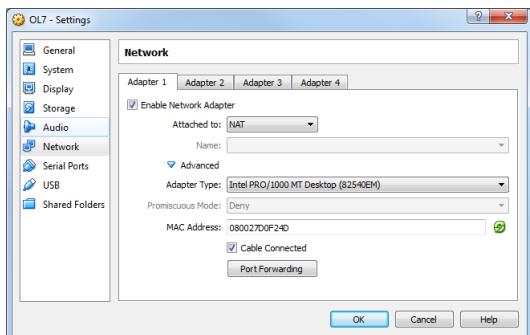
Click Settings



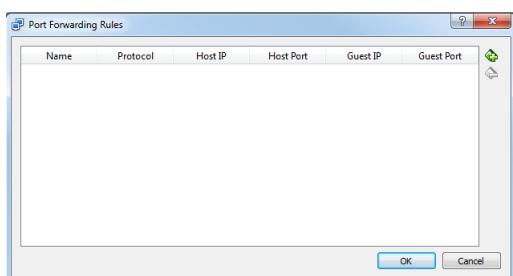
Click Network



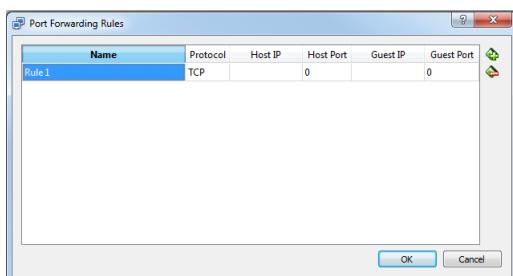
Expand Advanced



Click Port Forwarding



Click + sign to add a Port Forwarding Rule

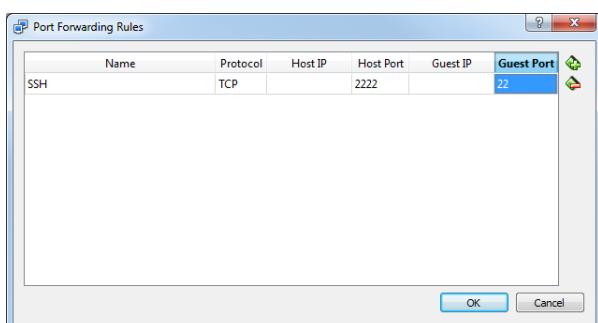


Name: SSH

Protocol: TCP

Host Port: 2222

Guest Port: 22



Click Ok, then click OK again

7.1.2 Connect to VM - Windows

7.1.2.1 PuTTY

- Download PuTTY from:
<http://www.putty.org/> <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
The latest release version (beta 0.65), for Windows on Intel x86, putty.exe
<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>
Put it in C:\opt\PuTTY\putty.exe
- Start Oracle Linux 7 VM

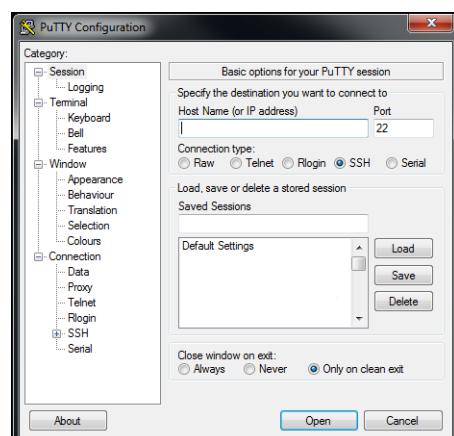


- Run PuTTY

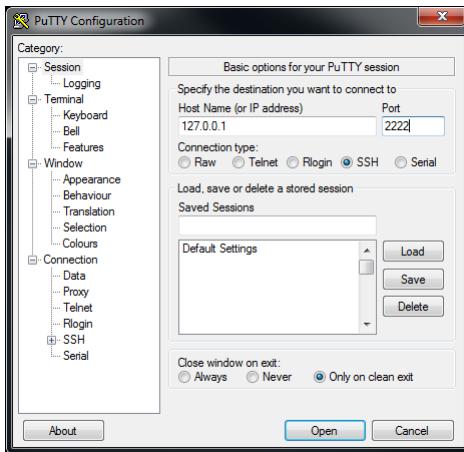
Double click C:\opt\PuTTY\putty.exe



Click Run



Let's login to guest linux.



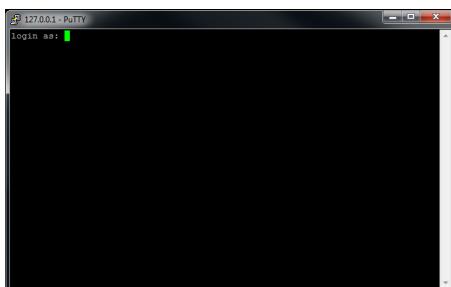
Host Name: 127.0.0.1

Port: 2222

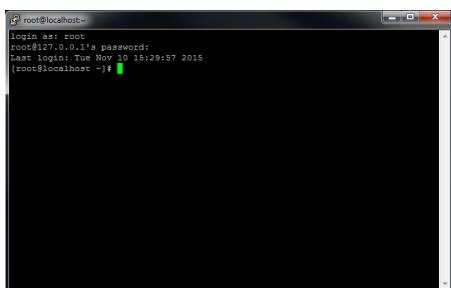
Click Open



Click Yes, you only need to do this one time.



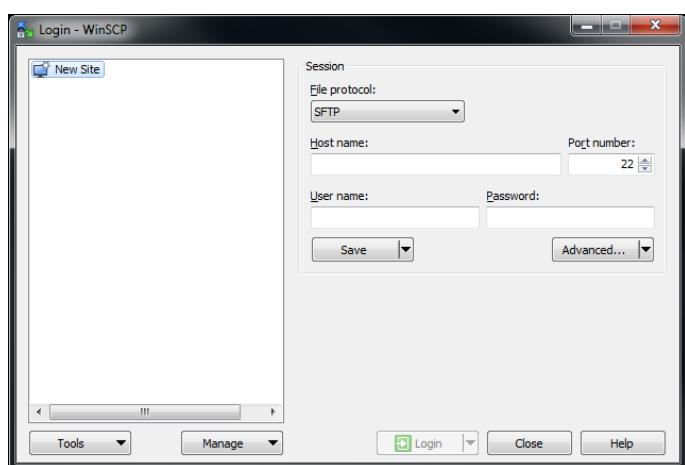
The Oracle Linux 7 login page shows up.



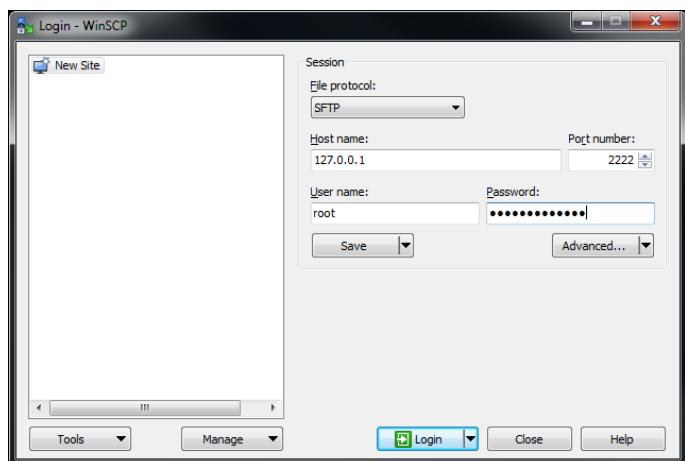
Login in as root.

7.1.2.2 WinSCP

- Download WinSCP from:
<https://winscp.net/eng/download.php>
 Portable executables (winscp576.zip)
<https://winscp.net/download/winscp576.zip>
- Extract WinSCP
 Extract winscp576.zip to C:\opt\WinSCP
- Start WinSCP

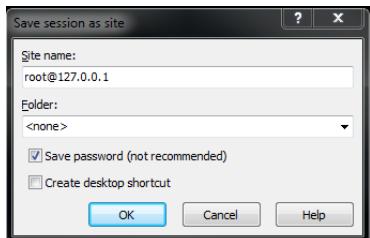


Let's connect to OL7

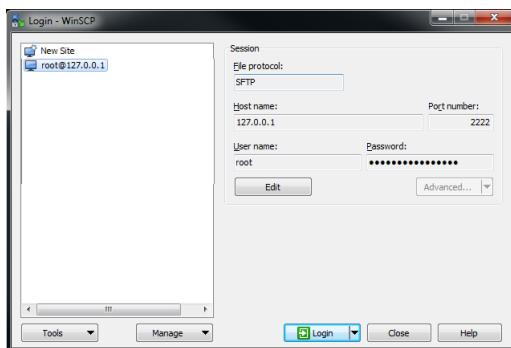


Host name: 127.0.0.1
 Port number: 2222
 User name: root
 Password: your password

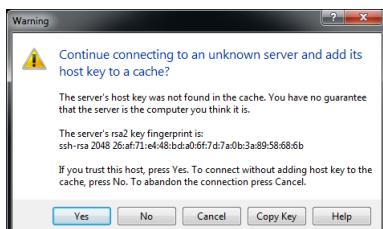
Click Save



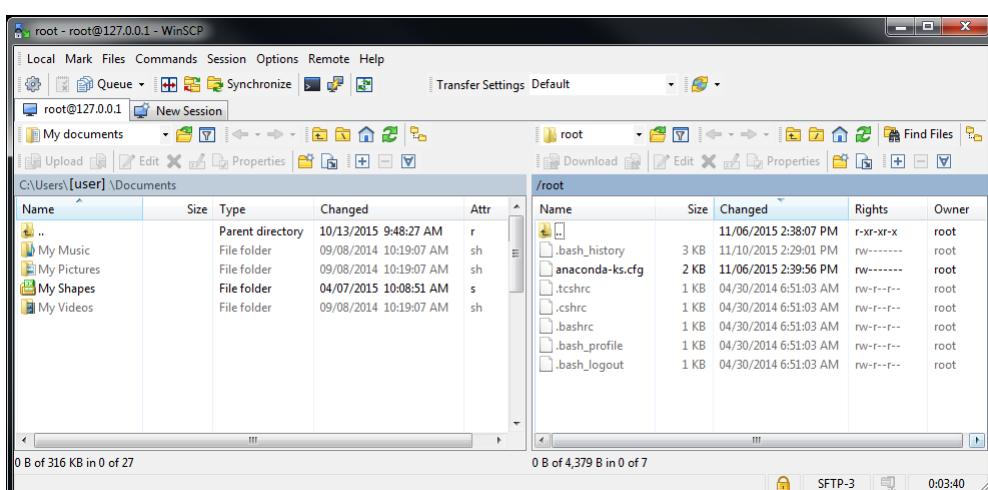
Check Save password (not recommend), then click OK



You can see root@127.0.0.1 on the left panel.
 Click Login



Click Yes, you only need to do this one time.



You are in. You can move files between your host and guest machines.

7.1.3 Connect to VM - Mac

7.1.3.1 SSH

- Using Terminal for SSH

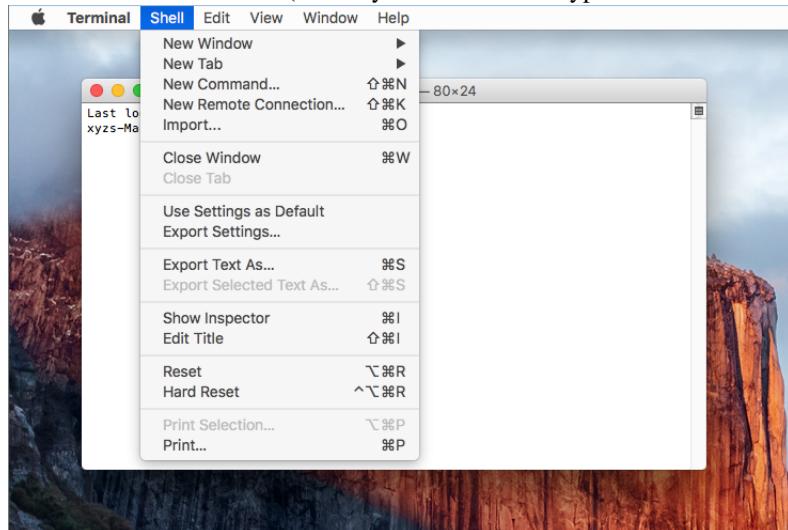
Open a Terminal window

```

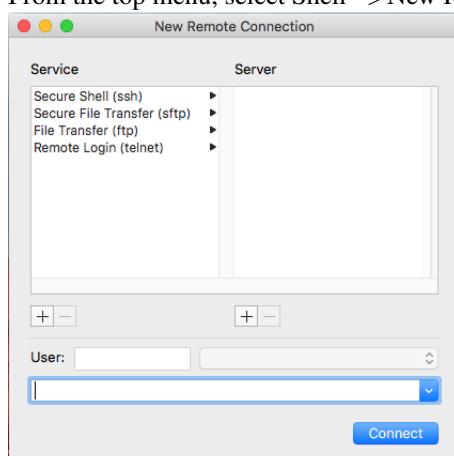
1 xyzs-Mac:~ xyz$ ssh root@localhost -p2222
2 The authenticity of host '[localhost]:2222 ([127.0.0.1]:2222)' can't be
   ↪ established.
3 RSA key fingerprint is 0d:0b:1f:2a:c5:09:3e:b6:c7:cc:6c:c3:f8:2e:24:1c.
4 Are you sure you want to continue connecting (yes/no)? yes
5 Warning: Permanently added '[localhost]:2222' (RSA) to the list of known hosts.
6 root@localhost's password:
7 Last login: Mon Jul 4 22:59:11 2016
8 [root@localhost ~]#

```

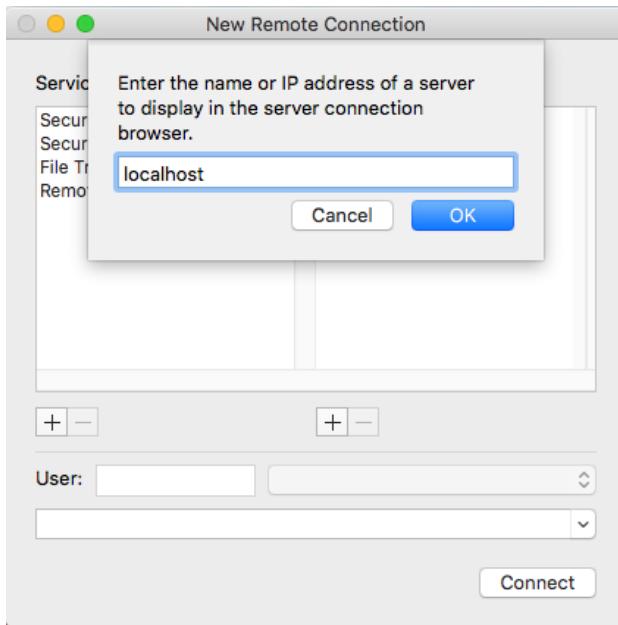
- Save a remote connection (so that you don't have to type in the command every time)



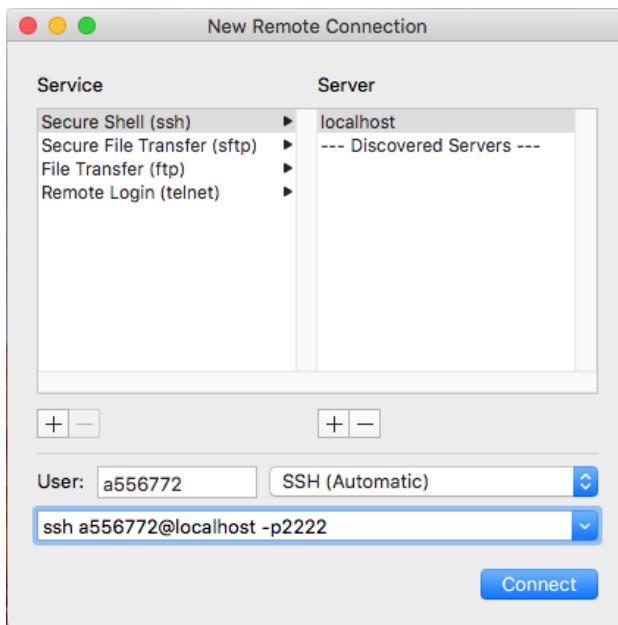
From the top menu, select Shell → New Remote Connection...



Under the Server column, click on the + icon to add a new connection



Enter the hostname when prompted by the dialog



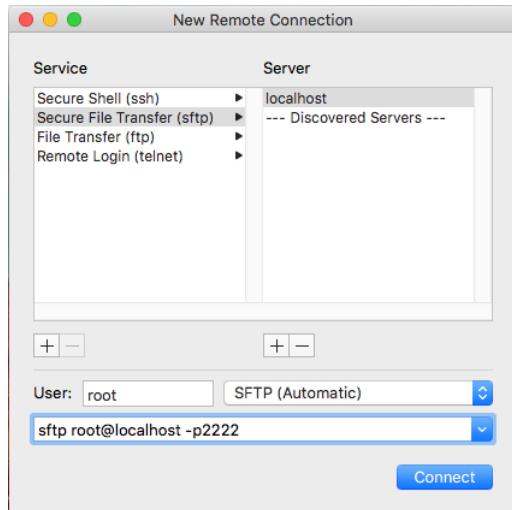
Enter the username in the User field, click localhost on the Server column
The command will show up in the bottom field, then add -p2222, click Connect

```

1 [root@localhost ~]#
2 [root@localhost ~]#
```

Type in password, then you are connected to VM

7.1.3.2 SFTP



Under the Service column, select Secure File Transfer (sftp), click localhost on the Server column

The command will show up in the bottom field, then add -p2222, click Connect

```

1 [root@localhost's password:
2 Connected to localhost.
3 sftp> help
4 Available commands:
5 bye                                Quit sftp
6 cd path                            Change remote directory to 'path'
7 chgrp grp path                     Change group of file 'path' to 'grp'
8 chmod mode path                   Change permissions of file 'path' to 'mode'
9 chown own path                    Change owner of file 'path' to 'own'
10 df [-hi] [path]                  Display statistics for current directory or
                                 filesystem containing 'path'
11
12 exit                               Quit sftp
13 get [-afPpRr] remote [local]      Download file
14 reget [-fFpRr] remote [local]     Resume download file
15 reput [-fFpRr] [local] remote    Resume upload file
16 help                               Display this help text
17 lcd path                           Change local directory to 'path'
18 lls [ls-options [path]]          Display local directory listing
19 lmkdir path                        Create local directory
20 ln [-s] oldpath newpath           Link remote file (-s for symlink)
21 lpwd                               Print local working directory
22 ls [-lafhlnRt] [path]            Display remote directory listing
23 lumask umask                      Set local umask to 'umask'
24 mkdir path                         Create remote directory
25 progress                           Toggle display of progress meter
26 put [-afPpRr] local [remote]      Upload file
27 pwd                                Display remote working directory
28 quit                               Quit sftp
29 rename oldpath newpath           Rename remote file
30 rm path                            Delete remote file
31 rmdir path                         Remove remote directory
32 symlink oldpath newpath          Symlink remote file
33 version                            Show SFTP version
34 !command                           Execute 'command' in local shell
35 !
36 ?                                 Escape to local shell
37 sftp>

```

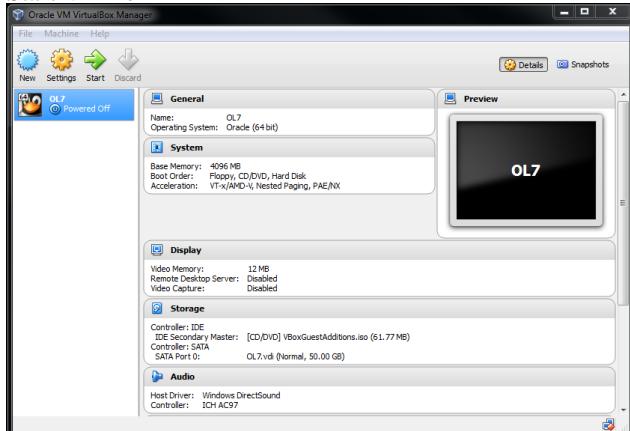
Then you can use get to copy files from VM, use put to copy files to VM.

7.2 Backup and Restore VM

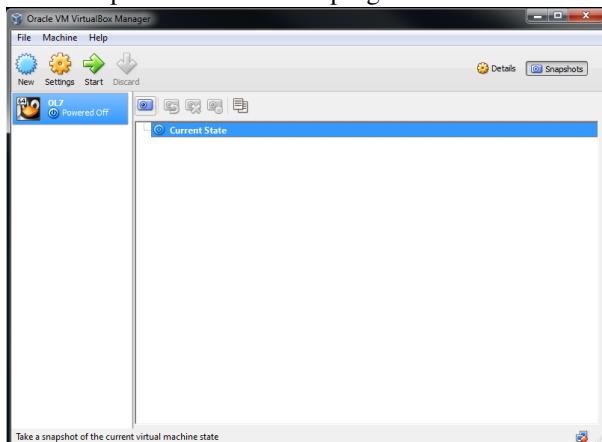
These steps apply to both Windows and Mac

7.2.1 Snapshots

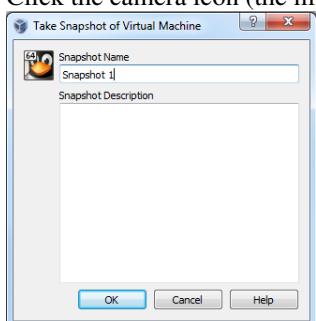
- Start VMBox



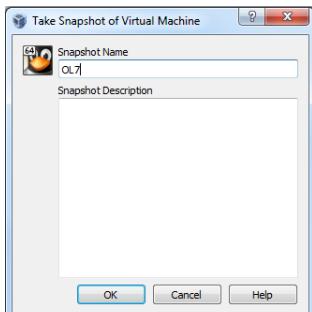
Click Snapshots icon on the top right corner.



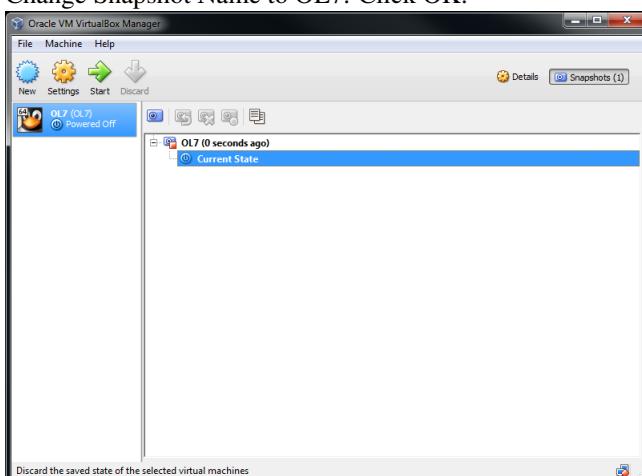
Click the camera icon (the first of 5 icons in the middle).



A new windows ask Snapshot Name.



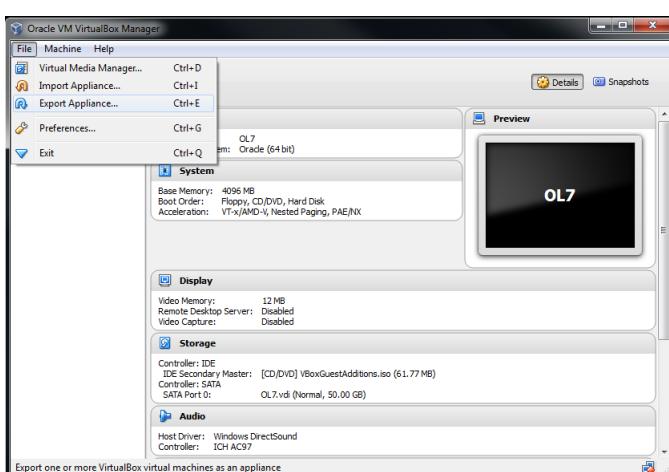
Change Snapshot Name to OL7. Click OK.



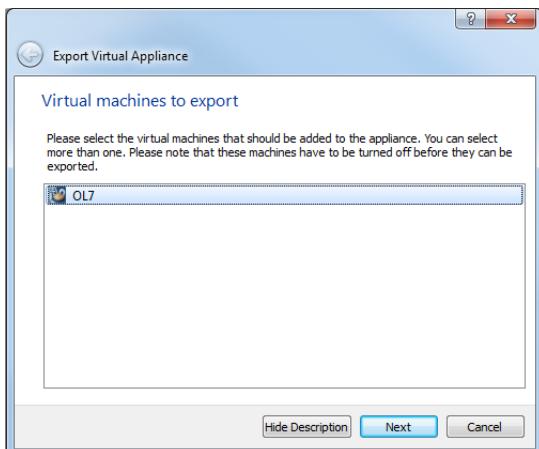
OL7 Snapshot saved.

7.2.2 Export Appliance

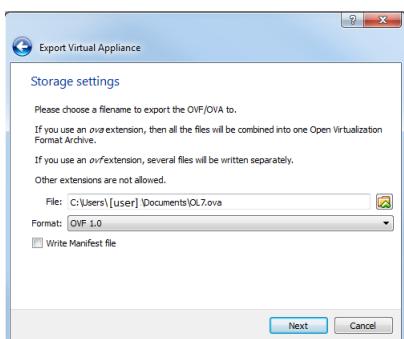
We can export an image of the current guest system, then import it later.



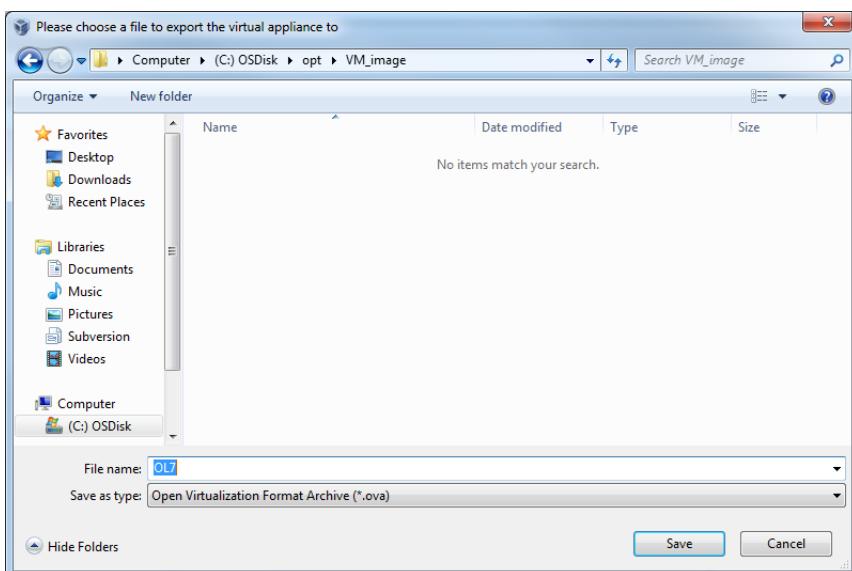
Select File → Export Appliance...



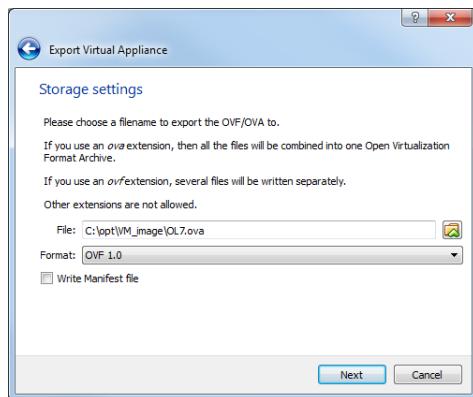
Click Next



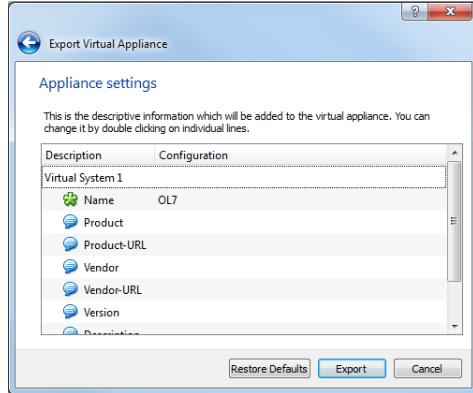
Click the folder icon on the right side to change destination.



Select your destination directory for the image. C:\opt\VM_image\, File name OL7



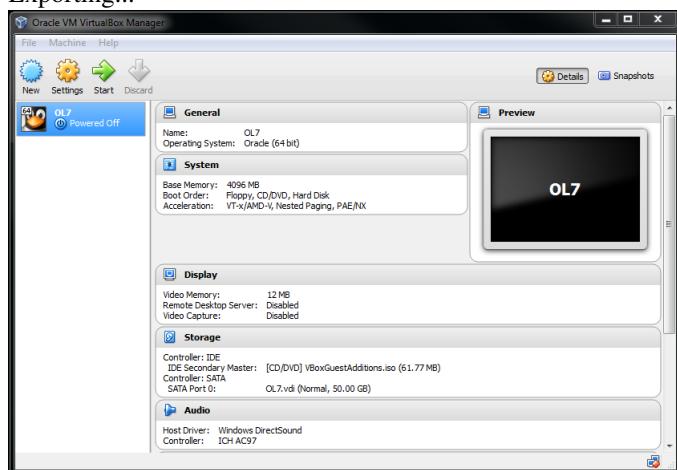
Click Next



Click Export

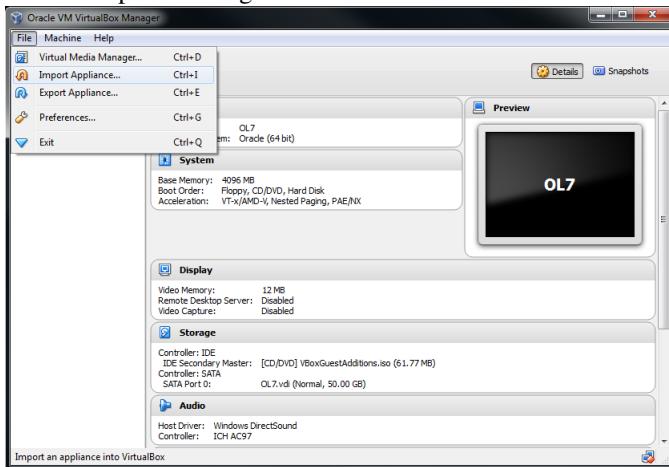


Exporting...

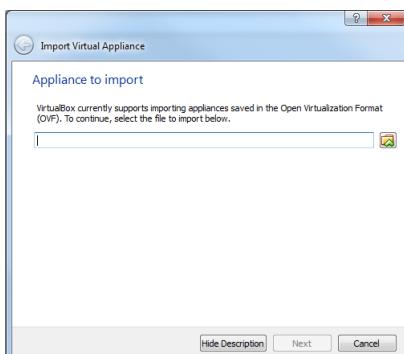


7.2.3 Import Appliance

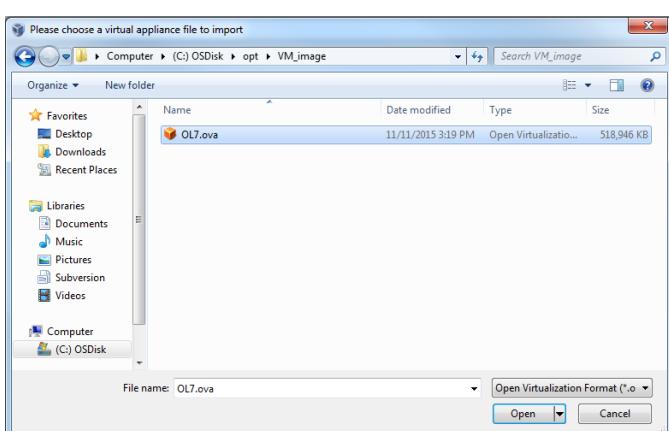
- We can import an image we saved earlier.



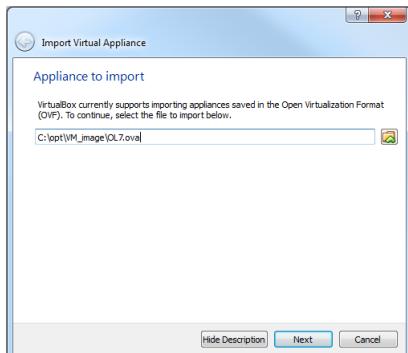
Select File ->Import Appliance...



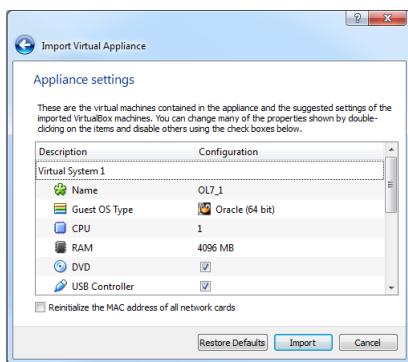
Click the folder icon on the right side to select image.



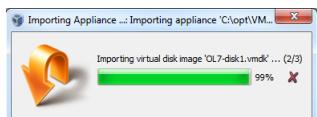
Browse to the saved image. Click Open



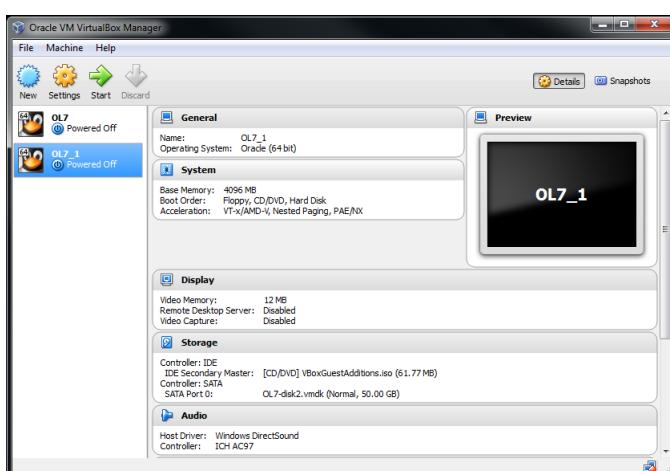
Click Next



Click Import



Importing...



Import completed. You can see there is another system call OL7_1 on the left panel.

CHAPTER 8

ORACLE ON VIRTUAL MACHINE

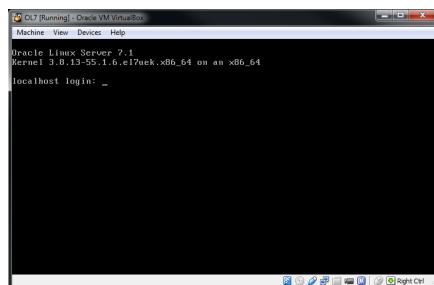
Oracle Database 12c introduces a new multitenant architecture that makes it easy to consolidate many databases quickly and manage them as a cloud service. Oracle Database 12c also includes in-memory data processing capabilities delivering breakthrough analytical performance. Additional database innovations deliver new levels of efficiency, performance, security, and availability. Oracle Database 12c comes in three editions to fit your business needs and budget: Enterprise Edition, Standard Edition, and Standard Edition One.

8.1 Download Oracle

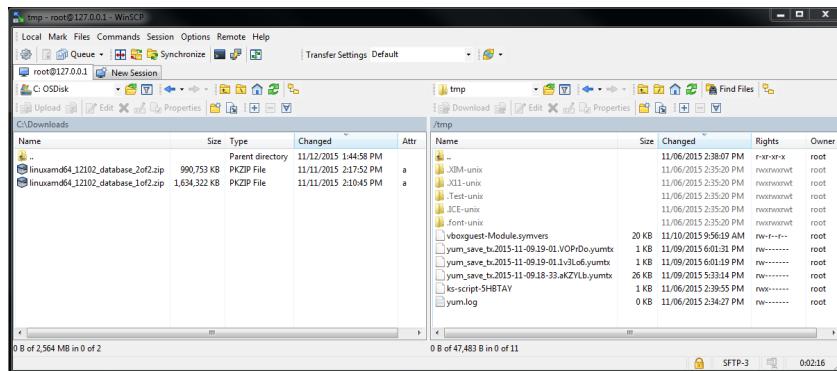
- These steps apply to both Windows and Mac
- Download Oracle Database 12c from:
<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html>
Oracle Database 12c Release 1
12.1.0.2.0
Enterprise Edition
Linux x86-64 (2.5GB)
linuxamd64_12102_database_1of2.zip
linuxamd64_12102_database_2of2.zip

8.2 Copy Oracle Database files to VMBox OL7

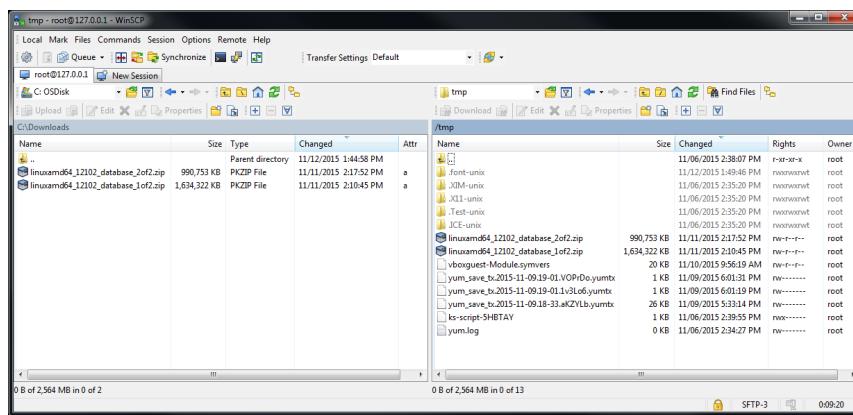
Start VMBox OL7



8.2.1 Using WinSCP to copy files - Windows



Drag and drop `linuxamd64_12102_database_1of2.zip` and `linuxamd64_12102_database_2of2.zip` from `C:\Downloads` to OL7 `/tmp`

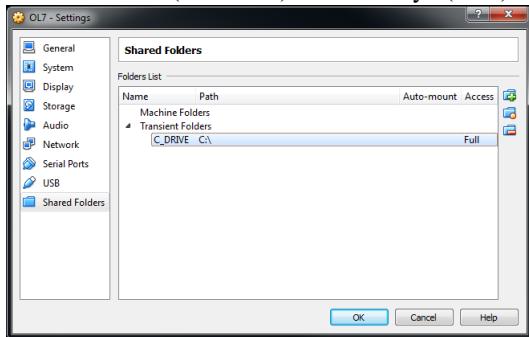


8.2.2 Copy files - Mac

Using Terminal → Shell → New Remote Connection... → Secure File Transfer (sftp) to copy files.

8.2.3 Using VMBox shared folder to copy files

- These steps apply to both Windows and Mac
- Add C DRIVE (Windows) or /Users/xyz (Mac) as a shared folder to VM



- Using PuTTY (Windows) or SSH (Mac) connect to VM OLT
- mount C Drive (Windows) or /Users/xyz (Mac)

```

1 [root@localhost ~]# cd /media
2 [root@localhost media]# mount -t vboxsf C_DRIVE /media/share
3 [root@localhost media]# cd share
4 [root@localhost share]# cd Downloads
5 [root@localhost Downloads]# cp linuxamd64_12102_database_1of2.zip /tmp
6 [root@localhost Downloads]# cp linuxamd64_12102_database_2of2.zip /tmp
7 [root@localhost Downloads]#

```

8.3 Network

- These steps apply to both Windows and Mac
- Proxy

```

1 [root@localhost ~]# vi .bash_profile
2 # .bash_profile
3
4 # Get the aliases and functions
5 if [ -f ~/.bashrc ]; then
6     . ~/.bashrc
7 fi
8
9 # User specific environment and startup programs
10
11 PATH=$PATH:$HOME/bin
12
13 export PATH
14 export http_proxy=http://http.proxy.xyz.com:8000
15
16 [root@localhost ~]# . .bash_profile
17 [root@localhost ~]#

```

- Hosts File
/etc/hosts

```

1 [root@localhost ~]# vi /etc/hosts
2 127.0.0.1 localhost.localdomain localhost4 localhost4.localdomain4
3 ::1 localhost.localdomain localhost6 localhost6.localdomain6
4
5 [root@localhost ~]#

```

/etc/hostname

```

1 [root@localhost ~]# vi /etc/hostname
2 localhost.localdomain
3
4 [root@localhost ~]#

```

/etc/sysconfig/network

```

1 [root@localhost ~]# vi /etc/sysconfig/network
2 # Created by anaconda
3 HOSTNAME=localhost.localdomain
4
5 [root@localhost ~]#

```

▪ Disable Firewall

Fedora 18 introduced firewalld as a replacement for the previous iptables service. Since RHEL7 and Oracle Linux 7 are based on Fedora 19, the switch from iptables service to firewalld is now part of the Enterprise Linux distributions. The firewall on Oracle Linux 7 system is enabled by default.

Check firewall status - you can see firewall active

```

1 [root@localhost ~]# systemctl status firewalld
2 firewalld.service - firewalld - dynamic firewall daemon
3   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled)
4     Active: active (running) since Thu 2015-11-12 12:29:22 EST; 4h 16min ago
5       Main PID: 510 (firewalld)
6         CGroup: /system.slice/firewalld.service
7             510 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid
8
9 Nov 12 12:29:22 localhost.localdomain systemd[1]: Started firewalld - dynamic
10   → firewall daemon.
11
12 [root@localhost ~]#

```

Stop firewall, but after reboot, firewall will automatically start

```

1 [root@localhost ~]# systemctl stop firewalld
2 [root@localhost ~]#

```

Check firewall status - you can see firewall inactive

```

1 [root@localhost ~]# systemctl status firewalld
2 firewalld.service - firewalld - dynamic firewall daemon
3   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled)
4     Active: inactive (dead)
5
6 Nov 12 12:29:20 localhost.localdomain systemd[1]: Starting firewalld - dynamic
7   → firewall daemon...
8 Nov 12 12:29:22 localhost.localdomain systemd[1]: Started firewalld - dynamic
9   → firewall daemon.
10 Nov 12 16:52:47 localhost.localdomain systemd[1]: Stopping firewalld - dynamic
11   → firewall daemon...
12 Nov 12 16:52:47 localhost.localdomain systemd[1]: Stopped firewalld - dynamic
13   → firewall daemon.
14
15 [root@localhost ~]#

```

In order to completely disable OL7 firewall, so it would not start after reboot, run:

```

1 [root@localhost ~]# systemctl disable firewalld
2 rm '/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service'
3 rm '/etc/systemd/system/basic.target.wants/firewalld.service'
4
5 [root@localhost ~]#

```

8.4 Install Zip and Unzip

- These steps apply to both Windows and Mac
- Zip

```
1 [root@localhost ~]# yum install zip
```

- Unzip

```
1 [root@localhost ~]# yum install unzip
```

8.5 Unpack Files

- These steps apply to both Windows and Mac

```
1 [root@localhost ~]# cd /tmp
2 [root@localhost tmp]# unzip linuxamd64_12102_database_1of2.zip
3 [root@localhost tmp]# unzip linuxamd64_12102_database_2of2.zip
4 [root@localhost tmp]#
```

All files are unzipped to /tmp/database

8.6 Oracle Installation Prerequisites

These steps apply to both Windows and Mac

8.6.1 Automatic Setup

- If you plan to use the "oracle-rdbms-server-12cR1-preinstall" package to perform all your prerequisite setup, issue the following command, all necessary prerequisites will be performed automatically.

```
1 [root@localhost ~]# yum install oracle-rdbms-server-12cR1-preinstall -y
```

- group oinstall, dba and user oracle are added

You can see this automatic setup created oinstall, dba group and user oracle

```
1 [root@localhost ~]# cat /etc/group
2 root:x:0:
3 bin:x:1:
4 daemon:x:2:
5 sys:x:3:
6 adm:x:4:
7 tty:x:5:
8 disk:x:6:
9 lp:x:7:
10 mem:x:8:
11 kmem:x:9:
12 wheel:x:10:
13 cdrom:x:11:
14 mail:x:12:postfix
15 man:x:15:
16 dialout:x:18:
17 floppy:x:19:
18 games:x:20:
19 tape:x:30:
```

```

20 video:x:39:
21 ftp:x:50:
22 lock:x:54:
23 audio:x:63:
24 nobody:x:99:
25 users:x:100:
26 utmp:x:22:
27 utempter:x:35:
28 avahi-autoipd:x:170:
29 ssh_keys:x:999:
30 systemd-journal:x:190:
31 dbus:x:81:
32 polkitd:x:998:
33 tss:x:59:
34 dip:x:40:
35 postdrop:x:90:
36 postfix:x:89:
37 sshd:x:74:
38 vboxsf:x:997:
39 rpc:x:32:
40 rpcuser:x:29:
41 nfsnobody:x:65534:
42 oinstall:x:54321:
43 dba:x:54322:oracle
44 [root@localhost ~]#

```

```

1 [root@localhost ~]# cat /etc/passwd
2 oracle:x:54321:54321::/home/oracle:/bin/bash
3 [root@localhost ~]#

```

- It is probably worth doing a full update as well, but this is not strictly speaking necessary.

```

1 [root@localhost ~]# yum update -y

```

8.6.2 Manual Setup

- If you have not used the "oracle-rdbms-server-12cR1-preinstall" package to perform all prerequisites, you will need to manually perform the following setup tasks.
- Add or amend the following lines in the "/etc/sysctl.conf" file.

```

1 fs.file-max = 6815744
2 kernel.sem = 250 32000 100 128
3 kernel.shmmni = 4096
4 kernel.shmall = 1073741824
5 kernel.shmmax = 4398046511104
6 net.core.rmem_default = 262144
7 net.core.rmem_max = 4194304
8 net.core.wmem_default = 262144
9 net.core.wmem_max = 1048576
10 fs.aio-max-nr = 1048576
11 net.ipv4.ip_local_port_range = 9000 65500

```

```

1 [root@localhost ~]# cat /etc/sysctl.conf
2 # System default settings live in /usr/lib/sysctl.d/00-system.conf.
3 # To override those settings, enter new settings here, or in an
4 #   /etc/sysctl.d/<name>.conf file
5 #
6 # For more information, see sysctl.conf(5) and sysctl.d(5).
7 #
8 # oracle-rdbms-server-12cR1-preinstall setting for fs.file-max is 6815744
9 fs.file-max = 6815744

```

```

10 # oracle-rdbms-server-12cR1-preinstall setting for kernel.sem is '250 32000 100
   ↳ 128'
11 kernel.sem = 250 32000 100 128
12
13 # oracle-rdbms-server-12cR1-preinstall setting for kernel.shmmni is 4096
14 kernel.shmmni = 4096
15
16 # oracle-rdbms-server-12cR1-preinstall setting for kernel.shmall is 1073741824
   ↳ on x86_64
17 kernel.shmall = 1073741824
18
19 # oracle-rdbms-server-12cR1-preinstall setting for kernel.shmmax is
   ↳ 4398046511104 on x86_64
20 kernel.shmmax = 4398046511104
21
22 # oracle-rdbms-server-12cR1-preinstall setting for kernel.panic_on_oops is 1 per
   ↳ Orabug 19212317
23 kernel.panic_on_oops = 1
24
25 # oracle-rdbms-server-12cR1-preinstall setting for net.core.rmem_default is
   ↳ 262144
26 net.core.rmem_default = 262144
27
28 # oracle-rdbms-server-12cR1-preinstall setting for net.core.rmem_max is 4194304
29 net.core.rmem_max = 4194304
30
31 # oracle-rdbms-server-12cR1-preinstall setting for net.core.wmem_default is
   ↳ 262144
32 net.core.wmem_default = 262144
33
34 # oracle-rdbms-server-12cR1-preinstall setting for net.core.wmem_max is 1048576
35 net.core.wmem_max = 1048576
36
37 # oracle-rdbms-server-12cR1-preinstall setting for net.ipv4.conf.all.rp_filter
   ↳ is 2
38 net.ipv4.conf.all.rp_filter = 2
39
40 # oracle-rdbms-server-12cR1-preinstall setting for
   ↳ net.ipv4.conf.default.rp_filter is 2
41 net.ipv4.conf.default.rp_filter = 2
42
43 # oracle-rdbms-server-12cR1-preinstall setting for fs.aio-max-nr is 1048576
44 fs.aio-max-nr = 1048576
45
46 # oracle-rdbms-server-12cR1-preinstall setting for net.ipv4.ip_local_port_range
   ↳ is 9000 65500
47 net.ipv4.ip_local_port_range = 9000 65500
48
49 [root@localhost ~]#

```

- Run the following command to change the current kernel parameters.

```
1 [root@localhost ~]# /sbin/sysctl -p
```

- Add the following lines to the "/etc/security/limits.conf" file.

```
1 oracle soft  nofile  1024
2 oracle hard  nofile  65536
3 oracle soft  nproc   2047
4 oracle hard  nproc   16384
5 oracle soft  stack   10240
6 oracle hard  stack   32768
```

Looks like the automatic setup did not add to "/etc/security/limits.conf" file.

```
1 [root@localhost ~]# cat /etc/security/limits.conf
2 # /etc/security/limits.conf
3 #
4 #This file sets the resource limits for the users logged in via PAM.
```

```

5  #It does not affect resource limits of the system services.
6  #
7  #Also note that configuration files in /etc/security/limits.d directory,
8  #which are read in alphabetical order, override the settings in this
9  #file in case the domain is the same or more specific.
10 #That means for example that setting a limit for wildcard domain here
11 #can be overridden with a wildcard setting in a config file in the
12 #subdirectory, but a user specific setting here can be overridden only
13 #with a user specific setting in the subdirectory.
14 #
15 #Each line describes a limit for a user in the form:
16 #
17 #<domain>      <type>  <item>  <value>
18 #
19 #Where:
20 #<domain> can be:
21 #      - a user name
22 #      - a group name, with @group syntax
23 #      - the wildcard *, for default entry
24 #      - the wildcard %, can be also used with %group syntax,
25 #            for maxlogin limit
26 #
27 #<type> can have the two values:
28 #      - "soft" for enforcing the soft limits
29 #      - "hard" for enforcing hard limits
30 #
31 #<item> can be one of the following:
32 #      - core - limits the core file size (KB)
33 #      - data - max data size (KB)
34 #      - fsize - maximum filesize (KB)
35 #      - memlock - max locked-in-memory address space (KB)
36 #      - nofile - max number of open file descriptors
37 #      - rss - max resident set size (KB)
38 #      - stack - max stack size (KB)
39 #      - cpu - max CPU time (MIN)
40 #      - nproc - max number of processes
41 #      - as - address space limit (KB)
42 #      - maxlogins - max number of logins for this user
43 #      - maxsyslogins - max number of logins on the system
44 #      - priority - the priority to run user process with
45 #      - locks - max number of file locks the user can hold
46 #      - sigpending - max number of pending signals
47 #      - msgqueue - max memory used by POSIX message queues (bytes)
48 #      - nice - max nice priority allowed to raise to values: [-20, 19]
49 #      - rtprio - max realtime priority
50 #
51 #<domain>      <type>  <item>  <value>
52 #
53
54 #*          soft    core      0
55 #*          hard    rss       10000
56 #@student   hard    nproc     20
57 #@faculty   soft    nproc     20
58 #@faculty   hard    nproc     50
59 #ftp        hard    nproc     0
60 #@student   -       maxlogins 4
61
62 # End of file
63 [root@localhost ~]#

```

- The following packages are listed as required, including the 32-bit version of some of the packages. Many of the packages should be installed already.

```

1  yum install binutils -y
2  yum install compat-libstdc++-33 -y
3  yum install compat-libstdc++-33.i686 -y
4  yum install gcc -y
5  yum install gcc-c++ -y
6  yum install glibc -y
7  yum install glibc.i686 -y
8  yum install glibc-devel -y

```

```

9  yum install glibc-devel.i686 -y
10 yum install ksh -y
11 yum install libgcc -y
12 yum install libgcc.i686 -y
13 yum install libstdc++ -y
14 yum install libstdc++.i686 -y
15 yum install libstdc++-devel -y
16 yum install libstdc++-devel.i686 -y
17 yum install libaio -y
18 yum install libaio.i686 -y
19 yum install libaio-devel -y
20 yum install libaio-devel.i686 -y
21 yum install libXext -y
22 yum install libXext.i686 -y
23 yum install libXtst -y
24 yum install libXtst.i686 -y
25 yum install libX11 -y
26 yum install libX11.i686 -y
27 yum install libXau -y
28 yum install libXau.i686 -y
29 yum install libxcb -y
30 yum install libxcb.i686 -y
31 yum install libXi -y
32 yum install libXi.i686 -y
33 yum install make -y
34 yum install sysstat -y
35 yum install unixODBC -y
36 yum install unixODBC-devel -y
37 yum install zlib-devel -y
38 yum install zlib-devel.i686 -y

```

- Create the new groups and users, uncomment the extra groups you require.

```

1 groupadd -g 54321 oinstall
2 groupadd -g 54322 dba
3 groupadd -g 54323 oper
4 #groupadd -g 54324 backupdba
5 #groupadd -g 54325 dgdba
6 #groupadd -g 54326 kmdba
7 #groupadd -g 54327 asmdba
8 #groupadd -g 54328 asmoper
9 #groupadd -g 54329 asmadmin
10
11 useradd -u 54321 -g oinstall -G dba,oper oracle

```

Looks like the automatic setup did not add group oper.

8.7 Additional Setup

The following steps must be performed, whether you did the manual or automatic setup. These steps apply to both Windows and Mac

- Set the password for the "oracle" user.

```

1 [root@localhost ~]# passwd oracle
2 Changing password for user oracle.
3 New password:
4 Retype new password:
5 passwd: all authentication tokens updated successfully.
6 [root@localhost ~]#

```

- Set secure Linux to permissive by editing the "/etc/selinux/config" file, making sure the SELINUX flag is set as follows.

```

1 [root@localhost ~]# cat /etc/selinux/config
2
3 # This file controls the state of SELinux on the system.
4 # SELINUX= can take one of these three values:
5 #       enforcing - SELinux security policy is enforced.
6 #       permissive - SELinux prints warnings instead of enforcing.
7 #       disabled - No SELinux policy is loaded.
8 SELINUX=permissive
9 # SELINUXTYPE= can take one of three two values:
10 #       targeted - Targeted processes are protected,
11 #       minimum - Modification of targeted policy. Only selected processes are
12 #                 protected.
13 #       mls - Multi Level Security protection.
14 SELINUXTYPE=targeted
15
16 [root@localhost ~]#

```

- Once the change is complete, restart the server or run the following command.

```

1 [root@localhost ~]# setenforce Permissive
2 [root@localhost ~]#

```

- If you have the Linux firewall enabled, you will need to disable or configure it. To disable it, do the following.

```

1 [root@localhost ~]# systemctl stop firewalld
2 [root@localhost ~]# systemctl disable firewalld

```

- Create the directories in which the Oracle software will be installed.

```

1 [root@localhost ~]# mkdir -p /u01/app/oracle/oraInventory
2 [root@localhost ~]# mkdir -p /u01/app/oracle/product/12.1.0.2/db_1
3 [root@localhost ~]# chown -R oracle:oinstall /u01
4 [root@localhost ~]# chmod -R 775 /u01

```

- Login into the "oracle" user. Or sudo to "oracle" user.

```

1 [root@localhost ~]# su - oracle
2 [oracle@localhost ~]$ 

```

- Add the following lines (start from # Oracle Settings) at the end of the "/home/oracle/.bash_profile" file.

```

1 [oracle@localhost ~]$ vi .bash_profile
2 # .bash_profile
3
4 # Get the aliases and functions
5 if [ -f ~/.bashrc ]; then
6     . ~/.bashrc
7 fi
8
9 # User specific environment and startup programs
10
11 PATH=$PATH:$HOME/.local/bin:$HOME/bin
12
13 export PATH
14
15 # Oracle Settings
16 export http_proxy=http://http.proxy.xyz.com:8000
17 export TMP=/tmp
18 export TMPDIR=$TMP
19
20 export ORACLE_HOSTNAME=localhost.localdomain
21 export ORACLE_UNQNAME=orcl
22 export ORACLE_BASE=/u01/app/oracle
23 export ORACLE_HOME=$ORACLE_BASE/product/12.1.0.2/db_1

```

```

24 export ORACLE_SID=orcl
25
26 export PATH=/usr/sbin:$PATH
27 export PATH=$ORACLE_HOME/bin:$PATH
28
29 export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib
30 export CLASSPATH=$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
31
32 if [ $USER = "oracle" ]; then
33     if [ $SHLL = "/bin/ksh" ]; then
34         ulimit -p 16384
35         ulimit -n 65536
36     else
37         ulimit -u 16384 -n 65536
38     fi
39 fi
40
41 [oracle@localhost ~]$

```

- .bash_profile or source .bash_profile, so the changes in .bash_profile effects immediately.

```

1 [oracle@localhost ~]$ . .bash_profile
2 [oracle@localhost ~]$

```

```

1 [oracle@localhost ~]$ source .bash_profile
2 [oracle@localhost ~]$

```

8.8 Installation

These steps apply to both Windows and Mac

8.8.1 Inventory File

- Create oraInst.loc file in /etc

```

1 [root@localhost ~]# vi /etc/oraInst.loc
2 # Oracle Installer Location File Location
3
4 inventory_loc=/u01/app/oracle/oraInventory
5 inst_group=oinstall
6
7 [root@localhost ~]#

```

8.8.2 Response Files

- Modify response files db_install.rsp and dbca.rsp files at /tmp/database/response.
- db_install.rsp

```

1 [root@localhost ~]# vi /tmp/database/response/db_install.rsp
2 oracle.install.responseFileVersion=/oracle/install/rspfmt_dbinstall_response_schema_v12.1.0
3 oracle.install.option=INSTALL_DB_SWONLY
4 ORACLE_HOSTNAME=
5 UNIX_GROUP_NAME=oinstall
6 INVENTORY_LOCATION=orcl
7 SELECTED_LANGUAGES=en
8 ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/db_1
9 ORACLE_BASE=/u01/app/oracle
10 oracle.install.db.InstallEdition=EE
11 oracle.install.db.DBA_GROUP=dba

```

```

12 oracle.install.db.OPER_GROUP=dba
13 oracle.install.db.BACKUPDBA_GROUP=dba
14 oracle.install.db.DGDBA_GROUP=dba
15 oracle.install.db.KMDBA_GROUP=dba
16 oracle.install.db.config.starterdb.type=GENERAL_PURPOSE
17 oracle.install.db.config.starterdb.characterSet=AL32UTF8
18 oracle.install.db.config.starterdb.memoryOption=true
19 oracle.install.db.config.starterdb.installExampleSchemas=false
20 oracle.install.db.config.starterdb.managementOption=DEFAULT
21 oracle.install.db.config.starterdb.storageType=FILE_SYSTEM_STORAGE
22 DECLINE_SECURITY_UPDATES=true
23
24 [root@localhost ~]#

```

■ dbca.rsp

```

1 [root@localhost ~]# vi /tmp/database/response/dbca.rsp
2 RESPONSEFILE_VERSION = "12.1.0"
3 OPERATION_TYPE = "createDatabase"
4 GDBNAME = "orcl"
5 SID = "orcl"
6 TEMPLATENAME = "General_Purpose.dbc"
7 SYSPASSWORD = "password"
8 SYSTEMPASSWORD = "password"
9 SOURCEDB = "myhost:1521:orcl"
10 SYSDBAUSERNAME = "system"
11 TEMPLATENAME = "My Copy TEMPLATE"
12 SOURCEDB = "orcl"
13 TEMPLATENAME = "My Clone TEMPLATE"
14 SOURCEDB = "orcl"
15 SYSDBAUSERNAME = "sys"
16 SYSDBAPASSWORD = "password"
17 TEMPLATENAME = "New Database"
18 GDBNAME = "orcl"
19 SOURCEDB = "orcl"
20 SYSDBAUSERNAME = "sys"
21 SYSDBAPASSWORD = "password"
22 DB_UNIQUE_NAME = "orcl"
23 INSTANCENAME = "orcl"
24 NODENAME=
25 SYSDBAUSERNAME = "sys"
26 SYSDBAPASSWORD = "password"
27 DB_UNIQUE_NAME = "orcl"
28 INSTANCENAME = "orcl"
29 SYSDBAUSERNAME = "sys"
30 SYSDBAPASSWORD = "password"
31 SOURCEDB = "orcl"
32 PDBNAME = "PDB1"
33 SOURCEDB = "orcl"
34 PDBNAME = "PDB1"
35 ARCHIVETYPE = "TAR"
36 SOURCEDB = "orcl"
37 PDBNAME = "PDB1"
38 SOURCEDB = "orcl"
39 PDBNAME = "PDB1"
40
41 [root@localhost ~]#

```

8.8.3 Install Oracle

- Login into the "oracle" user.
- Install Oracle - Silent Install with db_install.rsp response file.

```

1 [oracle@localhost ~]$ cd /tmp/database
2 [oracle@localhost database]$ ./runInstaller -silent -waitForCompletion
   ↳ -ignoreSysPrereqs -ignorePrereq -responseFile
   ↳ /tmp/database/response/db_install.rsp
3 [oracle@localhost database]$

```

```

1 Starting Oracle Universal Installer...
2
3 Checking Temp space: must be greater than 500 MB. Actual 8965 MB Passed
4 Checking swap space: must be greater than 150 MB. Actual 2047 MB Passed
5 Preparing to launch Oracle Universal Installer from
   ↳ /tmp/OraInstall2015-11-18_05-27-42PM. Please wait ...You can find the log
   ↳ of this install session at:
6   /u01/app/oracle/oraInventory/logs/installActions2015-11-18_05-27-42PM.log
7 The installation of Oracle Database 12c was successful.
8 Please check
   ↳ '/u01/app/oracle/oraInventory/logs/silentInstall2015-11-18_05-27-42PM.log'
   ↳ for more details.
9
10 As a root user, execute the following script(s):
11   1. ./u01/app/oracle/product/12.1.0.2/db_1/root.sh
12
13
14
15 Successfully Setup Software.

```

- Login or su as a root user, execute the following script(s):

```

1 [root@localhost ~]# ./u01/app/oracle/product/12.1.0.2/db_1/root.sh
2 [root@localhost ~]#

```

- Install Oracle dbca - Silent Install with dbca.rsp response file.

```

1 [oracle@localhost ~]$ cd /u01/app/oracle/product/12.1.0.2/db_1/bin
2 [oracle@localhost bin]$ ./dbca -silent -responseFile
   ↳ /tmp/database/response/dbca.rsp
3 [oracle@localhost bin]$

```

```

1 Copying database files
2 1% complete
3 3% complete
4 11% complete
5 18% complete
6 26% complete
7 37% complete
8 Creating and starting Oracle instance
9 40% complete
10 45% complete
11 50% complete
12 55% complete
13 56% complete
14 60% complete
15 62% complete
16 Completing Database Creation
17 66% complete
18 70% complete
19 73% complete
20 85% complete
21 96% complete
22 100% complete
23 Look at the log file "/u01/app/oracle/cfgtoollogs/dbca/orcl/orcl.log" for
   ↳ further details.

```

8.9 Post Installation

- These steps apply to both Windows and Mac
- Edit the "/etc/oratab" file setting the restart flag for each instance to 'Y'.

```

1 [root@localhost ~]# vi /etc/oratab
2 #
3
4
5
6 # This file is used by ORACLE utilities. It is created by root.sh
7 # and updated by either Database Configuration Assistant while creating
8 # a database or ASM Configuration Assistant while creating ASM instance.
9
10 # A colon, ':', is used as the field terminator. A new line terminates
11 # the entry. Lines beginning with a pound sign, '#', are comments.
12 #
13 # Entries are of the form:
14 #   $ORACLE_SID:$ORACLE_HOME:<N|Y>:
15 #
16 # The first and second fields are the system identifier and home
17 # directory of the database respectively. The third field indicates
18 # to the dbstart utility that the database should , "Y", or should not,
19 # "N", be brought up at system boot time.
20 #
21 # Multiple entries with the same $ORACLE_SID are not allowed.
22 #
23 #
24 orcl:/u01/app/oracle/product/12.1.0.2/db_1:Y
25 [root@localhost ~]#

```

8.10 TNS

- These steps apply to both Windows and Mac

```

1 [oracle@localhost ~]$ cd /u01/app/oracle/product/12.1.0.2/db_1/network/admin
2 [oracle@localhost admin]$ cat tnsnames.ora
3 # This file contains the syntax information for
4 # the entries to be put in any tnsnames.ora file
5 # The entries in this file are need based.
6 # There are no defaults for entries in this file
7 # that Sqlnet/Net3 use that need to be overridden
8 #
9 # Typically you could have two tnsnames.ora files
10 # in the system, one that is set for the entire system
11 # and is called the system tnsnames.ora file, and a
12 # second file that is used by each user locally so that
13 # he can override the definitions dictated by the system
14 # tnsnames.ora file.
15
16 # The entries in tnsnames.ora are an alternative to using
17 # the names server with the onames adapter.
18 # They are a collection of aliases for the addresses that
19 # the listener(s) is(are) listening for a database or
20 # several databases.
21
22 # The following is the general syntax for any entry in
23 # a tnsnames.ora file. There could be several such entries
24 # tailored to the user's needs.
25
26 orcl=(DESCRIPTION=
27   (ADDRESS
28     (PROTOCOL=tcp)
29     (HOST=localhost)
30     (PORT=1521)
31   )
32   (CONNECT_DATA=
33     (SID=orcl)
34   )
35 )
36
37
38 [oracle@localhost admin]$

```

8.11 Set Oracle as Service

- These steps apply to both Windows and Mac
- Create dbora file in /etc/init.d

```

1 [root@localhost ~]# vi /etc/init.d/dbora
2 #!/bin/bash
3 #
4 # chkconfig: 345 60 50
5 # description: Oracle auto start-stop script.
6
7 # Modify the variables $ORA_HOME, $ORA_OWNER and $LOG as appropriate for each
8 #       server.
8 #           - Nabeel Moidu
9
10 # Source function library.
11 . /etc/rc.d/init.d/functions
12 # Source networking configuration.
13 [ -r /etc/sysconfig/network ] && . /etc/sysconfig/network
14
15 # Modify the variables $ORA_HOME, $ORA_OWNER and $LOG as appropriate for each
15 #       server.
16 #           - Nabeel Moidu
17 ORA_HOME=/u01/app/oracle/product/12.1.0.2/db_1
18 ORA_OWNER=oracle
19 LOG=$ORA_HOME/log/dbora.log
20
21
22 if [ ! -f $ORA_HOME/bin/dbstart -o ! -d $ORA_HOME ]
23 then
24     echo "Oracle startup: cannot start"
25     exit
26 fi
27
28 case "$1" in
29     start)
30         echo "$0: starting up" >> $LOG
31         # Start the Oracle databases:
32         su - $ORA_OWNER -c "$ORA_HOME/bin/dbstart $ORA_HOME &" >> $LOG
33         # Start Oracle listener
34         date >> $LOG
35         #su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl start" >> $LOG 2>&1
36         touch /var/lock/subsys/dbora
37         echo "Refer to $LOG for details"
38         ;;
39     stop)
40         echo "$0: stopping down" >> $LOG
41         date >> $LOG
42         # Stop Oracle Net
43         #su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl stop" >> $LOG 2>&1
44         # Stop the Oracle databases:
45         su - $ORA_OWNER -c "$ORA_HOME/bin/dbshut &" >> $LOG
46         rm -f /var/lock/subsys/dbora
47         ;;
48     restart)
49         $0 stop
50         sleep 120
51         $0 start
52         ;;
53     status)
54         if [ -f /var/lock/subsys/dbora ]; then
55             echo $0 started.
56             su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl status"
57         else
58             echo $0 stopped.
59             su - $ORA_OWNER -c "$ORA_HOME/bin/lsnrctl status"
60         fi
61         ;;
62     *)
63         echo "usage: dbora {start|stop|restart|status}"
64         exit 1

```

```

65  esac
66  exit 0
67
68  [root@localhost ~]# chmod 755 /etc/init.d/dbora
69  [root@localhost ~]#

```

- Add dbora as a service

```

1  [root@localhost ~]# chkconfig --add dbora
2  [root@localhost ~]#

```

- reboot

8.12 Test Oracle

- These steps apply to both Windows and Mac
- sys

```

1  [oracle@localhost ~]$ sqlplus sys/password@//localhost/orcl as sysdba
2
3  SQL*Plus: Release 12.1.0.2.0 Production on Thu Nov 19 19:11:40 2015
4
5  Copyright (c) 1982, 2014, Oracle. All rights reserved.
6
7
8  Connected to:
9  Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
10 With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
     → options
11
12 SQL> select * from dual;
13
14 D
15 -
16 X
17
18 SQL> quit
19 Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
     → 64bit Production
20 With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
     → options
21 [oracle@localhost ~]$

```

- sysdba

```

1  [oracle@localhost ~]$ sqlplus / as sysdba
2
3  SQL*Plus: Release 12.1.0.2.0 Production on Thu Nov 19 19:13:56 2015
4
5  Copyright (c) 1982, 2014, Oracle. All rights reserved.
6
7
8  Connected to:
9  Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
10 With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
     → options
11
12 SQL> select * from dual;
13
14 D
15 -
16 X
17
18 SQL> quit
19 Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
     → 64bit Production

```

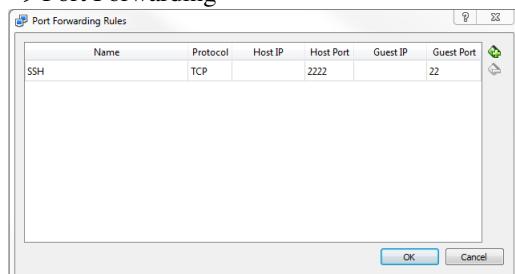
```

20 With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
  ↳ options
21 [oracle@localhost ~]$

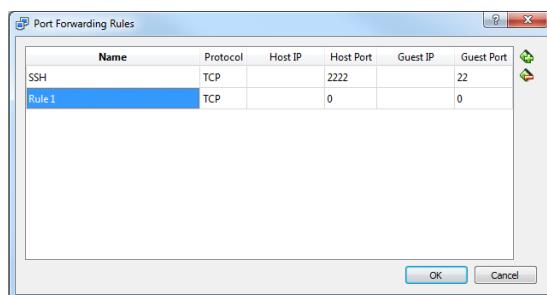
```

8.13 Map Oracle VM to host machine

- These steps apply to both Windows and Mac
- Stop your VirtualBox machine Click Settings ->Click Network ->Expand Advanced ->Port Forwarding



Click + sign to add a Port Forwarding Rule

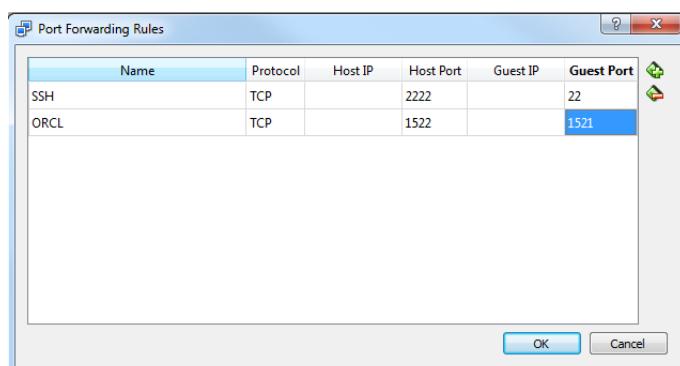


Name: ORCL

Protocol: TCP

Host Port: 1522

Guest Port: 1521

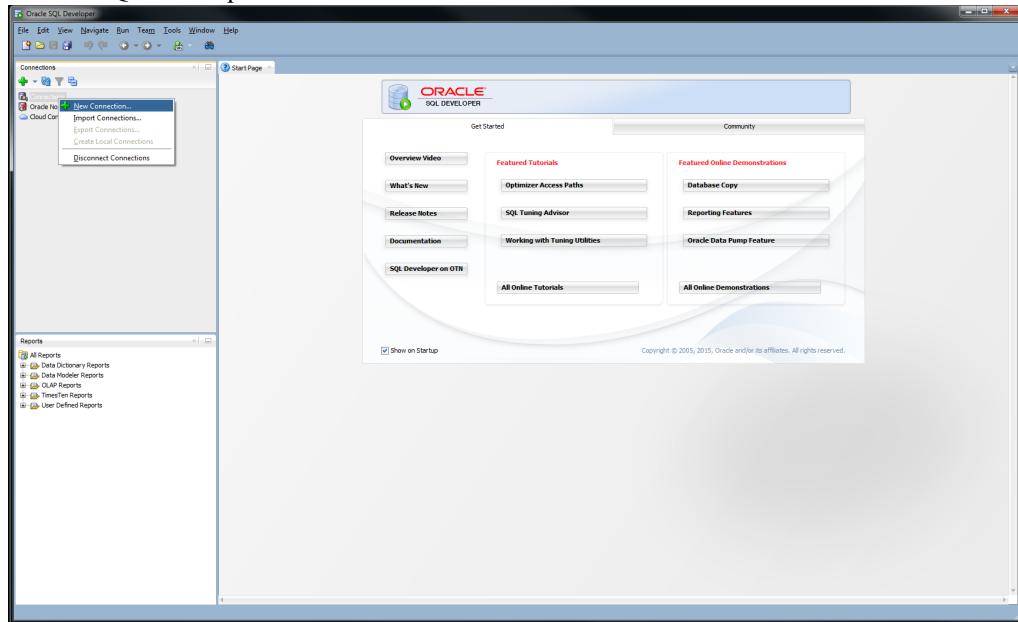


Click Ok, then click OK again

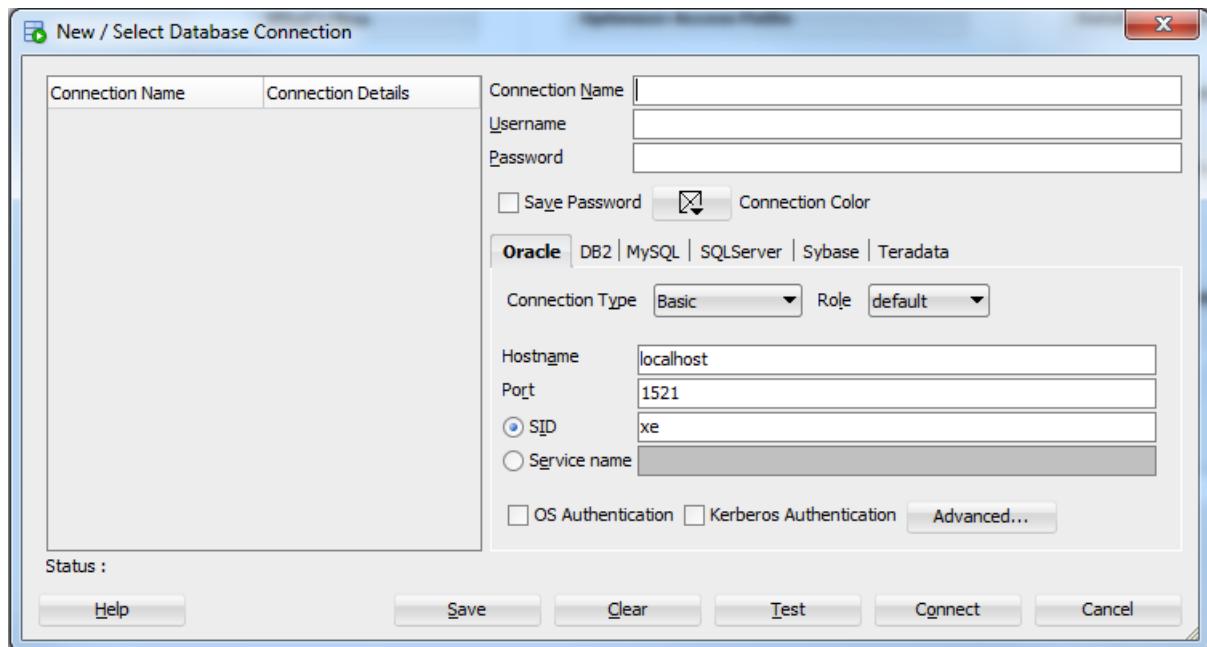
8.14 Connect to Oracle using SQL Developer

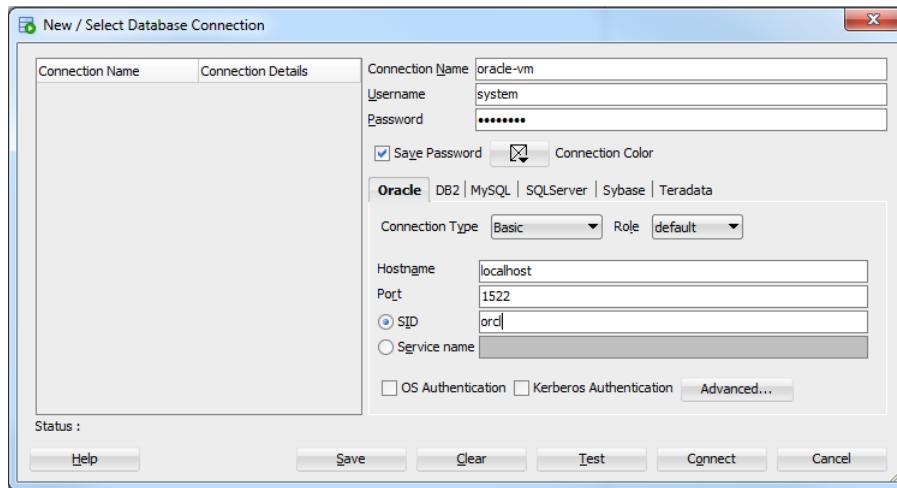
- These steps apply to both Windows and Mac

- Launch SQL Developer



Right click Connections → New Connection...





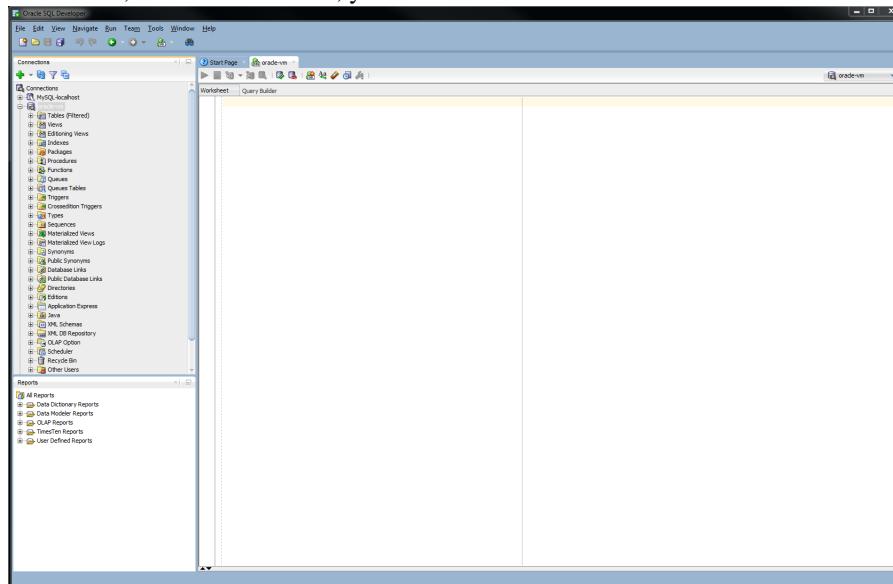
type in:

```
Connection Name: oracle-vm
Username: sys
Password: password
Save Password: checked
Hostname: localhost
Port: 1522
SID: orcl
```

Click Test

You will see Status : Success

Click Save, then click Connect, you are now connected to Oracle!



PART III

DOCKER

CHAPTER 9

UBUNTU LINUX ON VIRTUAL MACHINE

9.1 Linux

9.1.1 Download Ubuntu Linux

- Download Ubuntu Server 16.04 LTS (ubuntu-16.04-server-amd64.iso) from:
<http://www.ubuntu.com/download/server>

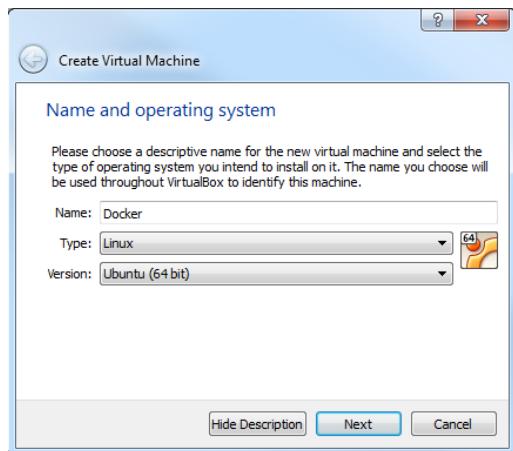
9.1.2 Install Ubuntu Linux - Create VM

- Launch VirtualBox



Click New

[†]for both windows and mac

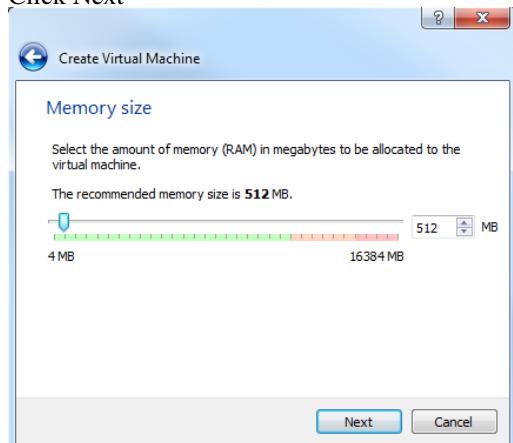


Name: Docker

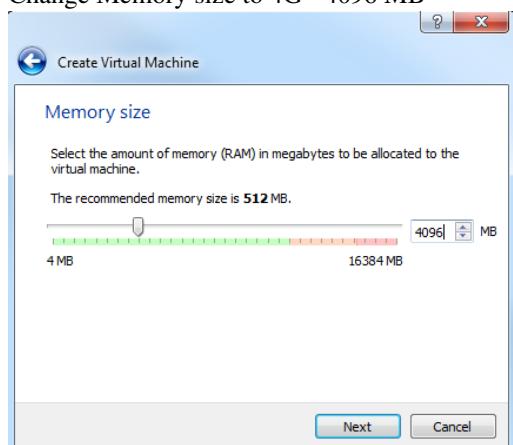
Type: Linux

Version: Ubuntu (64-bit)

Click Next



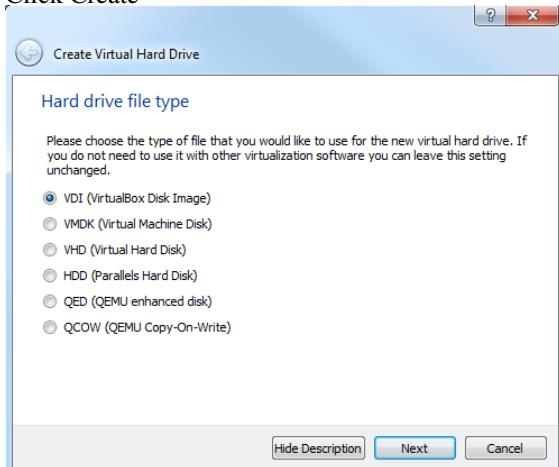
Change Memory size to 4G - 4096 MB



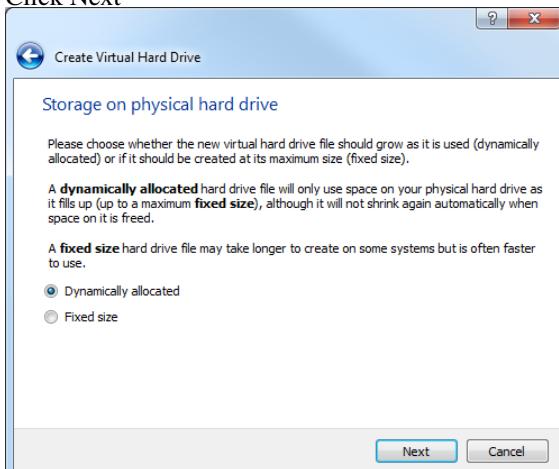
Click Next



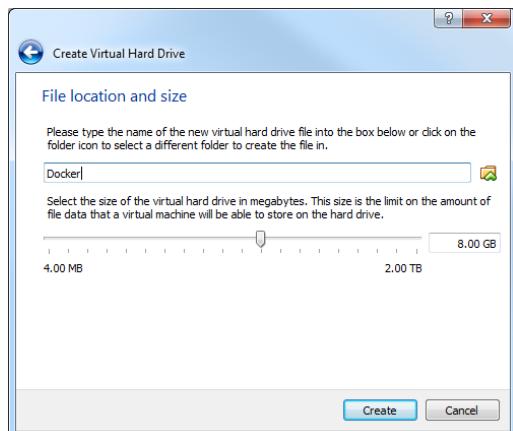
Click Create



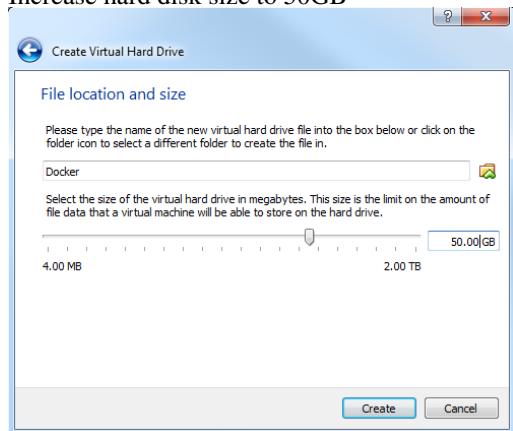
Click Next



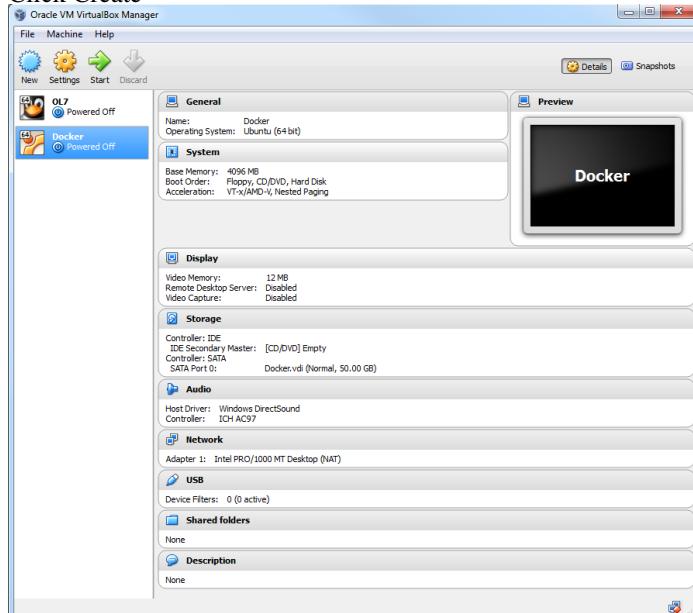
Click Next



Increase hard disk size to 50GB

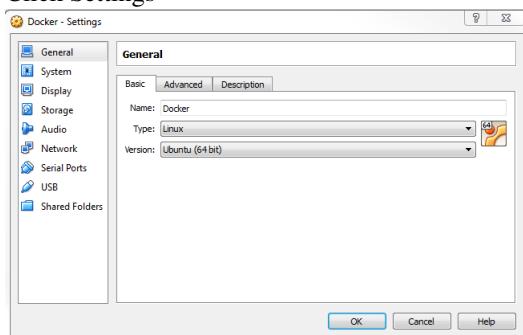


Click Create

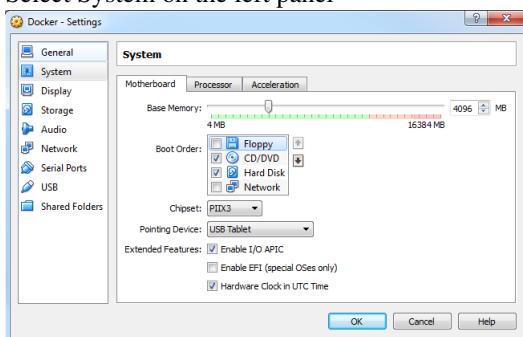


9.1.3 Ubuntu VM Configuration - Optional

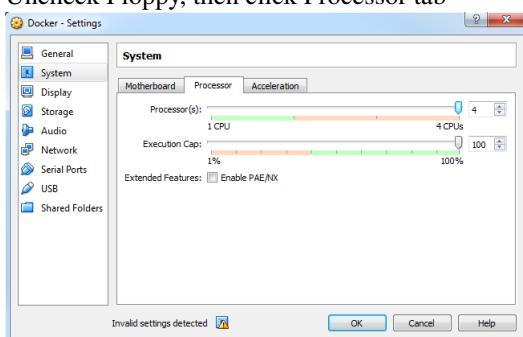
- Click Settings



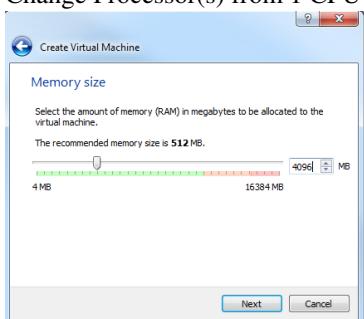
Select System on the left panel



Uncheck Floppy, then click Processor tab



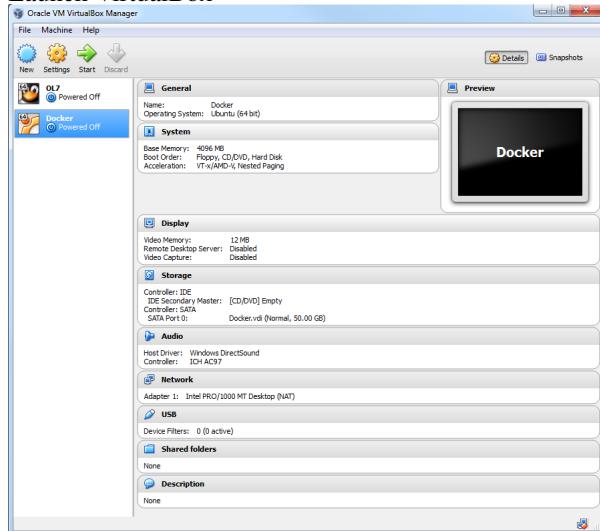
Change Processor(s) from 1 CPU to 4 CPU, then click Display on the left panel



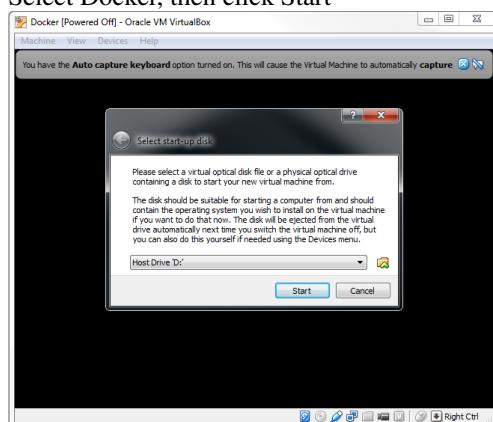
Change Video Memory from 12 MB to 128 MB, the click OK

9.1.4 Install Ubuntu Linux - From Ubuntu iso

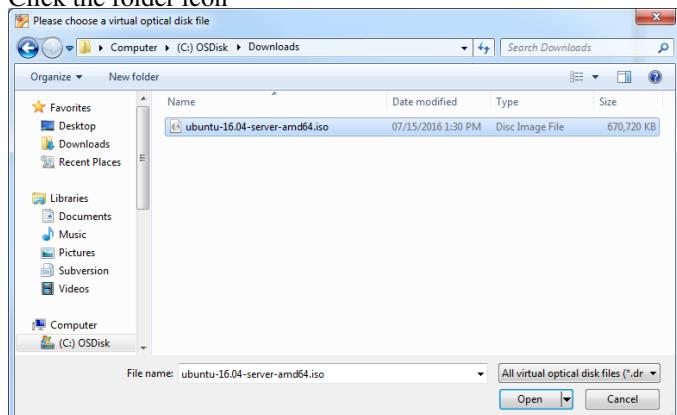
- Launch VirtualBox



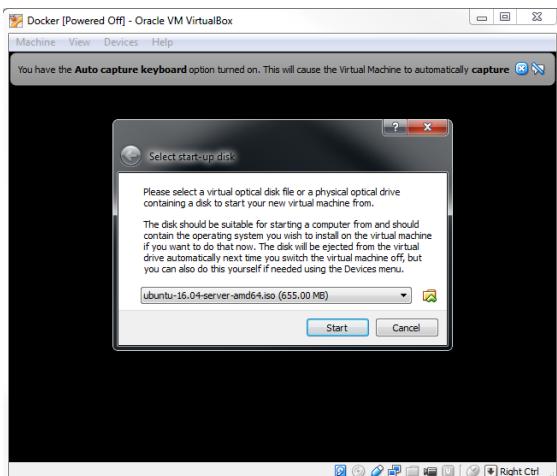
Select Docker, then click Start



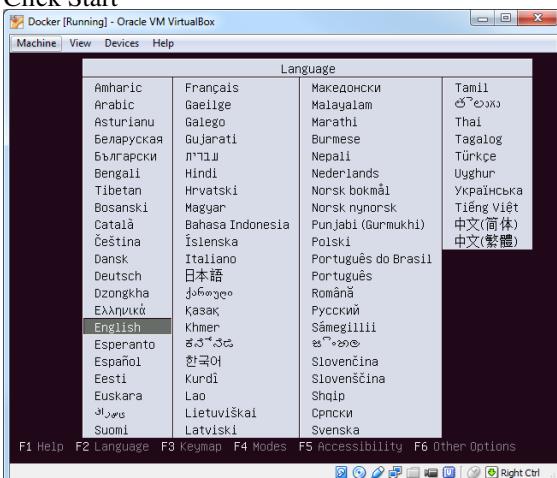
Click the folder icon



Browse to Ubuntu Linux iso file ubuntu-16.04-server-amd64.iso, click Open



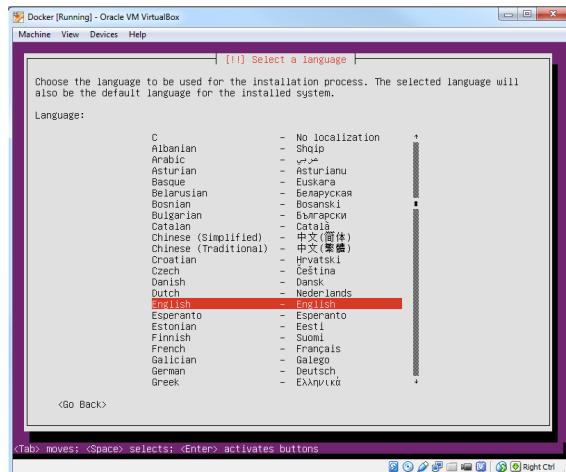
Click Start



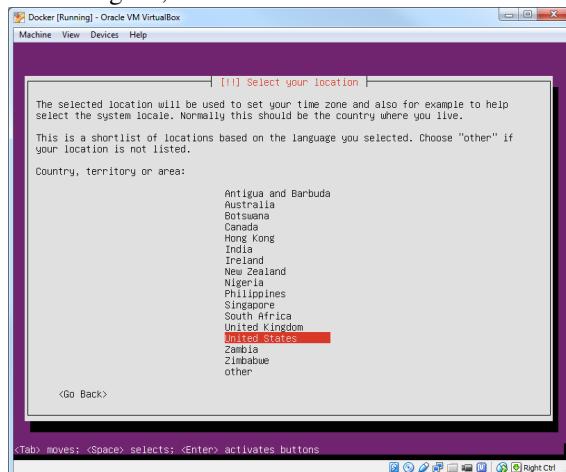
Select English, hit Enter



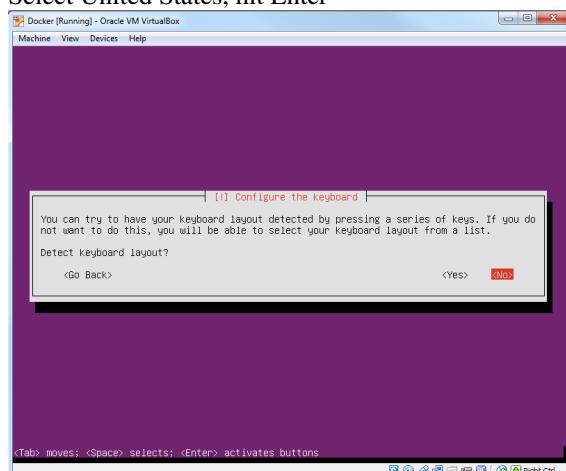
Select Install Ubuntu Server, hit Enter



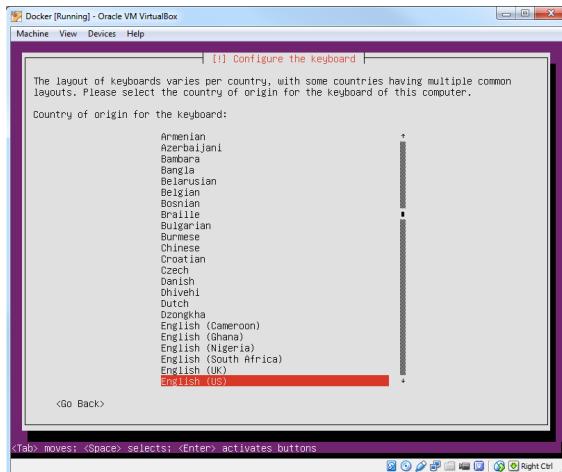
Select English, hit Enter



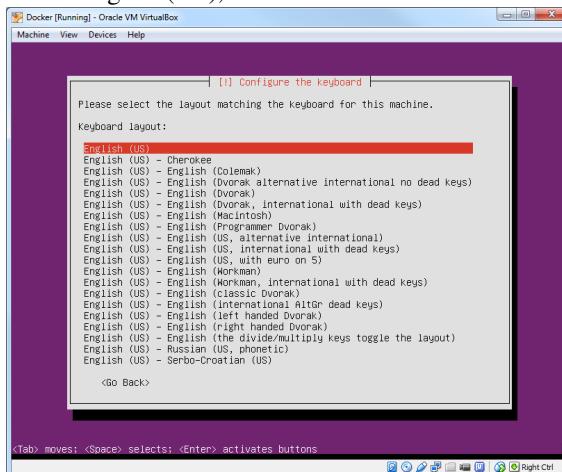
Select United States, hit Enter



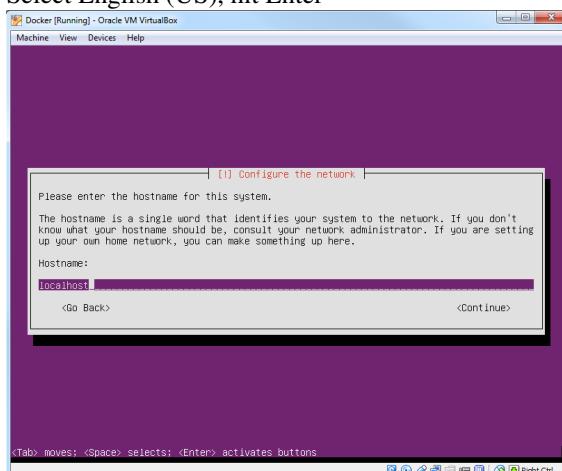
Select No, hit Enter



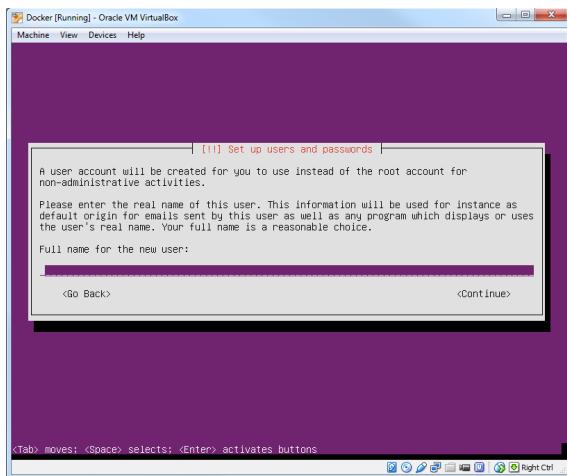
Select English (US), hit Enter



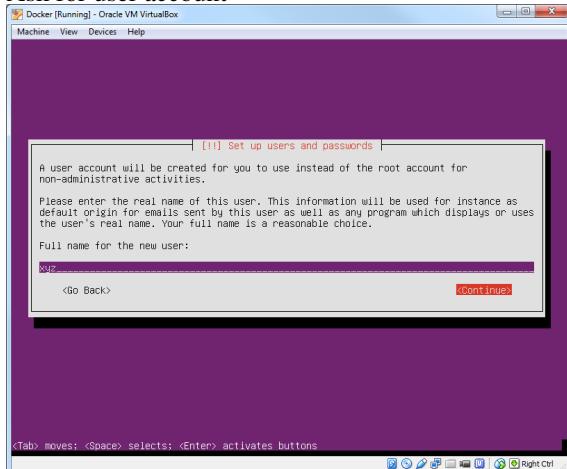
Select English (US), hit Enter



Type in Hostname, we use localhost he, then tab to Continue and hit Enter



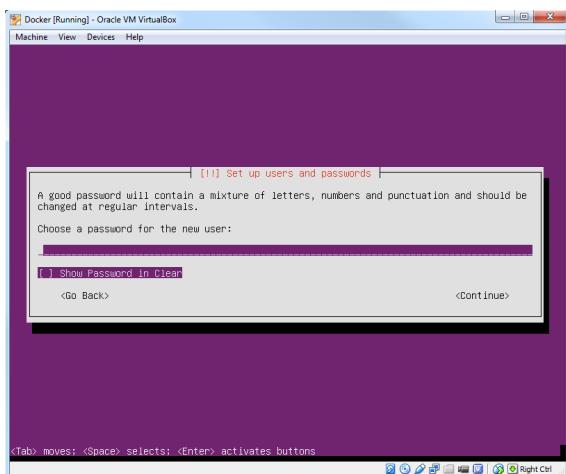
Ask for user account



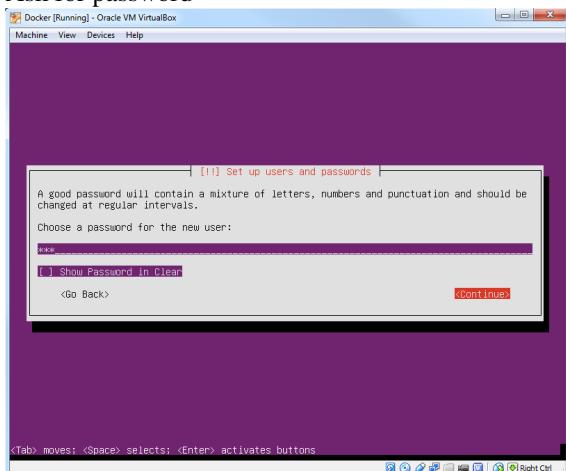
Type in Full name for the new user, then tab to Continue and hit Enter



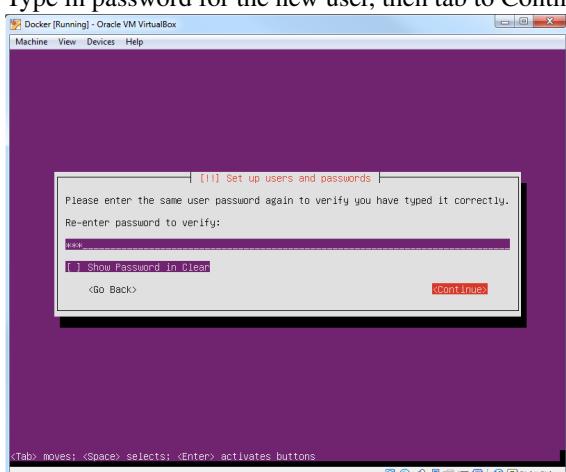
Type in Username for your account, then tab to Continue and hit Enter



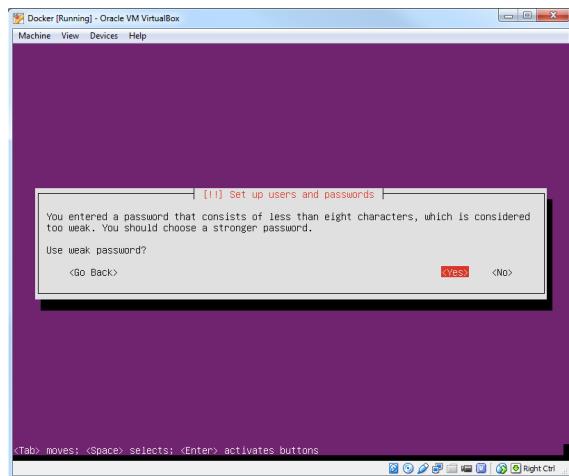
Ask for password



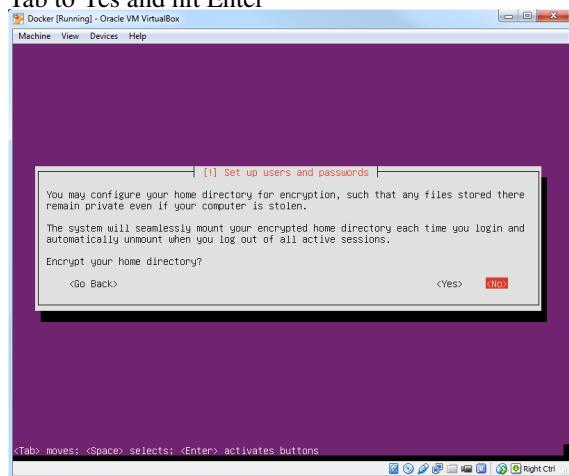
Type in password for the new user, then tab to Continue and hit Enter



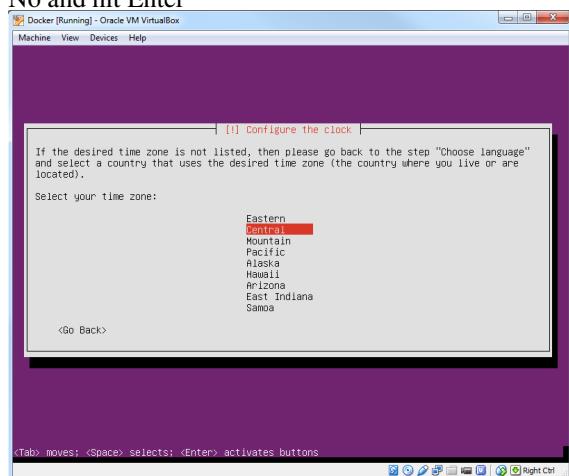
Re-Type password, then tab to Continue and hit Enter



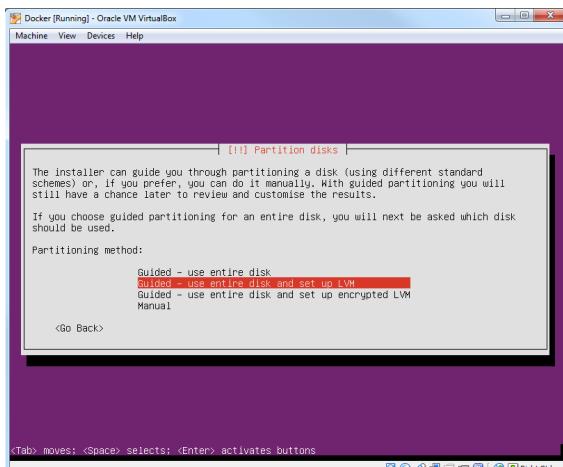
Tab to Yes and hit Enter



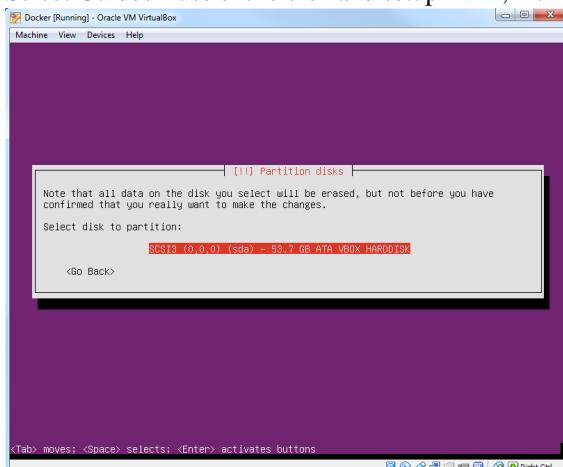
No and hit Enter



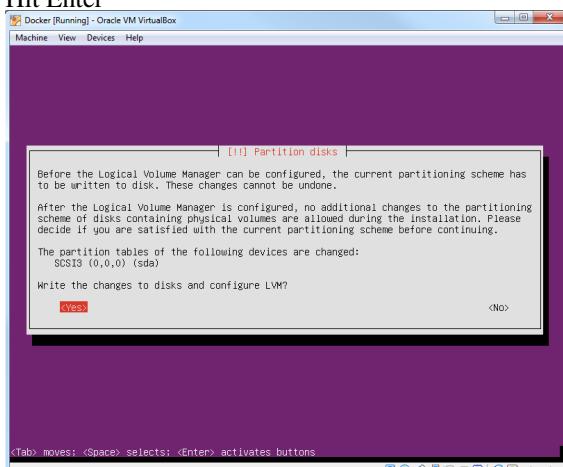
Select Central, hit Enter



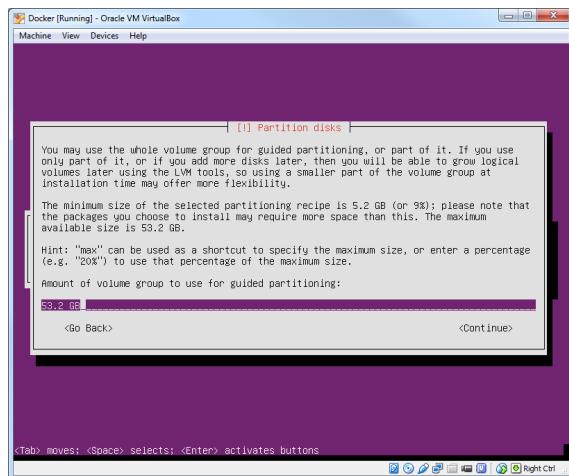
Select Guided - use entire disk and setup LVM, hit Enter



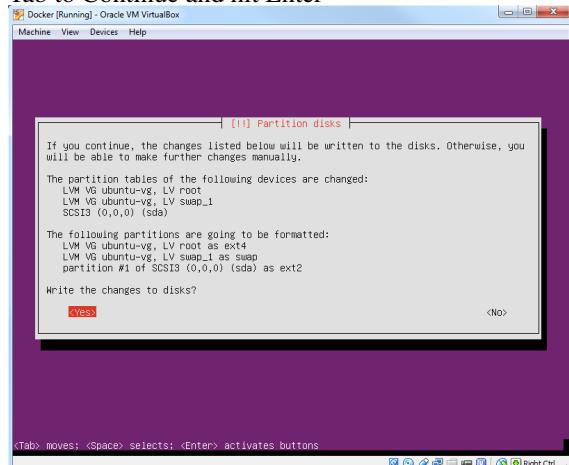
Hit Enter



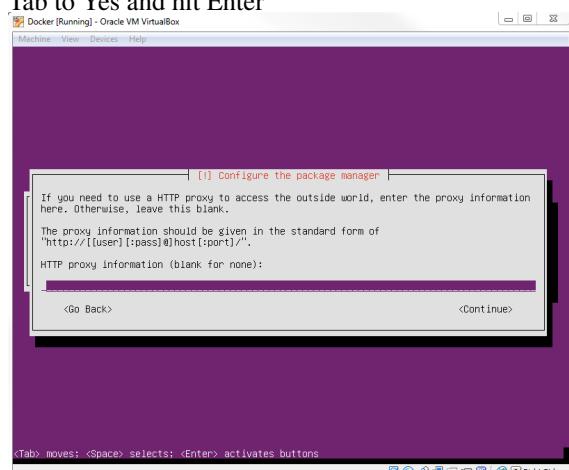
Tab to Yes and hit Enter



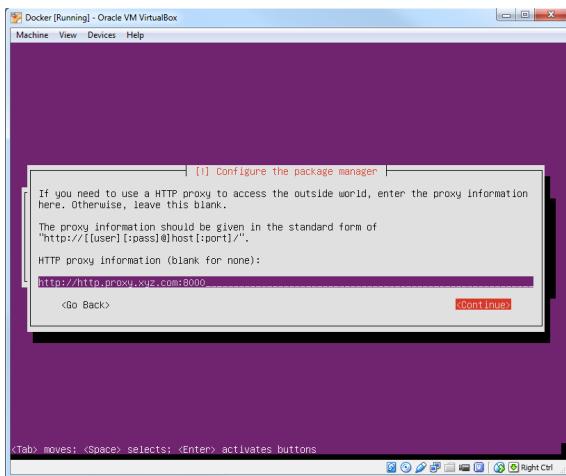
Tab to Continue and hit Enter



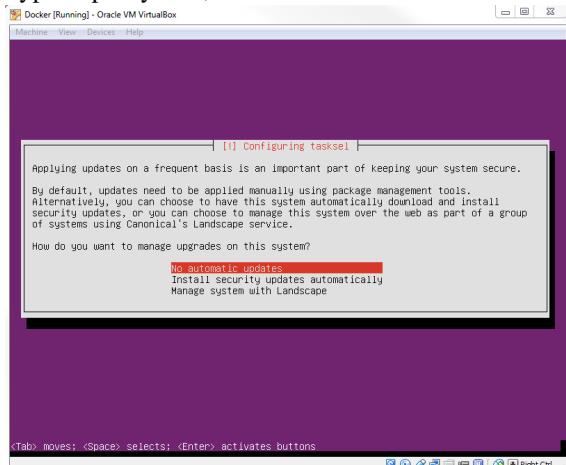
Tab to Yes and hit Enter



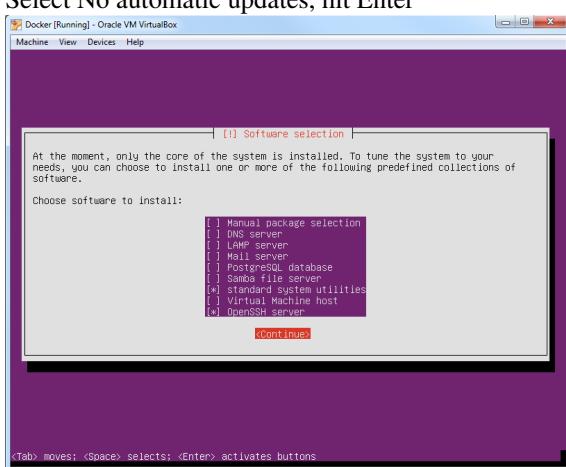
Asking proxy information



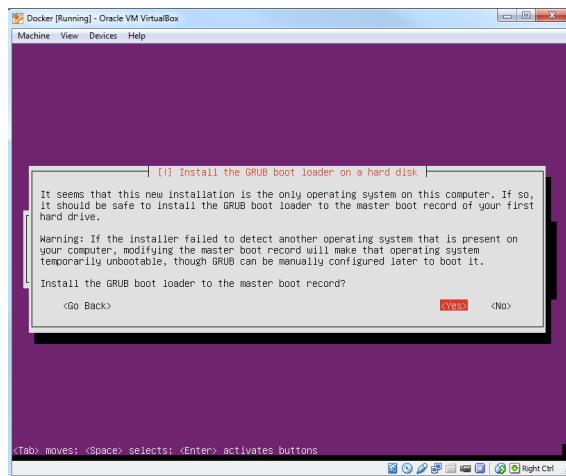
Type in proxy info, tab to Continue and hit Enter



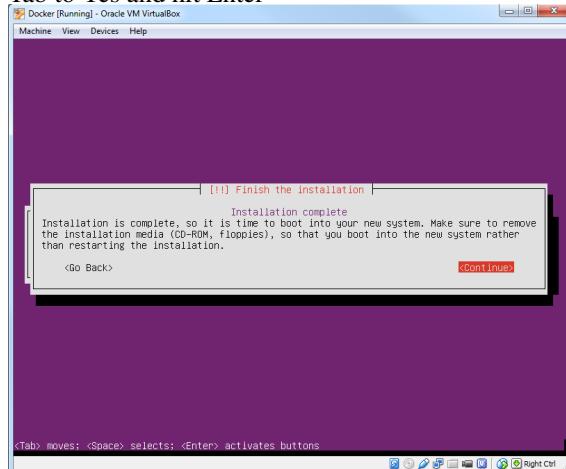
Select No automatic updates, hit Enter



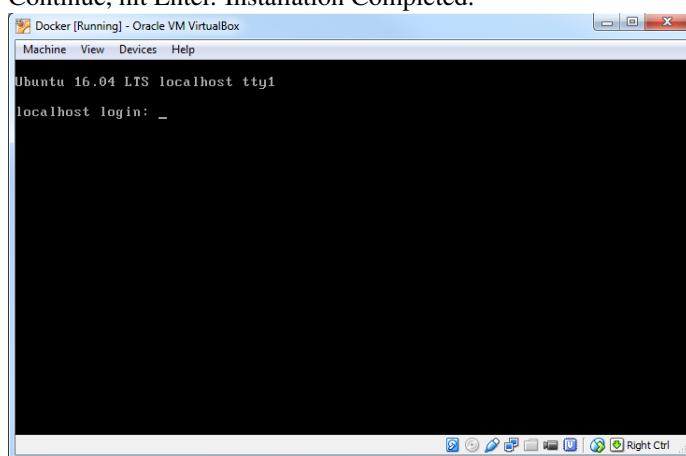
Select OpenSSH server, tab to Continue and hit Enter



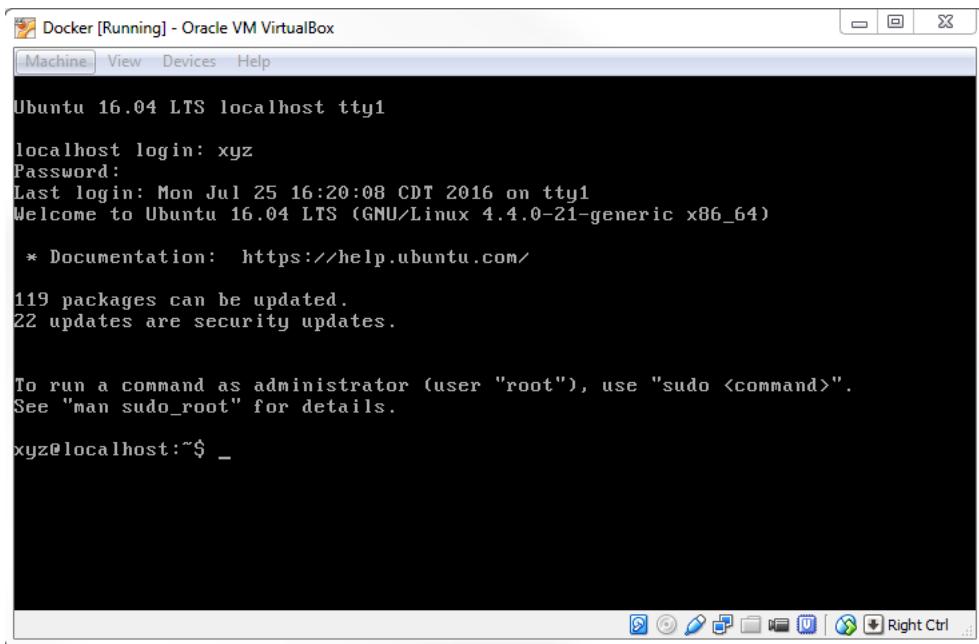
Tab to Yes and hit Enter



Continue, hit Enter. Installation Completed.



Login screen



```
Docker [Running] - Oracle VM VirtualBox
Machine View Devices Help

Ubuntu 16.04 LTS localhost tty1

localhost login: xyz
Password:
Last login: Mon Jul 25 16:20:08 CDT 2016 on tty1
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

119 packages can be updated.
22 updates are security updates.

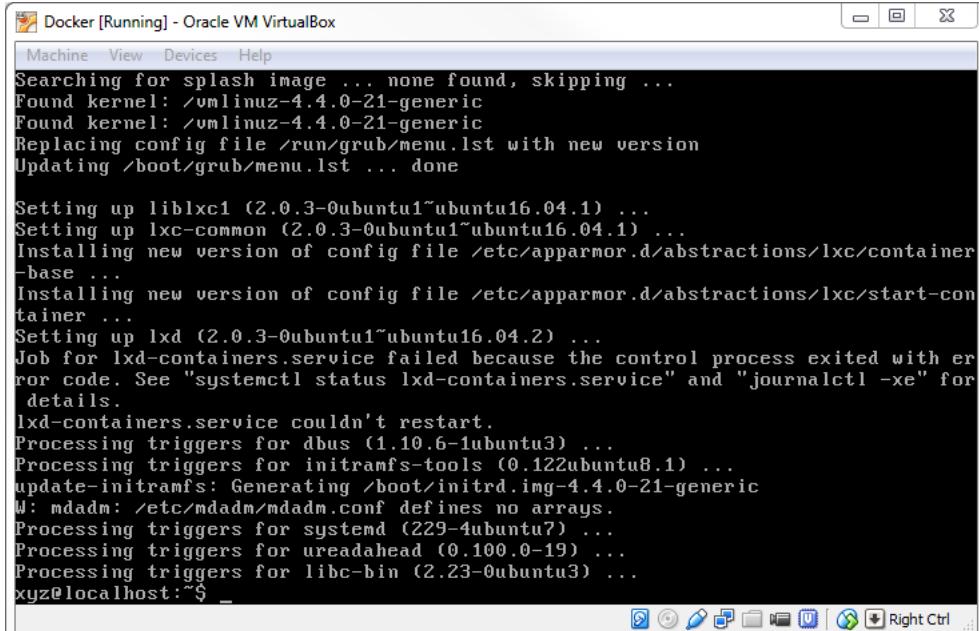
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

xyz@localhost:~$ _
```

Login

- Update your Ubuntu Server with the following command:

```
xyz@localhost:~$ sudo apt-get upgrade
```



```
Docker [Running] - Oracle VM VirtualBox
Machine View Devices Help

Searching for splash image ... none found, skipping ...
Found kernel: /vmlinuz-4.4.0-21-generic
Found kernel: /vmlinuz-4.4.0-21-generic
Replacing config file /run/grub/menu.lst with new version
Updating /boot/grub/menu.lst ... done

Setting up liblxc1 (2.0.3-0ubuntu1~ubuntu16.04.1) ...
Setting up lxc-common (2.0.3-0ubuntu1~ubuntu16.04.1) ...
Installing new version of config file /etc/apparmor.d/abstractions/lxc/container-base ...
Installing new version of config file /etc/apparmor.d/abstractions/lxc/start-container ...
Setting up lxd (2.0.3-0ubuntu1~ubuntu16.04.2) ...
Job for lxd-containers.service failed because the control process exited with error code. See "systemctl status lxd-containers.service" and "journalctl -xe" for details.
lxd-containers.service couldn't restart.
Processing triggers for dbus (1.10.6-1ubuntu3) ...
Processing triggers for initramfs-tools (0.122ubuntu8.1) ...
update-initramfs: Generating /boot/initrd.img-4.4.0-21-generic
W: mdadm: /etc/mdadm/mdadm.conf defines no arrays.
Processing triggers for systemd (229-4ubuntu7) ...
Processing triggers for ureadahead (0.100.0-19) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
xyz@localhost:~$ _
```

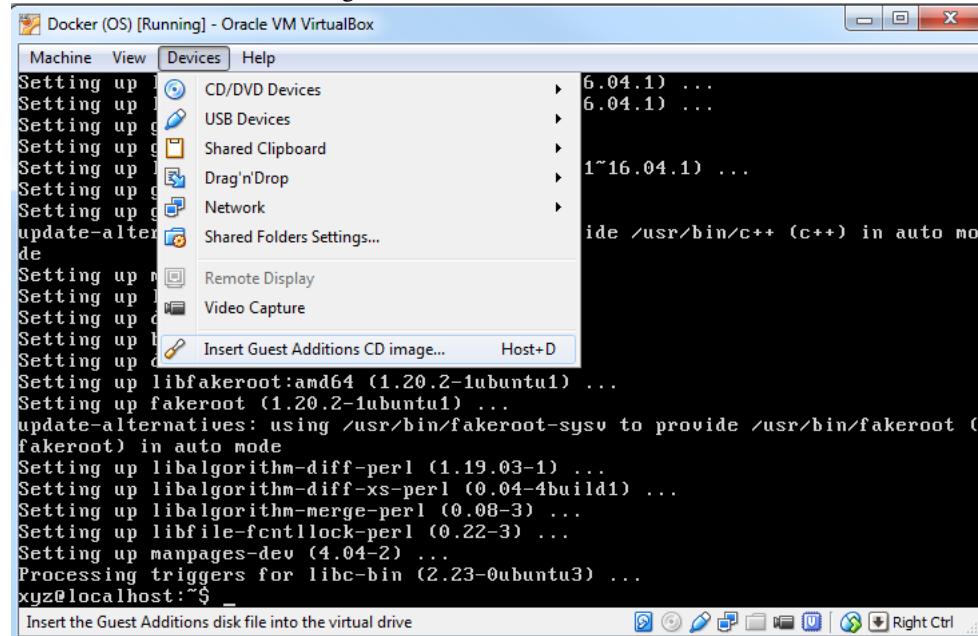
The Ubuntu Server is ready to use

9.2 Install Linux Guest Additions

- Install packages

```
1 xyz@localhost:~$ sudo apt-get install dkms build-essential linux-headers-$(uname
→ -r)
```

- Insert Guest Additions CD image...



Go to Devices → Insert Guest Additions CD image...

- mount cdrom

```
1 xyz@localhost:~$ sudo mount /dev/cdrom /media/cdrom
2 mount: /dev/sr0 is write-protected, mounting read-only
3 xyz@localhost:~$
```

- install guest additions

```
1 xyz@localhost:~$ cd /media/cdrom
2 xyz@localhost:/media/cdrom$ sudo ./VBoxLinuxAdditions.run
```

- Reboot your guest system.

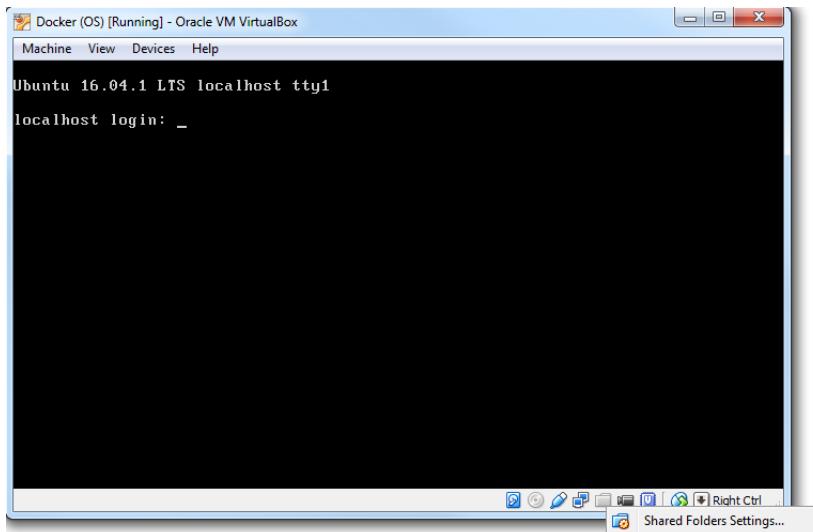
- You can Remove disk from virtual drive by right click the cdrom icon on the bottom of the VM window.

- Install additional packages

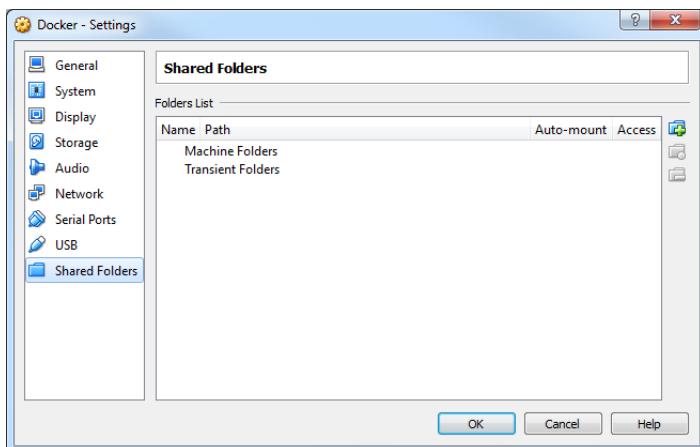
```
1 xyz@localhost:~$ sudo apt-get install virtualbox-guest-dkms
→ linux-headers-generic
```

- Reboot your guest system.

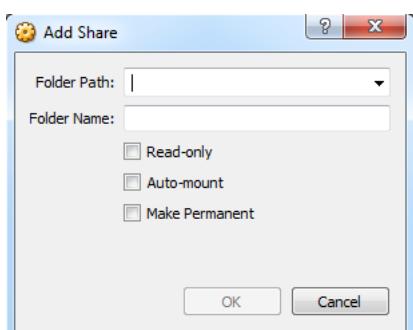
9.3 Shared folders



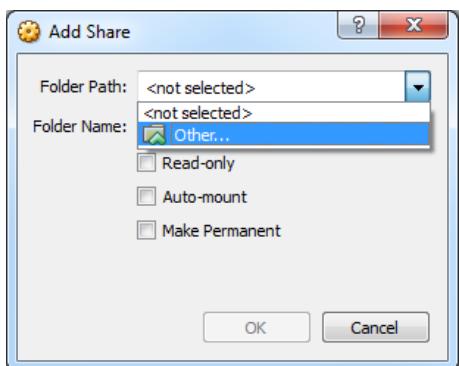
Right click guest window Shared folder icon on the bottom right corner. Select Shared Folders Settings...



Click the + sign on the right.

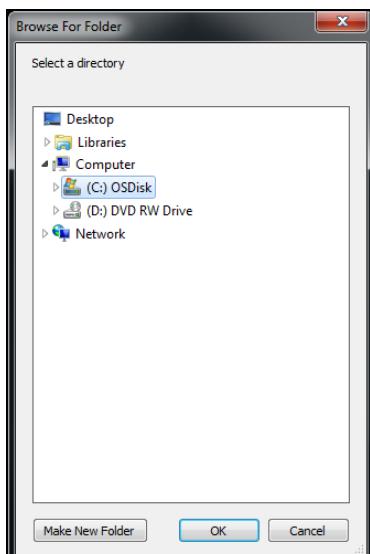


Select drop down menu for Folder Path.

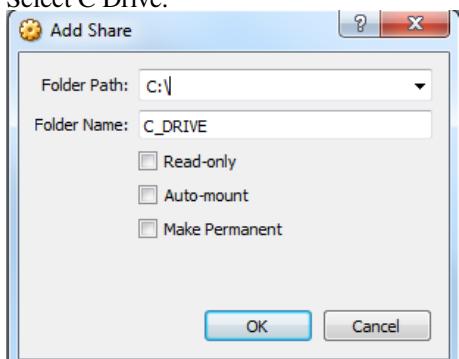


Select Other...

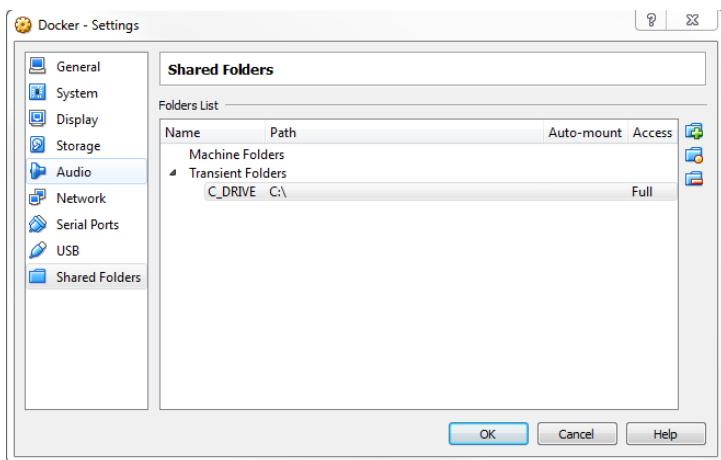
▪ For Windows



Select C Drive.



Click OK.



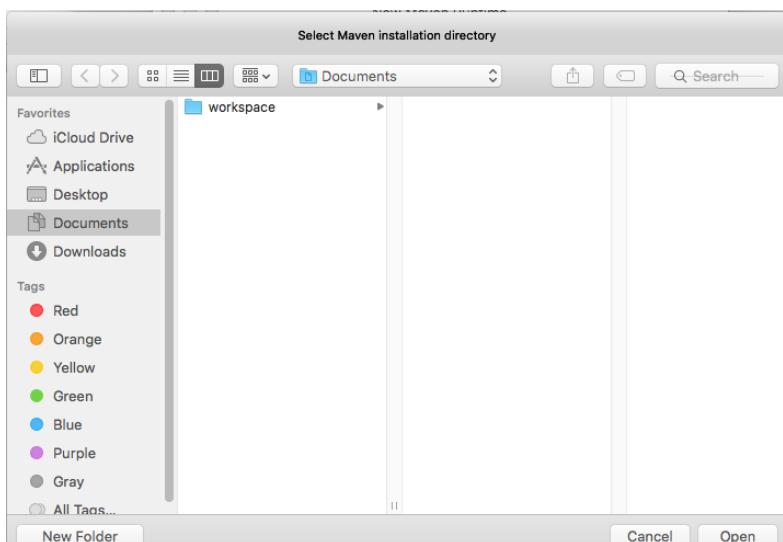
Click OK.

▪ mount C Drive

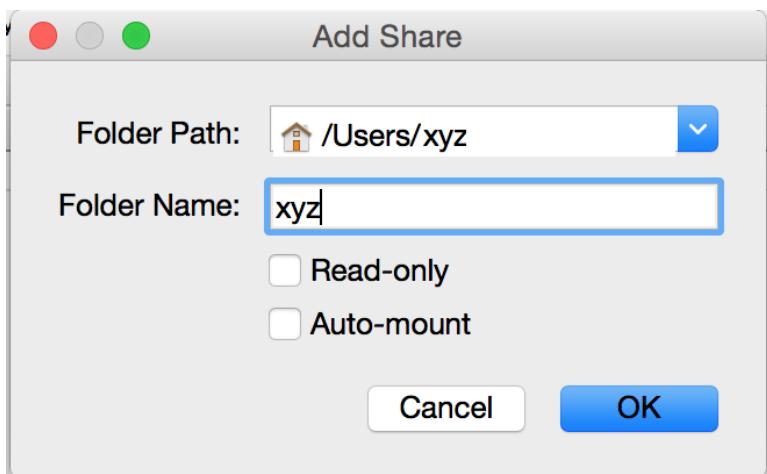
```

1 xyz@localhost:~$ cd /media
2 xyz@localhost:/media$ sudo mkdir share
3 xyz@localhost:/media$ sudo mount -t vboxsf C_DRIVE /media/share
4 xyz@localhost:/media$ cd share
5 xyz@localhost:/media/share$ ls -al
6 total 14249044
7 lrwxrwxrwx. 1 root root      0 Jul 14  2009 Documents and Settings
8 drwxrwxrwx. 1 root root  12288 Nov 10 10:07 ProgramData
9 dr-xr-xr-x. 1 root root   8192 Jun 11 09:08 Program Files
10 drwxrwxrwx. 1 root root  20480 Oct 22 10:01 Program Files (x86)
11 dr-xr-xr-x. 1 root root   4096 Apr 10  2015 Users
12 drwxrwxrwx. 1 root root  40960 Oct 30 14:43 Windows
13 ...
14 xyz@localhost:/media/share$
```

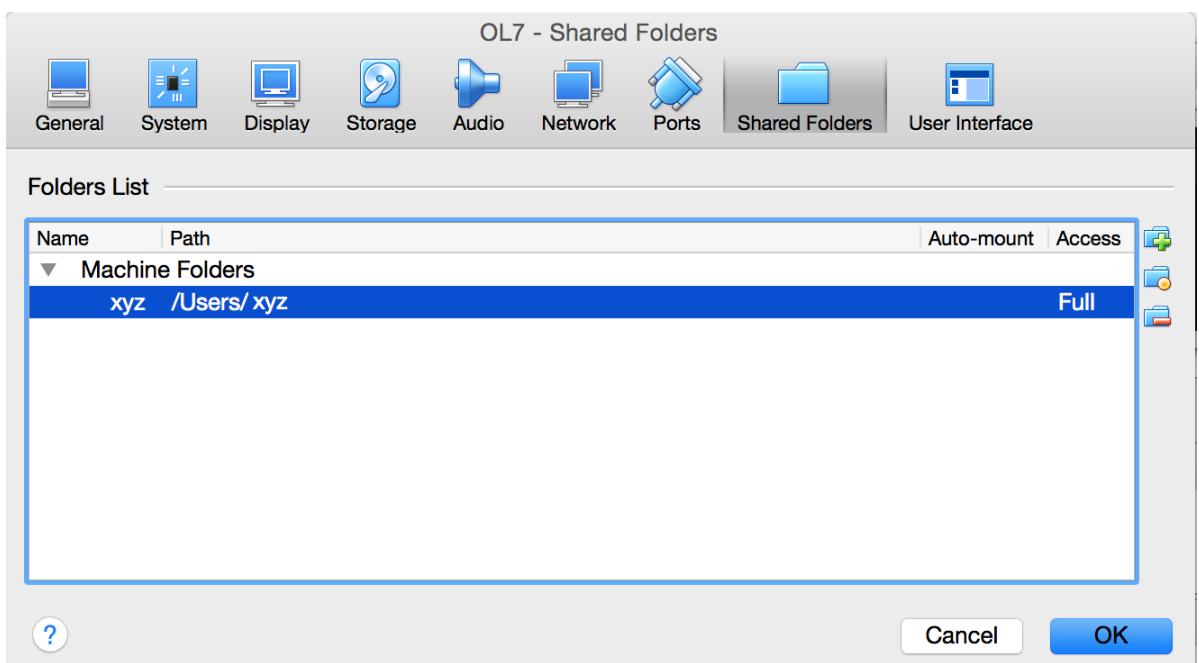
▪ For Mac



Select /Users/xyz



Click OK.



Click OK.

▪ mount xyz

```

1 xyz@localhost:~$ cd /media
2 xyz@localhost:/media$ sudo mkdir share
3 xyz@localhost:/media$ sudo mount -t vboxsf xyz /media/share
4 xyz@localhost:/media$ cd share
5 xyz@localhost:/media/share$ ls -al
6 ...
7 ...
8 ...
9 ...
10 xyz@localhost:/media/share$
```

9.4 Network

- Check network status

```

1 xyz@localhost:~$ ls /sys/class/net
2 enp0s3  lo
3
4 xyz@localhost:~$ cat /etc/network/interfaces
5 # This file describes the network interfaces available on your system
6 # and how to activate them. For more information, see interfaces(5).
7
8 source /etc/network/interfaces.d/*
9
10 # The loopback network interface
11 auto lo
12 iface lo inet loopback
13
14 # The primary network interface
15 auto enp0s3
16 iface enp0s3 inet dhcp
17
18 xyz@localhost:~$ ifconfig
19 enp0s3      Link encap:Ethernet HWaddr 08:00:27:38:95:db
20             inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0
21             inet6 addr: fe80::a00:27ff:fe38:95db/64 Scope:Link
22             UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
23             RX packets:167 errors:0 dropped:0 overruns:0 frame:0
24             TX packets:117 errors:0 dropped:0 overruns:0 carrier:0
25             collisions:0 txqueuelen:1000
26             RX bytes:16221 (16.2 KB)  TX bytes:17728 (17.7 KB)
27
28 lo          Link encap:Local Loopback
29             inet addr:127.0.0.1 Mask:255.0.0.0
30             inet6 addr: ::1/128 Scope:Host
31             UP LOOPBACK RUNNING MTU:65536 Metric:1
32             RX packets:0 errors:0 dropped:0 overruns:0 frame:0
33             TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
34             collisions:0 txqueuelen:1
35             RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
36
37 xyz@localhost:~$ ip addr show
38 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
39     >   default qlen 1
40     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
41     inet 127.0.0.1/8 scope host lo
42         valid_lft forever preferred_lft forever
43     inet6 ::1/128 scope host
44         valid_lft forever preferred_lft forever
45 2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
46     >   group default qlen 1000
47     link/ether 08:00:27:38:95:db brd ff:ff:ff:ff:ff:ff
48     inet 10.0.2.15/24 brd 10.0.2.255 scope global enp0s3
49         valid_lft forever preferred_lft forever
50         inet6 fe80::a00:27ff:fe38:95db/64 scope link
51             valid_lft forever preferred_lft forever
52 xyz@localhost:~$
```

- Setup Proxy, add http_proxy to /etc/environment file

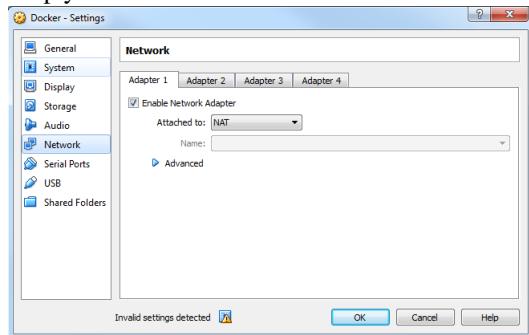
```

1 xyz@localhost:~$ sudo vi /etc/environment
2
3 PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games"
4 http_proxy="http://http.proxy.xyz.com:8000/"
5
6 xyz@localhost:~$
```

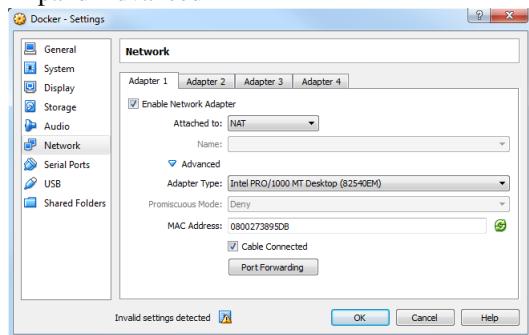
- Logoff and login

9.5 Connect to VM - VM Network Setting

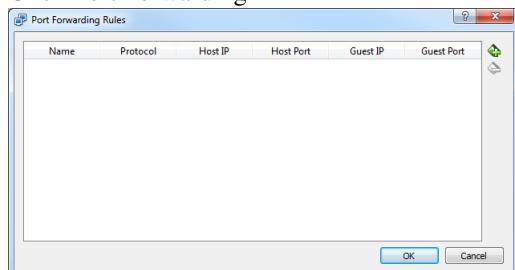
- Stop your VirtualBox machine



Click Settings –>Network
Expand Advanced



Click Port Forwarding



Click + sign to add a Port Forwarding Rule

Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
SSH	TCP		2222		22

Name: SSH / Protocol: TCP / Host Port: 2222 / Guest Port: 22

Click Ok, then click OK again

- You can use PuTTY / WinSCP / SSH, connect to localhost:2222, Ubuntu linux

CHAPTER 10

DOCKER ON UBUNTU LINUX

10.1 Prerequisites

10.1.1 Ubuntu kernel

- Docker requires a 64-bit installation regardless of your Ubuntu version. Additionally, your kernel must be 3.10 at minimum.
- To check current kernel version, open a terminal and use uname -r to display your kernel version:

```
1 xyz@localhost:~$ uname -r
2 4.4.0-31-generic
3 xyz@localhost:~$
```

10.1.2 Update apt sources

- Update package information.

```
1 xyz@localhost:~$ sudo apt-get update
2 [sudo] password for xyz:
3 Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
4 Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [94.5 kB]
5 Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [95.7 kB]
6 Hit:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
```

[†]this chapter is running on ubuntu linux vm, for both windows and mac

```

7  Get:5 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages
   → [300 kB]
8  Get:6 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe i386 Packages
   → [297 kB]
9  Fetched 788 kB in 4s (186 kB/s)
10 Reading package lists... Done
11 xyz@localhost:~$
```

- Ensure that APT works with the https method, and that CA certificates are installed.

```

1 xyz@localhost:~$ sudo apt-get install apt-transport-https ca-certificates
2 Reading package lists... Done
3 Building dependency tree
4 Reading state information... Done
5 ca-certificates is already the newest version (20160104ubuntu1).
6 apt-transport-https is already the newest version (1.2.12~ubuntu16.04.1).
7 0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
8 xyz@localhost:~$
```

- Add the new GPG key

```

1 xyz@localhost:~$ sudo apt-key adv --keyserver
   ↳ hkp://p80.pool.sks-keyservers.net:80 --recv-keys
   ↳ 58118E89F3A912897C070ADBF76221572C52609D
2 Executing: /tmp/tmp.jNL7GP3teB/gpg.1.sh --keyserver
3 hkp://p80.pool.sks-keyservers.net:80
4 --recv-keys
5 58118E89F3A912897C070ADBF76221572C52609D
6 gpg: requesting key 2C52609D from hkp server p80.pool.sks-keyservers.net
7 gpg: key 2C52609D: public key "Docker Release Tool (releasedocker)"
   ↳ <docker@docker.com> imported
8 gpg: Total number processed: 1
9 gpg:           imported: 1 (RSA: 1)
10 xyz@localhost:~$
```

We can check keyserver status by browse to:

<http://p80.pool.sks-keyservers.net/pks/lookup?op=stats>

- docker.list file

Open the /etc/apt/sources.list.d/docker.list file in your favorite editor.

If the file doesn't exist, create it.

Remove any existing entries.

Add an entry for your Ubuntu operating system.

For our Ubuntu Xenial 16.04 (LTS)

```

1 xyz@localhost:~$ sudo vi /etc/apt/sources.list.d/docker.list
2
3 deb https://apt.dockerproject.org/repo ubuntu-xenial main
4
5 xyz@localhost:~$
```

For Other Ubuntu versions

```

1 # On Ubuntu Precise 12.04 (LTS)
2 deb https://apt.dockerproject.org/repo ubuntu-precise main
3 # On Ubuntu Trusty 14.04 (LTS)
4 deb https://apt.dockerproject.org/repo ubuntu-trusty main
5 # Ubuntu Wily 15.10
6 deb https://apt.dockerproject.org/repo ubuntu-wily main
```

- Update the APT package index

```

1 xyz@localhost:~$ sudo apt-get update -y
2 Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
```

```

3  Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [94.5 kB]
4  Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [95.7 kB]
5  Hit:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
6  Get:5 https://apt.dockerproject.org/repo ubuntu-xenial InRelease [20.6 kB]
7  Get:6 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
   → [1,719 B]
8  Fetched 212 kB in 0s (234 kB/s)
9  Reading package lists... Done
10 xyz@localhost:~$
```

- Purge the old repo if it exists

```

1 xyz@localhost:~$ sudo apt-get purge lxc-docker
2 Reading package lists... Done
3 Building dependency tree
4 Reading state information... Done
5 Package 'lxc-docker' is not installed, so not removed
6 0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
7 xyz@localhost:~$
```

- Verify that APT is pulling from the right repository

```

1 xyz@localhost:~$ apt-cache policy docker-engine
2 docker-engine:
3   Installed: (none)
4   Candidate: 1.11.2-0~xenial
5   Version table:
6     1.11.2-0~xenial 500
7       500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
8     1.11.1-0~xenial 500
9       500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
10    1.11.0-0~xenial 500
11      500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
12 xyz@localhost:~$
```

From now on when you run apt-get upgrade, APT pulls from the new repository.

10.1.3 Recommend package

- For Ubuntu Trusty, Wily, and Xenial, its recommended to install the linux-image-extra kernel package. The linux-image-extra package allows you use the aufs storage driver.
- Update your package manager

```

1 xyz@localhost:~$ sudo apt-get update
2 Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
3 Hit:2 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
4 Get:3 http://security.ubuntu.com/ubuntu xenial-security InRelease [94.5 kB]
5 Hit:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
6 Hit:5 https://apt.dockerproject.org/repo ubuntu-xenial InRelease
7 Fetched 94.5 kB in 0s (136 kB/s)
8 Reading package lists... Done
9 xyz@localhost:~$
```

- Install the recommended package

```

1 xyz@localhost:~$ sudo apt-get install linux-image-extra-$(uname -r)
2 Reading package lists... Done
3 Building dependency tree
4 Reading state information... Done
5 linux-image-extra-4.4.0-31-generic is already the newest version (4.4.0-31.50).
6 linux-image-extra-4.4.0-31-generic set to manually installed.
7 0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
8 xyz@localhost:~$
```

10.2 Install Docker

- Update your APT package index

```

1 xyz@localhost:~$ sudo apt-get update
2 Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease
3 Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [94.5 kB]
4 Hit:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
5 Hit:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
6 Hit:5 https://apt.dockerproject.org/repo ubuntu-xenial InRelease
7 Fetched 94.5 kB in 0s (145 kB/s)
8 Reading package lists... Done
9 xyz@localhost:~$
```

- Install Docker

```

1 xyz@localhost:~$ sudo apt-get install docker-engine
2 Reading package lists... Done
3 Building dependency tree
4 Reading state information... Done
5 The following additional packages will be installed:
6   aufs-tools cgroupfs-mount
7 Suggested packages:
8   mountall
9 The following NEW packages will be installed:
10  aufs-tools cgroupfs-mount docker-engine
11 0 upgraded, 3 newly installed, 0 to remove and 5 not upgraded.
12 Need to get 14.6 MB of archives.
13 After this operation, 73.7 MB of additional disk space will be used.
14 Do you want to continue? [Y/n] y
15 Get:1 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 aufs-tools amd64
16   → 1:3.2+20130722-1.lubuntul [92.9 kB]
16 Get:2 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 docker-engine
17   → amd64 1.11.2-0~xenial [14.5 MB]
17 Get:3 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 cgroupfs-mount
18   → all 1.2 [4,970 B]
18 Fetched 14.6 MB in 8s (1,621 kB/s)
19 Selecting previously unselected package aufs-tools.
20 (Reading database ... 112320 files and directories currently installed.)
21 Preparing to unpack .../aufs-tools_1%3a3.2+20130722-1.lubuntul_amd64.deb ...
22 Unpacking aufs-tools (1:3.2+20130722-1.lubuntul) ...
23 Selecting previously unselected package cgroupfs-mount.
24 Preparing to unpack .../cgroupfs-mount_1.2_all.deb ...
25 Unpacking cgroupfs-mount (1.2) ...
26 Selecting previously unselected package docker-engine.
27 Preparing to unpack .../docker-engine_1.11.2-0~xenial_amd64.deb ...
28 Unpacking docker-engine (1.11.2-0~xenial) ...
29 Processing triggers for libc-bin (2.23-0ubuntu3) ...
30 Processing triggers for man-db (2.7.5-1) ...
31 Processing triggers for ureadahead (0.100.0-19) ...
32 Processing triggers for systemd (229-4ubuntu7) ...
33 Setting up aufs-tools (1:3.2+20130722-1.lubuntul) ...
34 Setting up cgroupfs-mount (1.2) ...
35 Setting up docker-engine (1.11.2-0~xenial) ...
36 Processing triggers for libc-bin (2.23-0ubuntu3) ...
37 Processing triggers for systemd (229-4ubuntu7) ...
38 Processing triggers for ureadahead (0.100.0-19) ...
39 xyz@localhost:~$
```

- Start the docker daemon

```

1 xyz@localhost:~$ sudo systemctl start docker
2 # or on older distributions, you may need to use
3 xyz@localhost:~$ sudo service docker start
4 xyz@localhost:~$
```

- Check Docker Version

```

1 xyz@localhost:~$ sudo docker version
2 Client:
3   Version:      1.11.2
4   API version:  1.23
5   Go version:   go1.5.4
6   Git commit:   b9f10c9
7   Built:        Wed Jun  1 22:00:43 2016
8   OS/Arch:      linux/amd64
9
10 Server:
11   Version:      1.11.2
12   API version:  1.23
13   Go version:   go1.5.4
14   Git commit:   b9f10c9
15   Built:        Wed Jun  1 22:00:43 2016
16   OS/Arch:      linux/amd64
17 xyz@localhost:~$
```

▪ Check docker info

```

1 xyz@localhost:~$ sudo docker info
2 Containers: 0
3   Running: 0
4   Paused: 0
5   Stopped: 0
6 Images: 0
7 Server Version: 1.11.2
8 Storage Driver: aufs
9   Root Dir: /var/lib/docker/aufs
10  Backing Filesystem: extfs
11  Dirs: 0
12  Dirperm1 Supported: true
13  Logging Driver: json-file
14  Cgroup Driver: cgroupfs
15  Plugins:
16    Volume: local
17    Network: bridge null host
18  Kernel Version: 4.4.0-31-generic
19  Operating System: Ubuntu 16.04.1 LTS
20  OSType: linux
21  Architecture: x86_64
22  CPUs: 4
23  Total Memory: 3.859 GiB
24  Name: localhost
25  ID: QOYN:IOEL:MSHS:2ONW:IOAS:OQDK:P5RI:VIWO:KJRW:VJAS:JOLJ:3KDV
26  Docker Root Dir: /var/lib/docker
27  Debug mode (client): false
28  Debug mode (server): false
29  Registry: https://index.docker.io/v1/
30  WARNING: No swap limit support
31 xyz@localhost:~$
```

▪ Stop the docker daemon

```

1 xyz@localhost:~$ sudo systemctl stop docker
2 # or on older distributions, you may need to use
3 xyz@localhost:~$ sudo service docker stop
```

▪ If you want Docker to start at boot, you should also:

```

1 xyz@localhost:~$ sudo systemctl enable docker
2 [sudo] password for xyz:
3 Synchronizing state of docker.service with SysV init with
4   →  /lib/systemd/systemd-sysv-install...
4 Executing /lib/systemd/systemd-sysv-install enable docker
5 xyz@localhost:~$
6 # or on older distributions, you may need to use
7 xyz@localhost:~$ sudo chkconfig docker on
8 xyz@localhost:~$
```

10.3 Control and configure Docker with systemd

10.3.1 Custom Docker daemon options

- There are a number of ways to configure the daemon flags and environment variables for your Docker daemon.
- The recommended way is to use a systemd drop-in file. These are local files named <something>.conf in the /etc/systemd/system/docker.service.d directory. This could also be /etc/systemd/system/docker.service, which also works for overriding the defaults from /lib/systemd/system/docker.service.
- find out where the service file is located

```

1 xyz@localhost:~$ systemctl show --property=FragmentPath docker
2 FragmentPath=/lib/systemd/system/docker.service
3
4 xyz@localhost:~$ cat /lib/systemd/system/docker.service
5 [Unit]
6 Description=Docker Application Container Engine
7 Documentation=https://docs.docker.com
8 After=network.target docker.socket
9 Requires=docker.socket
10
11 [Service]
12 Type=notify
13 # the default is not to use systemd for cgroups because the delegate issues
14 # still
15 # exists and systemd currently does not support the cgroup feature set required
16 # for containers run by docker
17 ExecStart=/usr/bin/docker daemon -H fd://
18 MountFlags=slave
19 LimitNOFILE=1048576
20 LimitNPROC=1048576
21 LimitCORE=infinity
22 TimeoutStartSec=0
23 # set delegate yes so that systemd does not reset the cgroups of docker
24 # containers
25 Delegate=yes
26
27 [Install]
28 WantedBy=multi-user.target
xyz@localhost:~$
```

- You can customize the Docker daemon options using override files. The files located in /usr/lib/systemd/system or /lib/systemd/system contain the default options and should not be edited.

10.3.2 HTTP proxy

- This example overrides the default docker.service file.
- If you are behind an HTTP proxy server, for example in corporate settings, you will need to add this configuration in the Docker systemd service file.
- First, create a systemd drop-in directory for the docker service:

```

1 xyz@localhost:~$ sudo mkdir /etc/systemd/system/docker.service.d
2 xyz@localhost:~$
```

- Now create a file called /etc/systemd/system/docker.service.d/http-proxy.conf that adds the HTTP_PROXY environment variable:

```

1 xyz@localhost:~$ sudo vi /etc/systemd/system/docker.service.d/http-proxy.conf
2
```

```

3 [Service]
4 Environment="HTTP_PROXY=http://http.proxy.xyz.com:8000/"
5
6 xyz@localhost:~$
```

- If you have internal Docker registries that you need to contact without proxying you can specify them via the NO_PROXY environment variable:

```

1 xyz@localhost:~$ sudo vi /etc/systemd/system/docker.service.d/http-proxy.conf
2
3 [Service]
4 Environment="HTTP_PROXY=http://http.proxy.xyz.com:8000/"
5   ↳ "NO_PROXY=localhost,127.0.0.1,docker-registry.xyz.com"
6
7 xyz@localhost:~$
```

- Flush changes:

```

1 # If Docker daemon not started
2 xyz@localhost:~$ sudo systemctl start docker
3 # If Docker daemon already started
4 xyz@localhost:~$ sudo systemctl daemon-reload
5 xyz@localhost:~$
```

- Verify that the configuration has been loaded:

```

1 xyz@localhost:~$ sudo systemctl show --property=Environment docker
2 Environment=HTTP_PROXY=http://http.proxy.xyz.com:8000/
3 xyz@localhost:~$
```

- Restart Docker:

```

1 xyz@localhost:~$ sudo systemctl restart docker
2 xyz@localhost:~$
```

- Check Docker status:

```

1 xyz@localhost:~$ sudo service docker status
2 docker.service - Docker Application Container Engine
3   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
4     ↳ enabled)
4   Drop-In: /etc/systemd/system/docker.service.d
5     ↳ http-proxy.conf
6     Active: active (running) since Wed 2016-07-27 15:48:39 CDT; 14min ago
7       Docs: https://docs.docker.com
8     Main PID: 3387 (docker)
9       Tasks: 20
10      Memory: 36.3M
11        CPU: 6.791s
12      CGroup: /system.slice/docker.service
13           3387 /usr/bin/docker daemon -H fd://
14           3395 docker-containerd -l
15             ↳ /var/run/docker/libcontainerd/docker-containerd.sock --runtime
16               ↳ docker-runc --start-timeout 2m
17
18 Jul 27 15:48:39 localhost docker[3387]:
19   ↳ time="2016-07-27T15:48:39.544910723-05:00" level=info msg="Firewalld
20     ↳ running: false"
21 Jul 27 15:48:39 localhost docker[3387]:
22   ↳ time="2016-07-27T15:48:39.628826695-05:00" level=info msg="Default bridge
23     ↳ (docker0) is assigned with an IP address 172.17.0.0/16. Daemon option --bip
24     ↳ can be used to set a preferred IP address"
25 Jul 27 15:48:39 localhost docker[3387]:
26   ↳ time="2016-07-27T15:48:39.667906522-05:00" level=warning msg="Your kernel
27     ↳ does not support swap memory limit."
28 Jul 27 15:48:39 localhost docker[3387]:
29   ↳ time="2016-07-27T15:48:39.668667792-05:00" level=info msg="Loading
30     ↳ containers: start."
```

```

20 Jul 27 15:48:39 localhost docker[3387]:
21   ↳ time="2016-07-27T15:48:39.668784008-05:00" level=info msg="Loading
22     ↳ containers: done."
21 Jul 27 15:48:39 localhost docker[3387]:
22   ↳ time="2016-07-27T15:48:39.668832618-05:00" level=info msg="Daemon has
23     ↳ completed initialization"
22 Jul 27 15:48:39 localhost docker[3387]:
23   ↳ time="2016-07-27T15:48:39.668862230-05:00" level=info msg="Docker daemon"
24     ↳ commit=b9f10c9 graphdriver=aufs version=1.11.2
23 Jul 27 15:48:39 localhost docker[3387]:
24   ↳ time="2016-07-27T15:48:39.691928849-05:00" level=info msg="API listen on
25     ↳ /var/run/docker.sock"
24 Jul 27 15:48:39 localhost systemd[1]: Started Docker Application Container
25   ↳ Engine.
25 Jul 27 15:54:24 localhost docker[3387]:
26   ↳ time="2016-07-27T15:54:24.437788689-05:00" level=error msg="Handler for
26     ↳ POST /v1.23/containers/create returned error: No such image:
27       ↳ hello-world:latest"
26 xyz@localhost:~$
```

10.4 Docker - Hello World

- Verify docker is installed correctly

```

1 xyz@localhost:~$ sudo docker run hello-world
2 Unable to find image 'hello-world:latest' locally
3 latest: Pulling from library/hello-world
4
5 c04b14da8d14: Pull complete
6 Digest: sha256:0256e8a36e2070f7bf2d0b0763dbabdd67798512411de4cdcf9431a1feb60fd9
7 Status: Downloaded newer image for hello-world:latest
8
9 Hello from Docker!
10 This message shows that your installation appears to be working correctly.
11
12 To generate this message, Docker took the following steps:
13 1. The Docker client contacted the Docker daemon.
14 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
15 3. The Docker daemon created a new container from that image which runs the
16   executable that produces the output you are currently reading.
17 4. The Docker daemon streamed that output to the Docker client, which sent it
18   to your terminal.
19
20 To try something more ambitious, you can run an Ubuntu container with:
21 $ docker run -it ubuntu bash
22
23 Share images, automate workflows, and more with a free Docker Hub account:
24 https://hub.docker.com
25
26 For more examples and ideas, visit:
27 https://docs.docker.com/engine/userguide/
28 xyz@localhost:~$
```

This command downloads a test image and runs it in a container. When the container runs, it prints an informational message. Then, it exits.

10.5 Docker group

- The docker daemon binds to a Unix socket instead of a TCP port. By default that Unix socket is owned by the user root and other users can access it with sudo. For this reason, docker daemon always runs as the root user.

- To avoid having to use sudo when you use the docker command, create a Unix group called docker and add users to it. When the docker daemon starts, it makes the ownership of the Unix socket read/writable by the docker group.
- Check /etc/group, docker group already created

```

1 xyz@localhost:~$ cat /etc/group
2 root:x:0:
3 daemon:x:1:
4 ...
5 users:x:100:
6 ...
7 xyz:x:1000:
8 ...
9 docker:x:998:
10 xyz@localhost:~$
```

- Add your user to docker group

```

1 xyz@localhost:~$ sudo usermod -aG docker $(whoami)
2 xyz@localhost:~$
```

- Log out and log back in. This ensures your user is running with the correct permissions.
- Verify your work by running docker without sudo.

```

1 xyz@localhost:~$ docker run hello-world
2
3 Hello from Docker!
4 This message shows that your installation appears to be working correctly.
5
6 To generate this message, Docker took the following steps:
7 1. The Docker client contacted the Docker daemon.
8 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
9 3. The Docker daemon created a new container from that image which runs the
10   executable that produces the output you are currently reading.
11 4. The Docker daemon streamed that output to the Docker client, which sent it
12   to your terminal.
13
14 To try something more ambitious, you can run an Ubuntu container with:
15 $ docker run -it ubuntu bash
16
17 Share images, automate workflows, and more with a free Docker Hub account:
18 https://hub.docker.com
19
20 For more examples and ideas, visit:
21 https://docs.docker.com/engine/userguide/
22
23 xyz@localhost:~$
```

10.6 Upgrade Docker

- To install the latest version of Docker with apt-get

```

1 xyz@localhost:~$ sudo apt-get upgrade docker-engine
2 Reading package lists... Done
3 Building dependency tree
4 Reading state information... Done
5 docker-engine is already the newest version (1.11.2-0~xenial).
6 Calculating upgrade... Done
7 The following packages have been kept back:
8   vim vim-common vim-runtime vim-tiny
9 The following packages will be upgraded:
10    liblvm3.8
```

```

11 1 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
12 Need to get 10.3 MB of archives.
13 After this operation, 0 B of additional disk space will be used.
14 Do you want to continue? [Y/n] y
15 Get:1 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libl1vm3.8
   → amd64 1:3.8-2ubuntu4 [10.3 MB]
16 Fetched 10.3 MB in 6s (1,477 kB/s)
17 (Reading database ... 112484 files and directories currently installed.)
18 Preparing to unpack .../libl1vm3.8_1%3a3.8-2ubuntu4_amd64.deb ...
19 Unpacking libl1vm3.8:amd64 (1:3.8-2ubuntu4) over (1:3.8-2ubuntu3) ...
20 Processing triggers for libc-bin (2.23-0ubuntu3) ...
21 Setting up libl1vm3.8:amd64 (1:3.8-2ubuntu4) ...
22 Processing triggers for libc-bin (2.23-0ubuntu3) ...
23 xyz@localhost:~$
```

10.7 Uninstallation

- To uninstall the Docker package

```

1 xyz@localhost:~$ sudo apt-get purge docker-engine
2 Reading package lists... Done
3 Building dependency tree
4 Reading state information... Done
5 The following packages were automatically installed and are no longer required:
6   aufs-tools cgroupfs-mount
7 Use 'sudo apt autoremove' to remove them.
8 The following packages will be REMOVED:
9   docker-engine*
10 0 upgraded, 0 newly installed, 1 to remove and 4 not upgraded.
11 After this operation, 73.4 MB disk space will be freed.
12 Do you want to continue? [Y/n] y
13 (Reading database ... 112484 files and directories currently installed.)
14 Removing docker-engine (1.11.2-0~xenial) ...
15 Purging configuration files for docker-engine (1.11.2-0~xenial) ...
16 Processing triggers for man-db (2.7.5-1) ...
17 xyz@localhost:~$
```

- To uninstall the Docker package and dependencies that are no longer needed

```

1 xyz@localhost:~$ sudo apt-get autoremove --purge docker-engine
2 Reading package lists... Done
3 Building dependency tree
4 Reading state information... Done
5 Package 'docker-engine' is not installed, so not removed
6 The following packages will be REMOVED:
7   aufs-tools cgroupfs-mount*
8 0 upgraded, 0 newly installed, 2 to remove and 4 not upgraded.
9 After this operation, 283 kB disk space will be freed.
10 Do you want to continue? [Y/n] y
11 (Reading database ... 112382 files and directories currently installed.)
12 Removing aufs-tools (1:3.2+20130722-1.lubuntul) ...
13 Purging configuration files for aufs-tools (1:3.2+20130722-1.lubuntul) ...
14 Removing cgroupfs-mount (1.2) ...
15 Purging configuration files for cgroupfs-mount (1.2) ...
16 Processing triggers for libc-bin (2.23-0ubuntu3) ...
17 Processing triggers for man-db (2.7.5-1) ...
18 xyz@localhost:~$
```

The above commands will not remove images, containers, volumes, or user created configuration files on your host.

- If you wish to delete all images, containers, and volumes run the following command

```

1 xyz@localhost:~$ sudo rm -rf /var/lib/docker
```

You must delete the user created configuration files manually.

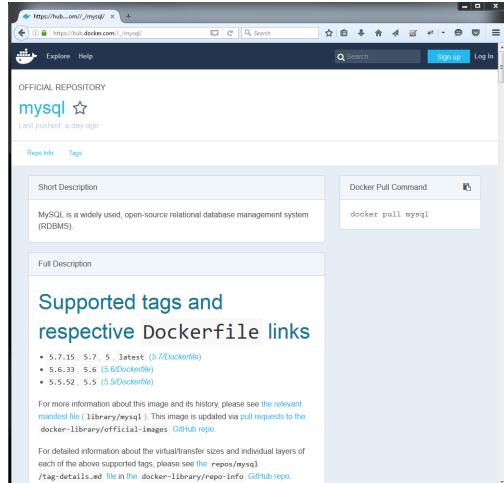
CHAPTER 11

MYSQL ON DOCKER

11.1 MySQL official image

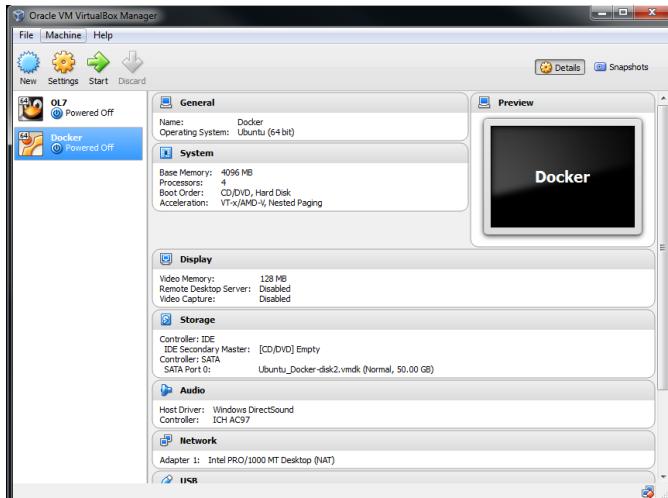
Official MySQL image:

https://hub.docker.com/_/mysql/

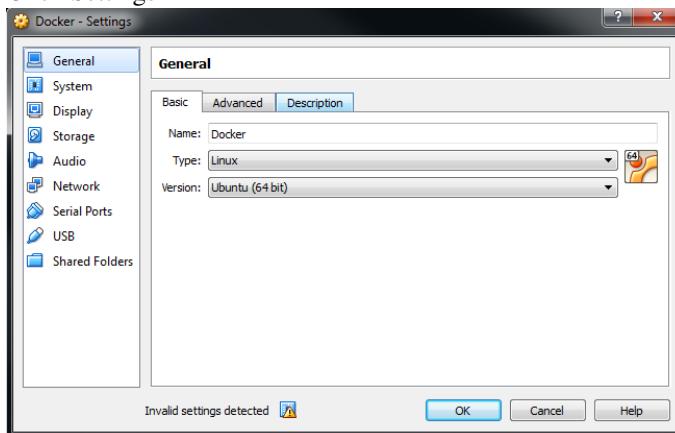


[†]this chapter is running on ubuntu linux vm, for both windows and mac

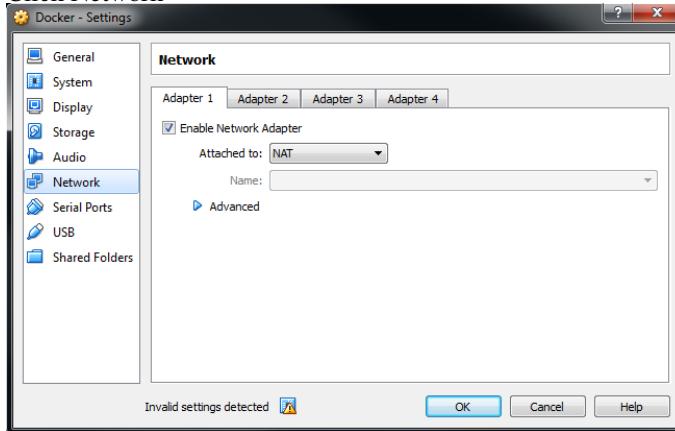
11.2 MySQL Port Forwarding



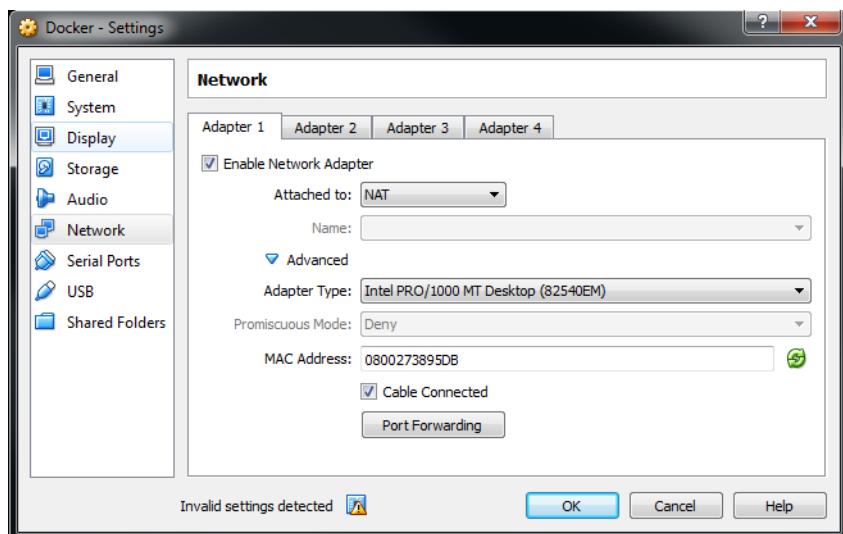
Click Settings



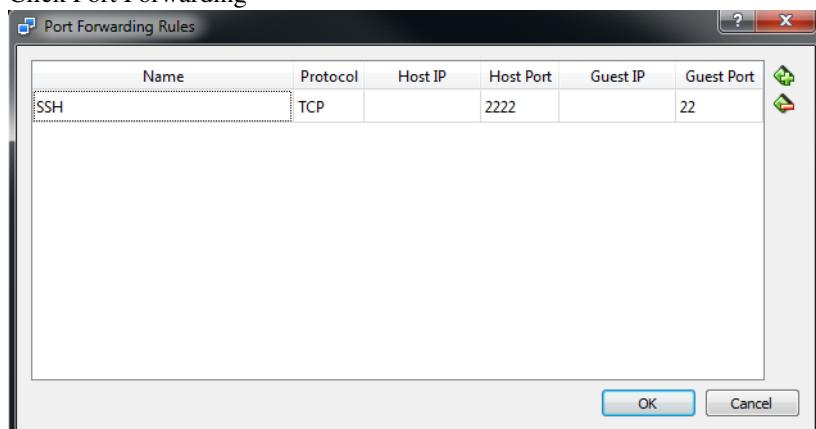
Click Network



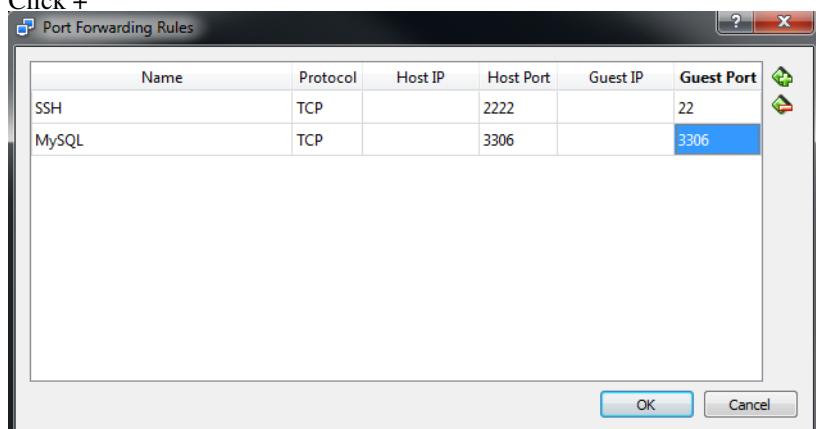
Expand Advanced



Click Port Forwarding



Click +



Add MySQL/TCP/Host Port: 3306/Guest Port: 3306

Click OK, OK

11.3 Run MySQL

- Start Docker VM and login
- Create persistent data directory first

```

1 xyz@localhost:~$ pwd
2 /home/xyz
3 xyz@localhost:~$ mkdir -p mysql/data
4 xyz@localhost:~$
```

- Here is the Docker run command looks like

```

1 $ docker run \
2 --detach \
3 --name=mysql-docker \
4 --env="MYSQL_ROOT_PASSWORD=mypassword" \
5 --publish 3306:3306 \
6 --volume=/home/xyz/mysql/data:/var/lib/mysql \
7 mysql
```

- Run MySQL

```

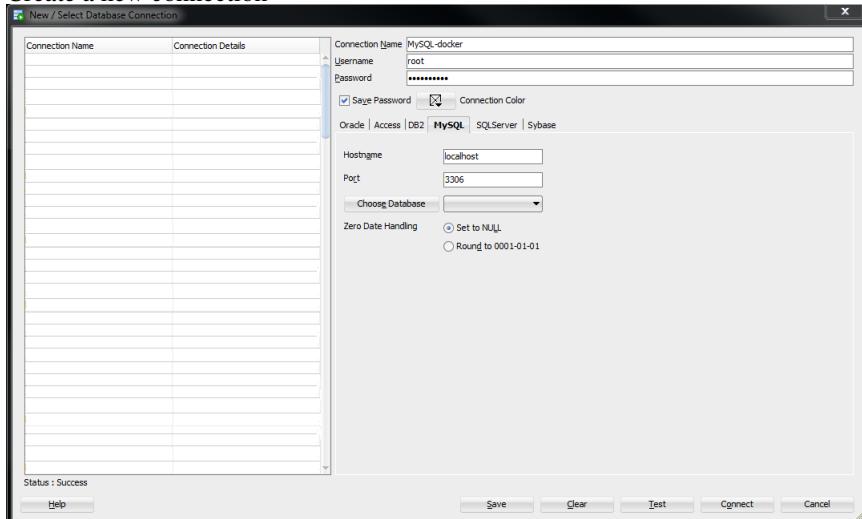
1 xyz@localhost:~$ docker run -d --name=mysql-docker -e
  ↳ MYSQL_ROOT_PASSWORD=mypassword -p 3306:3306 -v
  ↳ /home/xyz/mysql/data:/var/lib/mysql mysql
2 Unable to find image 'mysql:latest' locally
3 latest: Pulling from library/mysql
4
5 6a5a5368e0c2: Pull complete
6 0689904e86f0: Pull complete
7 486087a8071d: Pull complete
8 3eff318f6785: Pull complete
9 3df41d8a4cfb: Pull complete
10 1b4a00485931: Pull complete
11 0bab0b2c2630: Pull complete
12 264fc9ce512d: Pull complete
13 e0181dcdbbe8: Pull complete
14 53b082fa47c7: Pull complete
15 e5cf4fe00c4c: Pull complete
16 Digest: sha256:966490bda4576655dc940923c4883db68cca0b3607920be5efff7514e0379aa7
17 Status: Downloaded newer image for mysql:latest
18 a67aad225aa41f4b761bb0fc7b720040af7d8ade2405811891b6bfc6faf37a23
19 xyz@localhost:~$
```

- Check MySQL

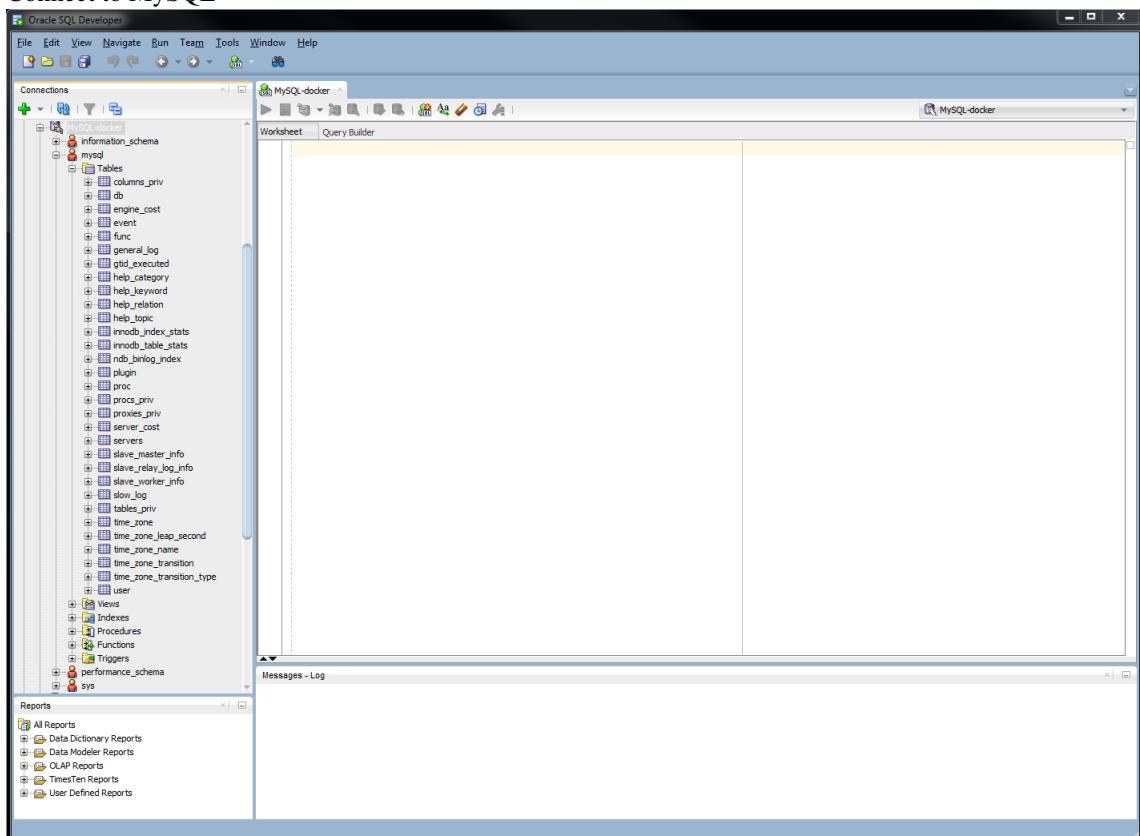
REPOSITORY	TAG	IMAGE ID	CREATED
mysql	latest	18f13d72f7f0	4 days ago
hello-world	latest	c54a2cc56ccb	12 weeks ago
xyz@localhost:~\$			
CONTAINER ID	IMAGE	COMMAND	CREATED
a67aad225aa4	mysql	"docker-entrypoint.sh"	32 seconds ago
2006d5961e6f	hello-world	"/hello"	7 weeks ago
90f2f945cb04	hello-world	"./hello"	7 weeks ago
xyz@localhost:~\$			

11.4 Connect to MySQL

- In your host machine, launch SQL Developer
- Create a new connection



- Connect to MySQL



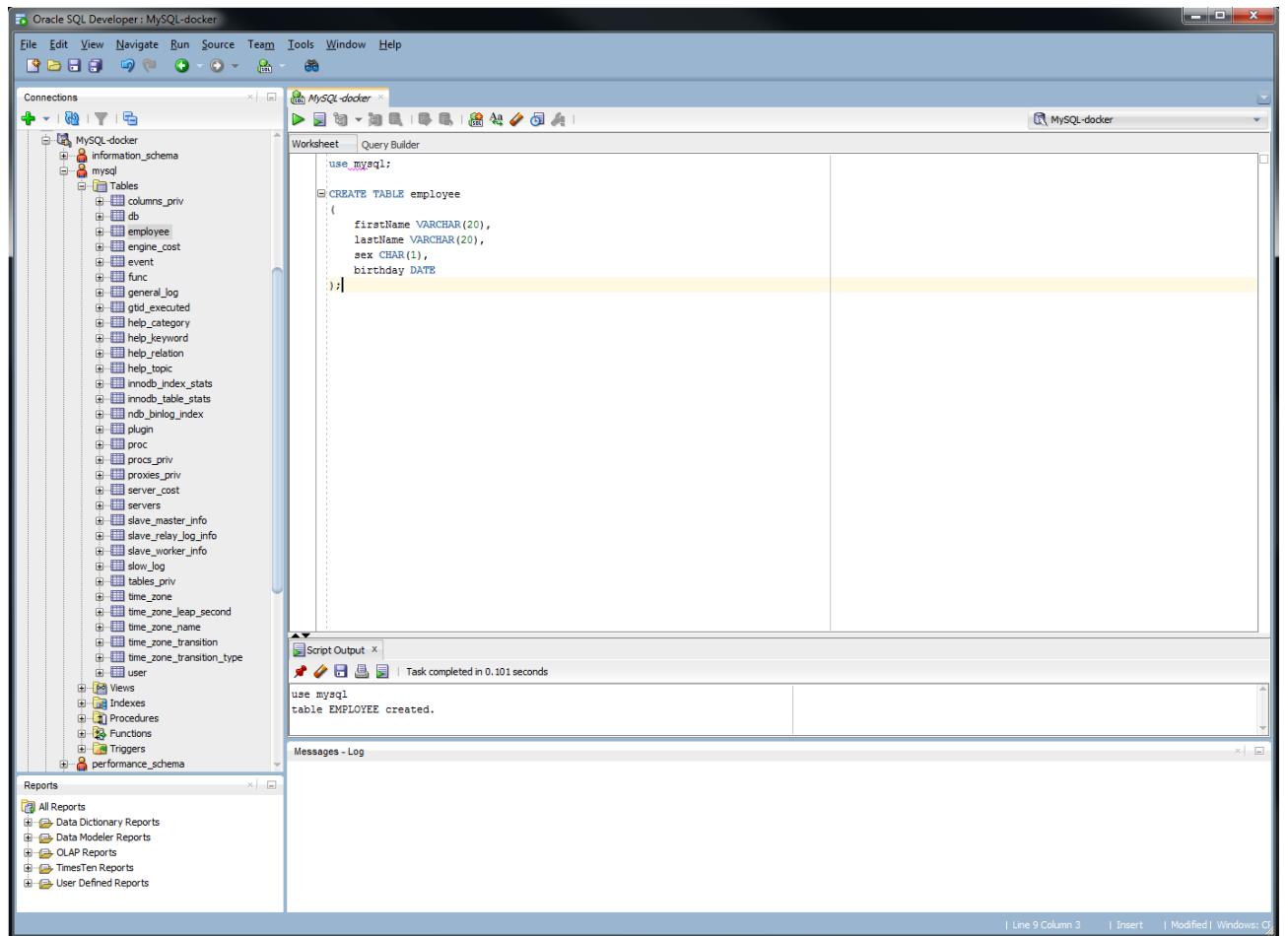
- Let's create a table

```

1 use mysql;
2
3 CREATE TABLE employee
4 (
5     firstName VARCHAR(20),
6     lastName VARCHAR(20),
7     sex CHAR(1),
8     birthday DATE
9 );

```

- Check the table is there



11.5 Remove MySQL Container

- login to Docker VM
- Stop and delete MySQL container

```

1 xyz@localhost:~$ docker ps -a
2 CONTAINER ID        IMAGE               COMMAND                  CREATED             NAMES
   ↗ STATUS

```

```

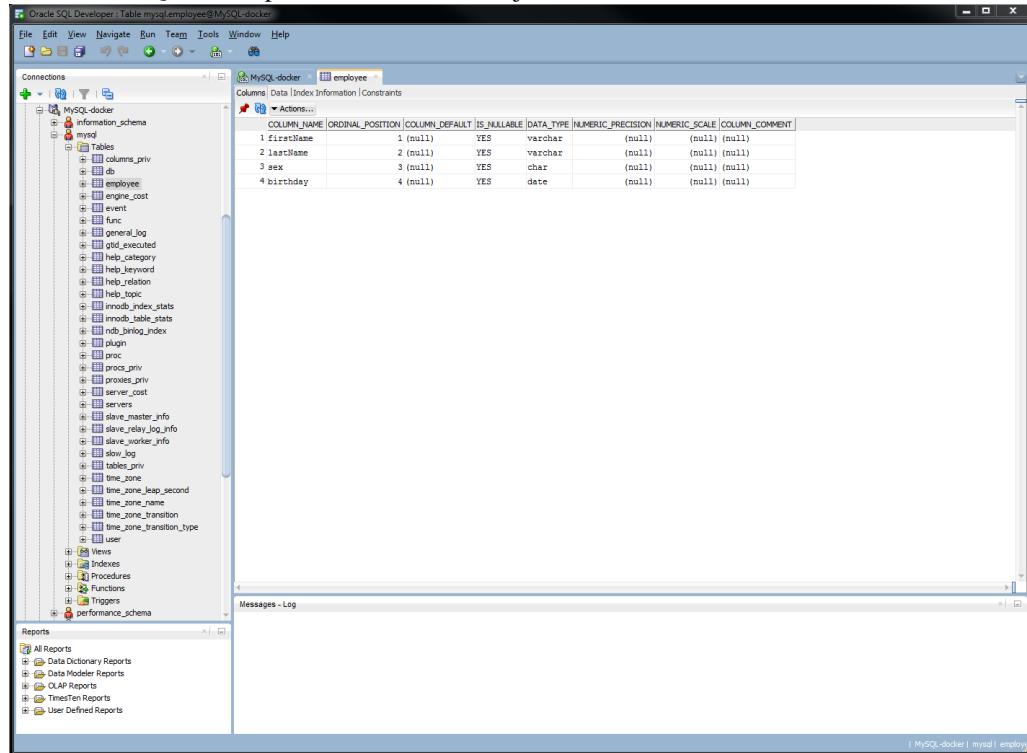
3  a67aad225aa4      mysql          "docker-entrypoint.sh"   35 minutes ago
   & Up 35 minutes           0.0.0.0:3306->3306/tcp  mysql-docker
4  2006d5961e6f      hello-world    "/hello"                7 weeks ago
   & Exited (0) 7 weeks ago                         pensive_curran
5  90f2f945cb04      hello-world    "/hello"                7 weeks ago
   & Exited (0) 7 weeks ago                         jolly_chandrasekhar
6 xyz@localhost:~$ docker stop mysql-docker
mysql-docker
xyz@localhost:~$ docker rm mysql-docker
mysql-docker
xyz@localhost:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            NAMES
   & STATUS              PORTS
12 2006d5961e6f      hello-world    "/hello"           7 weeks ago       pensive_curran
13 90f2f945cb04      hello-world    "/hello"           7 weeks ago       jolly_chandrasekhar
xyz@localhost:~$
```

▪ Run MySQL again

```

1 xyz@localhost:~$ docker run -d --name=mysql-docker -p 3306:3306 -v
   & /home/xyz/mysql/data:/var/lib/mysql mysql
2 03e1214cab3d7547e6ee5a989c35a032f08006f10b553046f354cd36a60d2b21
3 xyz@localhost:~$ docker ps -a
4 CONTAINER ID        IMAGE               COMMAND             CREATED            NAMES
   & STATUS              PORTS
5 03e1214cab3d      mysql          "docker-entrypoint.sh"   5 seconds ago
   & Up 4 seconds         0.0.0.0:3306->3306/tcp  mysql-docker
6 2006d5961e6f      hello-world    "/hello"           7 weeks ago       pensive_curran
7 90f2f945cb04      hello-world    "/hello"           7 weeks ago       jolly_chandrasekhar
xyz@localhost:~$
```

▪ Go back to SQL Developer, check the table we just created, it is still there!



PART IV

DATABASE

CHAPTER 12

DATABASE

12.1 Oracle XE - Windows

12.1.1 Download OracleXE

- Download Oracle Database Express Edition 11g Release 2 for Windows x64 (OracleXE112_Win64.zip) from:
<http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>

12.1.2 Install OracleXE

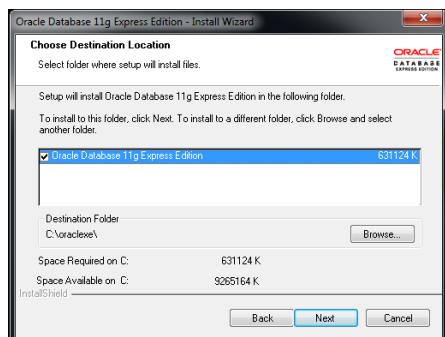
- Installing OracleXE on Microsoft Windows
Extracting OracleXE112_Win64.zip to a temporary directory, go to that directory, DISK1, right click setup.exe, Run as administrator



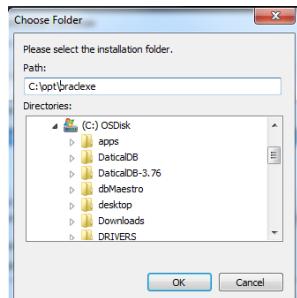
Click Next



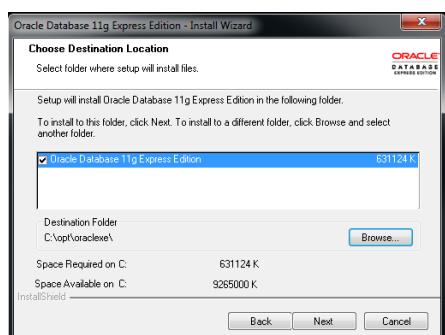
Select I accept the terms in the license agreement, and click Next



Default installation directory is C:\oraclexe, we want to change it, click Browse..



Type in C:\opt\oraclexe in the Path, then click OK

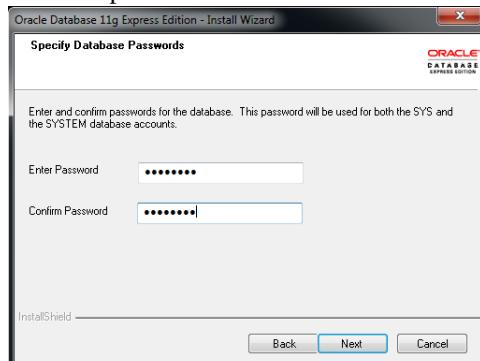


Click Next

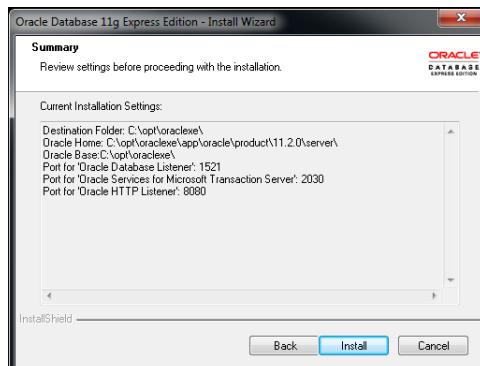
If you are prompted for a port number, then specify one.
 The following port numbers are the default values:

-
- 1 1521: Oracle database listener
 - 2 2030: Oracle Services for Microsoft Transaction Server
 - 3 8080: HTTP port for the Oracle Database XE graphical user interface
-

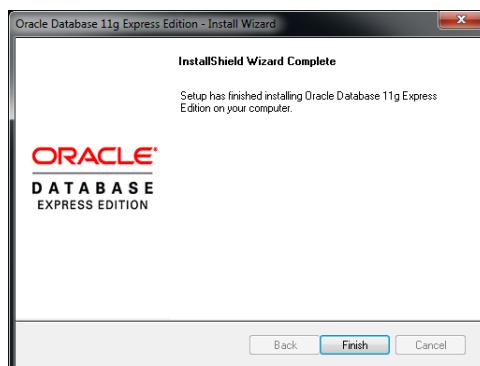
If these port numbers are not currently used, then the installation uses them automatically without prompting you. If they are in use, then you will be prompted to enter an available port number.



Type in password (oraclexe), then click Next



Click Install



Click Finish

12.1.3 Making Oracle Database XE Available to Remote Clients

- After you install Oracle Database XE, the Get Started With Oracle Database 11g Express Edition home page is only available from the local server, not remotely. To use the SQL Command Line, follow these steps:

```

1 c:\>sqlplus system
2
3 SQL*Plus: Release 11.2.0.2.0 Production on Tue Nov 3 10:29:44 2015
4
5 Copyright (c) 1982, 2014, Oracle. All rights reserved.
6
7 Enter password:
8
9 Connected to:
10 Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
11
12 SQL> EXEC DBMS_XDB.SETLISTENERLOCALACCESS(FALSE);
13
14 PL/SQL procedure successfully completed.
15
16 SQL>

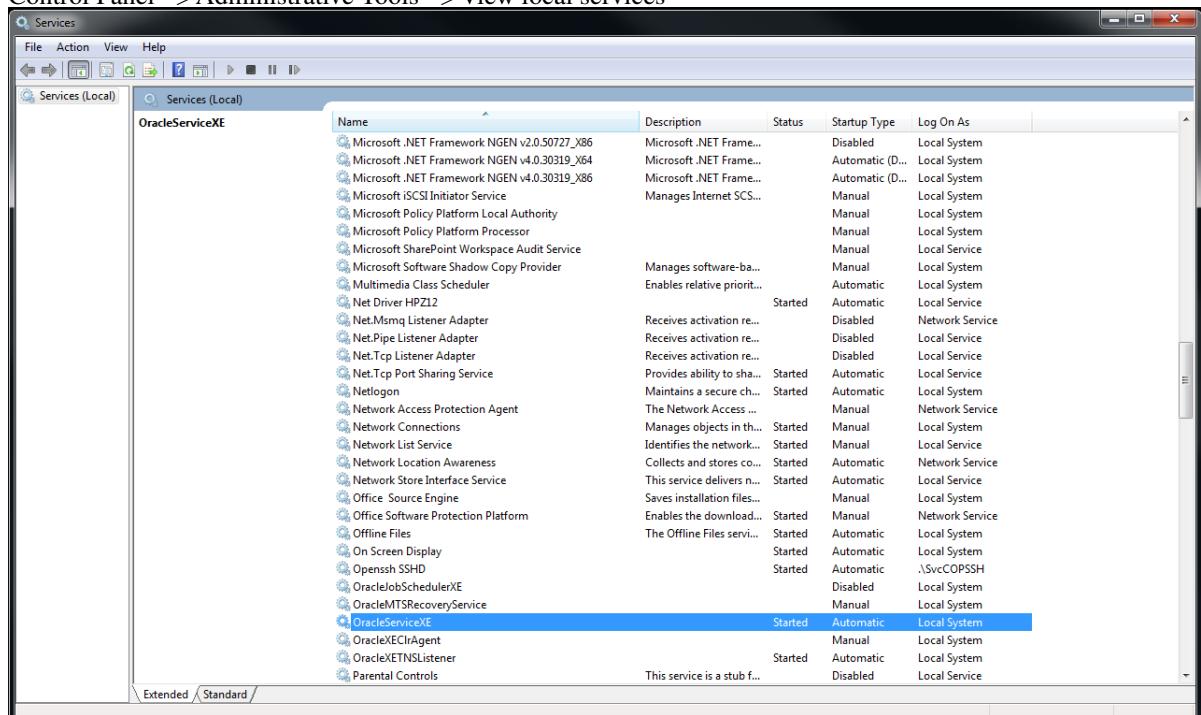
```

12.1.4 Start Oracle XE

- After you have installed Oracle Database XE, the database is up and running and you can begin using it right away.

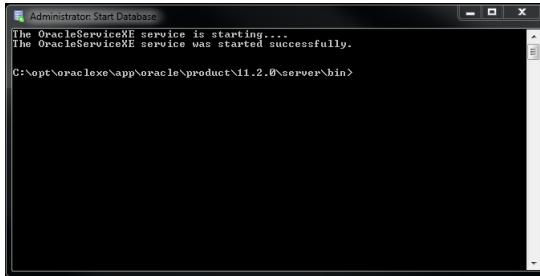
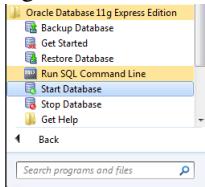
You can check Oracle XE is running or not:

Control Panel → Administrative Tools → View local services



You can see OracleServiceXE and OracleXETNLSListener are Started.

- If the database is currently stopped, start it as follows: from the Start menu, select Programs (or All Programs), then Oracle Database 11g Express Edition, and then right click Start Database, Run as administrator.



12.1.5 Connect to Oracle XE

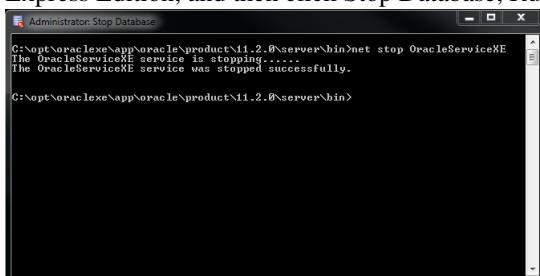
- from the Start menu, select Programs (or All Programs), then Oracle Database 11g Express Edition, and then right click Run SQL Command Line.

```

1 SQL*Plus: Release 11.2.0.2.0 Production on Mon Nov 2 16:29:11 2015
2
3 Copyright (c) 1982, 2014, Oracle. All rights reserved.
4
5 SQL> connect
6 Enter user-name: system
7 Enter password:
8 Connected.
9 SQL> select * from dual;
10
11 D
12 -
13 X
14
15 SQL>
```

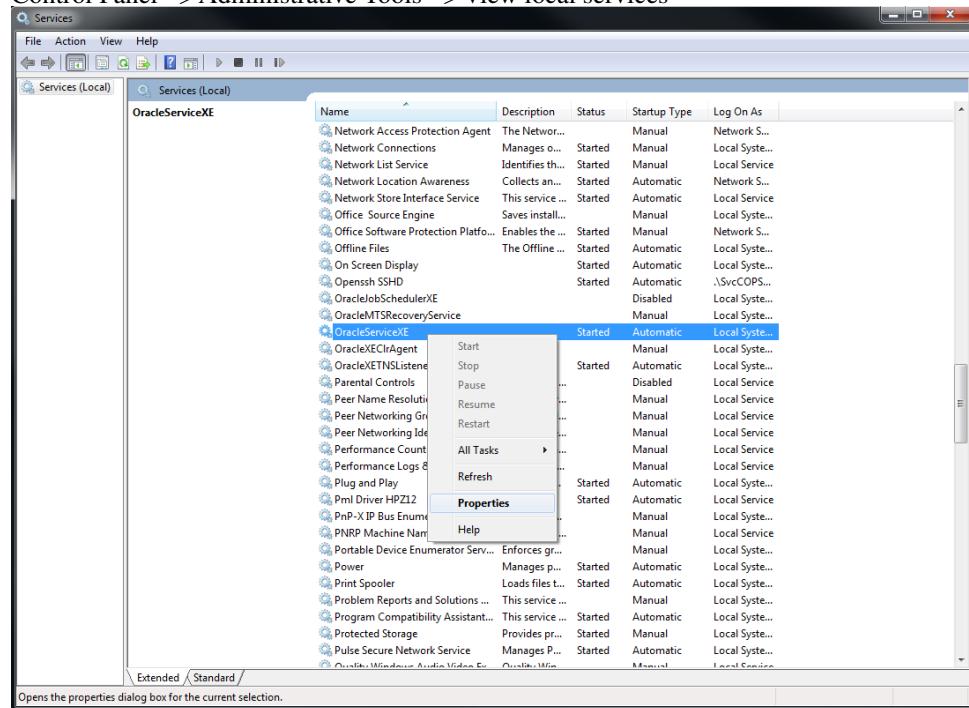
12.1.6 Stop Oracle XE

- from the Start menu, select Programs (or All Programs), then Oracle Database 11g Express Edition, and then click Stop Database, Run as administrator.



12.1.7 Stop Oracle XE Service

- Control Panel -> Administrative Tools -> View local services



Right click on OracleServiceXE, select Properties, then change Startup type from Automatic to Manual.

Same for OracleXETNSListener

- You can also change it from command line as administrator:

```

1 net stop OracleXETNSListener
2 net stop OracleServiceXE
3 sc config OracleXETNSListener start= demand
4 sc config OracleServiceXE start= demand

```

Result:

```

1 c:\>net stop OracleXETNSListener
2 The OracleXETNSListener service is stopping.
3 The OracleXETNSListener service was stopped successfully.
4
5
6 c:\>net stop OracleServiceXE
7 The OracleServiceXE service is stopping....
8 The OracleServiceXE service was stopped successfully.
9
10
11 c:\>sc config OracleXETNSListener start= demand
12 [SC] ChangeServiceConfig SUCCESS
13
14 c:\>sc config OracleServiceXE start= demand
15 [SC] ChangeServiceConfig SUCCESS
16
17 c:\>

```

12.1.8 Changing Listener Port Numbers

You would need to change a default listener port number only if there were a port number conflict with another TCP/IP service. You are given the opportunity to change listener port numbers during installation (Windows) or configuration (Linux). This section explains how to change port numbers after installation or configuration.

12.1.8.1 Changing the Listener Port Number for Database Connection Requests

- If you change the listener port number for database connection requests, you must ensure that all future database connection requests use the new port number. This means that connection requests such as those "Connecting Remotely with SQL Command Line" must explicitly include the port number.
- For example, if you change the port number for database connection requests to 1522, subsequent SQL Command Line (SQL*Plus) connect statements must be similar to the following (assuming a connection from Oracle Database Express Edition Client):

```
connect system/mypassword@myhost.mydomain.com:1522
```

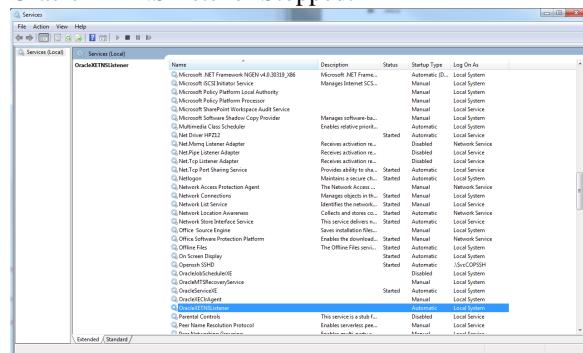
- Assume that your Oracle Database XE host computer is named myhost.mydomain.com and that you want to install a new software package on this computer that requires TCP port number 1521. Assume also that the port number for that software package cannot be configured, and that you must therefore resolve the port number conflict by reconfiguring Oracle Database XE. You decide to change the listener port number for database connection requests to 1522.
- To change the listener port number for database connection requests to 1522:

1. Stop the listener.

Right click Command Prompt → Run as administrator

```
c:\>lsnrctl stop
2
3 LSNRCTL for 64-bit Windows: Version 11.2.0.2.0 - Production on 03-NOV-2015
   → 14:50:53
4
5 Copyright (c) 1991, 2014, Oracle. All rights reserved.
6
7 Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC1)))
8 The command completed successfully
9
10 c:\>
```

You can see from Control Panel → Administrative Tools → View local services OracleXETNSListener Stopped.



2. Modify listener.ora file.

Locate listener.ora file.

It is under \${oracleInstallDirectory}/app/oracle/product/\${oracleVersion}/server/network/admin/

In our case, it is C:/opt/oracle/app/oracle/product/11.2.0/server/network/ADMIN

Change listener port 1521 to 1522. (line 19 below)

```

1 SID_LIST_LISTENER =
2   (SID_LIST =
3     (SID_DESC =
4       (SID_NAME = PLSExtProc)
5       (ORACLE_HOME = C:/opt/oracle/app/oracle/product/11.2.0/server)
6       (PROGRAM = extproc)
7     )
8     (SID_DESC =
9       (SID_NAME = CLRExtProc)
10      (ORACLE_HOME = C:/opt/oracle/app/oracle/product/11.2.0/server)
11      (PROGRAM = extproc)
12    )
13  )
14
15 LISTENER =
16   (DESCRIPTION_LIST =
17     (DESCRIPTION =
18       (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
19       (ADDRESS = (PROTOCOL = TCP) (HOST = myhost.mydomain.com) (PORT = 1522) )
20     )
21   )
22
23 DEFAULT_SERVICE_LISTENER = (XE)

```

3. Start the listener.

```

1 c:\>lsnrctl start
2
3 LSNRCTL for 64-bit Windows: Version 11.2.0.2.0 - Production on 03-NOV-2015
4   ↵ 16:04:00
5 Copyright (c) 1991, 2014, Oracle. All rights reserved.
6
7 Starting tnslsnr: please wait...
8
9 TNS-12560: TNS:protocol adapter error
10    TNS-00530: Protocol adapter error
11
12
13 c:\>

```

Let's fix it.

Locate tnsnames.ora.

It is under \${oracleInstallDirectory}/app/oracle/product/\${oracleVersion}/server/network/admin/

In our case, it is C:/opt/oracle/app/oracle/product/11.2.0/server/network/ADMIN

Change listener port 1521 to 1522. (line 3 below)

```

1 XE =
2   (DESCRIPTION =
3     (ADDRESS = (PROTOCOL = TCP) (HOST = myhost.mydomain.com) (PORT = 1522) )
4     (CONNECT_DATA =
5       (SERVER = DEDICATED)
6       (SERVICE_NAME = XE)
7     )
8   )
9
10 EXTPROC_CONNECTION_DATA =

```

```

11 (DESCRIPTION =
12   (ADDRESS_LIST =
13     (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
14   )
15   (CONNECT_DATA =
16     (SID = PLSExtProc)
17     (PRESENTATION = RO)
18   )
19 )
20
21 ORACLR_CONNECTION_DATA =
22 (DESCRIPTION =
23   (ADDRESS_LIST =
24     (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
25   )
26   (CONNECT_DATA =
27     (SID = CLRExtProc)
28     (PRESENTATION = RO)
29   )
30 )

```

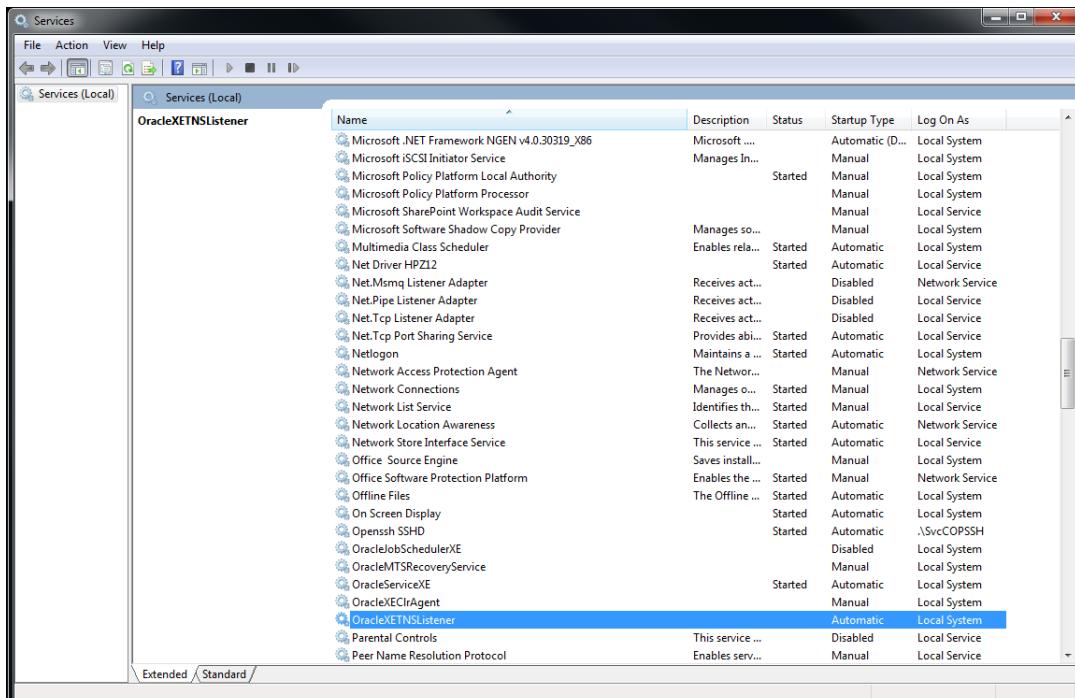
Start listener.

```

1 c:\>net start OracleXETNSListener
2 The OracleXETNSListener service is starting.
3 The OracleXETNSListener service was started successfully.
4
5
6 c:\>

```

You can see from Control Panel → Administrative Tools → View local services
OracleXETNSListener Started.



4. Start SQL Command Line and connect to the database as user SYSTEM.

```

1 c:\>sqlplus system
2
3 SQL*Plus: Release 11.2.0.2.0 Production on Tue Nov 3 16:47:29 2015
4
5 Copyright (c) 1982, 2014, Oracle. All rights reserved.
6
7 Enter password:
8
9 Connected to:
10 Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
11
12 SQL> ALTER SYSTEM SET LOCAL_LISTENER =
13   ↪ "(ADDRESS=(PROTOCOL=TCP) (HOST=myhost.mydomain.com) (PORT=1522))";
14 System altered.
15
16 SQL> ALTER SYSTEM REGISTER;
17
18 System altered.
19
20 SQL>

```

5. Exit SQL Command Line and run the lsnrctl status command to verify the port number change.

```

1 c:\>lsnrctl status
2
3 LSNRCTL for 64-bit Windows: Version 11.2.0.2.0 - Production on 03-NOV-2015
4   ↪ 17:25:49
5
6 Copyright (c) 1991, 2014, Oracle. All rights reserved.
7
8 Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC1)))
9 STATUS of the LISTENER
10 -----
11 Alias           LISTENER
12 Version        TNSLSNR for 64-bit Windows: Version 11.2.0.2.0 -
13   ↪ Production
14 Start Date     03-NOV-2015 17:09:32
15 Uptime          0 days 0 hr. 16 min. 16 sec
16 Trace Level    off
17 Security        ON: Local OS Authentication
18 SNMP            OFF
19 Default Service XE
20 Listener Parameter File
21   ↪ C:\opt\oraclexe\app\oracle\product\11.2.0\server\network\admin\listener.ora
22 Listener Log File
23   ↪ C:\opt\oraclexe\app\oracle\diag\tnslsnr\myhost\listener\alert\log.xml
24 Listening Endpoints Summary...
25   (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (PIPENAME=\\.\\pipe\\EXTPROC1 ipc)))
26   (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=myhost.mydomain.com) (PORT=1522)))
27 Services Summary...
28 Service "CLRExtProc" has 1 instance(s).
29   Instance "CLRExtProc", status UNKNOWN, has 1 handler(s) for this service...
30 Service "PLSExtProc" has 1 instance(s).
31   Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
32 Service "XEXDB" has 1 instance(s).
33   Instance "xe", status READY, has 1 handler(s) for this service...
34 Service "xe" has 1 instance(s).
35   Instance "xe", status READY, has 1 handler(s) for this service...
36 The command completed successfully
37
38 c:\>

```

12.1.8.2 Changing the Listener Port Number for HTTP Connection Requests

- If you change the listener port number for HTTP connection requests, you must ensure that all future HTTP connection requests use the new port number.
- Start SQL Command Line and connect to the database as user SYSTEM.

```

1 c:\>sqlplus system
2
3 SQL*Plus: Release 11.2.0.2.0 Production on Tue Nov 3 17:33:17 2015
4
5 Copyright (c) 1982, 2014, Oracle. All rights reserved.
6
7 Enter password:
8
9 Connected to:
10 Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
11
12 SQL> EXEC DBMS_XDB.SETHTTPPORT(8180);
13
14 PL/SQL procedure successfully completed.
15
16 SQL>

```

- Exit SQL Command Line (by entering the exit command) and view listener status to verify the port number change.

```

1 c:\>lsnrctl status
2
3 LSNRCTL for 64-bit Windows: Version 11.2.0.2.0 - Production on 03-NOV-2015
   → 17:35:29
4
5 Copyright (c) 1991, 2014, Oracle. All rights reserved.
6
7 Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC1)))
8 STATUS of the LISTENER
9 -----
10 Alias          LISTENER
11 Version        TNSLSNR for 64-bit Windows: Version 11.2.0.2.0 -
   → Production
12 Start Date    03-NOV-2015 17:09:32
13 Uptime         0 days 0 hr. 25 min. 56 sec
14 Trace Level   off
15 Security       ON: Local OS Authentication
16 SNMP           OFF
17 Default Service XE
18 Listener Parameter File
   → C:\oraclexe\app\oracle\product\11.2.0\server\network\admin\listener.ora
19 Listener Log File
   → C:\oraclexe\app\oracle\diag\tnslsnr\myhost\listener\alert\log.xml
20 Listening Endpoints Summary...
21   (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (PIPENAME=\\.\\pipe\\EXTPROClipc)))
22   (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=myhost.mydomain.com) (PORT=1522)))
23
   → (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=myhost.mydomain.com) (PORT=8180)) (Presentation=HTTP) (Session=RAW))
24 Services Summary...
25 Service "CLRExtProc" has 1 instance(s).
26   Instance "CLRExtProc", status UNKNOWN, has 1 handler(s) for this service...
27 Service "PLSExtProc" has 1 instance(s).
28   Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
29 Service "XEADB" has 1 instance(s).
30   Instance "xe", status READY, has 1 handler(s) for this service...
31 Service "xe" has 1 instance(s).
32   Instance "xe", status READY, has 1 handler(s) for this service...
33 The command completed successfully
34
35 c:\>

```

12.2 MySQL

12.2.1 MySQL - Windows

12.2.1.1 Download MySQL

- Download MySQL Community Server, Windows (x86, 64-bit), ZIP Archive (mysql-5.6.26-winx64.zip) from:
<http://dev.mysql.com/downloads/mysql/>

12.2.1.2 Install MySQL

- Installing MySQL on Microsoft Windows Using a noinstall Zip Archive
 Extracting the Install Archive mysql-5.6.26-winx64.zip to C:\opt
 It will create a directory mysql-5.6.26-winx64
 C:\opt\mysql-5.6.26-winx64
 Rename directory mysql-5.6.26-winx64 to mysql
 Now we have C:\opt\mysql\bin
- Add mysql bin directory to Path
 Add C:\opt\mysql\bin to Windows Path.
 PATH=C:\opt\mysql\bin;%PATH%
- Creating an Option File
 Create a text file C:\opt\mysql\my.ini

```

1 [mysqld]
2 # set basedir to your installation path
3 basedir=C:/opt/mysql
4 # set datadir to the location of your data directory
5 datadir=C:/opt/mysql/data

```

12.2.1.3 Start MySQL

- Starting the Server for the First Time

```
C:\> mysqld --console
```

12.2.1.4 Stop MySQL

- Stop the Server

```
C:\> mysqladmin.exe -u root shutdown
```

12.2.1.5 Testing The MySQL Installation

- You can test whether the MySQL server is working by executing any of the following commands:
- mysqlshow

```
C:\> mysqlshow
```

Result:

```

1 c:\>mysqlshow
2 +-----+
3 | Databases      |
4 +-----+
5 | information_schema |
6 | test           |
7 +-----+
8
9 c:\>

```

▪ mysqlshow -u root mysql

```
C:\> mysqlshow -u root mysql
```

Result:

```

1 c:\>mysqlshow -u root mysql
2 Database: mysql
3 +-----+
4 |       Tables      |
5 +-----+
6 | columns_priv     |
7 | db               |
8 | event            |
9 | func              |
10 | general_log      |
11 | help_category    |
12 | help_keyword     |
13 | help_relation    |
14 | help_topic        |
15 | innodb_index_stats |
16 | innodb_table_stats |
17 | ndb_binlog_index  |
18 | plugin            |
19 | proc              |
20 | procs_priv        |
21 | proxies_priv      |
22 | servers            |
23 | slave_master_info |
24 | slave_relay_log_info |
25 | slave_worker_info  |
26 | slow_log           |
27 | tables_priv        |
28 | time_zone          |
29 | time_zone_leap_second |
30 | time_zone_name     |
31 | time_zone_transition |
32 | time_zone_transition_type |
33 | user               |
34 +-----+
35
36 c:\>
```

▪ mysqladmin version status proc

```
C:\> mysqladmin version status proc
```

Result:

```

1 c:\>mysqladmin version status proc
2 mysqladmin Ver 8.42 Distrib 5.6.26, for Win64 on x86_64
3 Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.
4
5 Oracle is a registered trademark of Oracle Corporation and/or its
6 affiliates. Other names may be trademarks of their respective
7 owners.
```

```

8
9  Server version          5.6.26
10 Protocol version        10
11 Connection              localhost via TCP/IP
12 TCP port                 3306
13 Uptime:                13 min 32 sec
14
15 Threads: 1  Questions: 7  Slow queries: 0  Opens: 67  Flush tables: 1  Open
     →  tables: 60  Queries per second avg: 0.008
16 Uptime: 812  Threads: 1  Questions: 8  Slow queries: 0  Opens: 67  Flush tables:
17   1  Open tables: 60  Queries per second avg: 0.009
18 mysqladmin: process list failed; error: 'Access denied; you need (at least one
     →  of) the PROCESS privilege(s) for this operation'
19
20 c:>

```

- **mysql test**

```
C:> mysql test
```

Result:

```

1 c:>mysql test
2 Welcome to the MySQL monitor. Commands end with ; or \g.
3 Your MySQL connection id is 4
4 Server version: 5.6.26 MySQL Community Server (GPL)
5
6 Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.
7
8 Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
     → Other names may be trademarks of their respective owners.
9
10 Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
11
12 mysql> quit
13 Bye
14
15 c:>

```

12.2.1.6 Securing the Initial MySQL Accounts

- **Start MySQL**

```
C:> mysqld --console
```

- **Connect to MySQL using root**

```
C:> mysql -u root
```

Result:

```

1 c:>mysql -u root
2 Welcome to the MySQL monitor. Commands end with ; or \g.
3 Your MySQL connection id is 1
4 Server version: 5.6.26 MySQL Community Server (GPL)
5
6 Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.
7
8 Oracle is a registered trademark of Oracle Corporation and/or its
9 affiliates. Other names may be trademarks of their respective
10 owners.
11
12 Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
13
14 mysql>

```

- To display which accounts exist in the mysql.user table and check whether their passwords are empty, use the following statement:

```
mysql> SELECT User, Host, Password FROM mysql.user;
```

Result:

```
1 mysql> SELECT User, Host, Password FROM mysql.user;
2 +-----+-----+
3 | User | Host      | Password |
4 +-----+-----+
5 | root | localhost |          |
6 | root | 127.0.0.1 |          |
7 | root | ::1       |          |
8 |      | localhost |          |
9 +-----+-----+
10 4 rows in set (0.00 sec)
11
12 mysql>
```

- Assigning root Account Passwords - method 1

Use the SET PASSWORD statement

To assign passwords using SET PASSWORD, connect to the server as root and issue a SET PASSWORD statement for each root account listed in the mysql.user table.

```
1 mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');
2 mysql> SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('newpwd');
3 mysql> SET PASSWORD FOR 'root'@'::1' = PASSWORD('newpwd');
4 mysql> SET PASSWORD FOR 'root'@'%` = PASSWORD('newpwd');
```

The last statement is unnecessary if the mysql.user table has no root account with a host value of %.

Result:

```
1 mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');
2 Query OK, 0 rows affected (0.00 sec)
3
4 mysql> SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('newpwd');
5 Query OK, 0 rows affected (0.00 sec)
6
7 mysql> SET PASSWORD FOR 'root'@'::1' = PASSWORD('newpwd');
8 Query OK, 0 rows affected (0.00 sec)
9
10 mysql> SET PASSWORD FOR 'root'@'%` = PASSWORD('newpwd');
11 ERROR 1133 (42000): Can't find any matching row in the user table
12 mysql> SELECT User, Host, Password FROM mysql.user;
13 +-----+-----+
14 | User | Host      | Password           |
15 +-----+-----+
16 | root | localhost | *1FA85AA204CC12B39B20E8F1E839D11B3F9E6AA4 |
17 | root | 127.0.0.1 | *1FA85AA204CC12B39B20E8F1E839D11B3F9E6AA4 |
18 | root | ::1       | *1FA85AA204CC12B39B20E8F1E839D11B3F9E6AA4 |
19 |      | localhost |          |
20 +-----+-----+
21 4 rows in set (0.00 sec)
22
23 mysql>
```

- Assigning root Account Passwords - method 2

Use the UPDATE statement

You can also use a single statement that assigns a password to all root accounts by

using UPDATE to modify the mysql.user table directly. This method works on any platform:

```

1 mysql> UPDATE mysql.user SET Password = PASSWORD('newpwd')
2   -> WHERE User = 'root';
3 mysql> FLUSH PRIVILEGES;

```

A command need not be given all on a single line, so lengthy commands that require several lines are not a problem. mysql determines where your statement ends by looking for the terminating semicolon, not by looking for the end of the input line. (In other words, mysql accepts free-format input: it collects input lines but does not execute them until it sees the semicolon.)

Result:

```

1 mysql> UPDATE mysql.user SET Password = PASSWORD('newpwd')
2   -> WHERE User = 'root';
3 Query OK, 0 rows affected (0.00 sec)
4 Rows matched: 3  Changed: 0  Warnings: 0
5
6 mysql> FLUSH PRIVILEGES;
7 Query OK, 0 rows affected (0.00 sec)
8
9 mysql> SELECT User, Host, Password FROM mysql.user;
10 +-----+-----+-----+
11 | User | Host      | Password          |
12 +-----+-----+-----+
13 | root | localhost | *1FA85AA204CC12B39B20E8F1E839D11B3F9E6AA4 |
14 | root | 127.0.0.1 | *1FA85AA204CC12B39B20E8F1E839D11B3F9E6AA4 |
15 | root | ::1       | *1FA85AA204CC12B39B20E8F1E839D11B3F9E6AA4 |
16 |      | localhost |                               |
17 +-----+-----+-----+
18 4 rows in set (0.00 sec)
19
20 mysql>

```

The FLUSH statement causes the server to reread the grant tables. Without it, the password change remains unnoticed by the server until you restart it.

- Assigning root Account Passwords - method 3

Use the mysqladmin command-line client program

To assign passwords to the root accounts using mysqladmin, execute the following commands:

```

1 c:\>mysqladmin -u root password "newpwd"
2 c:\>mysqladmin -u root -p password "newpwd"
3 c:\>mysqladmin -u root -h host_name password "newpwd"

```

Result:

```

1 c:\>mysqladmin -u root password "newpwd"
2 Warning: Using a password on the command line interface can be insecure.
3
4 c:\>mysqladmin -u root -p password "newpwd"
5 Enter password: *****
6 Warning: Using a password on the command line interface can be insecure.
7
8 c:\>

```

The mysqladmin method of setting the root account passwords does not work for the 'root'@'127.0.0.1' or 'root'@'::1' account. Use the SET PASSWORD method shown earlier.

- After the root passwords have been set, you must supply the appropriate password whenever you connect as root to the server.

For example, to shut down the server with mysqladmin, use this command:

```
1 c:\>mysqladmin -u root -p shutdown
2 Enter password: (enter root password here)
3 c:\>
```

Or

```
1 c:\>mysqladmin -u root -pnewpwd shutdown
2 Warning: Using a password on the command line interface can be insecure.
3 c:\>
```

12.2.1.7 Connecting to and Disconnecting from the Server

- Connect to the server

To connect to the server, you will usually need to provide a MySQL user name when you invoke mysql and, most likely, a password. If the server runs on a machine other than the one where you log in, you will also need to specify a host name.

```
1 c:\>mysql -h localhost -u root -p
2 or
3 c:\>mysql -h localhost -u root -pnewpwd
```

Result:

```
1 c:\>mysql -h localhost -u root -p
2 Enter password: *****
3 Welcome to the MySQL monitor. Commands end with ; or \g.
4 Your MySQL connection id is 1
5 Server version: 5.6.26 MySQL Community Server (GPL)
6
7 Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.
8
9 Oracle is a registered trademark of Oracle Corporation and/or its
10 affiliates. Other names may be trademarks of their respective
11 owners.
12
13 Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
14
15 mysql>
```

- If you are logging in on the same machine that MySQL is running on, you can omit the host, and simply use the following:

```
1 c:\>mysql -u root -p
2 or
3 c:\>mysql -u root -pnewpwd
```

- Disconnect from the server After you have connected successfully, you can disconnect any time by typing QUIT (or \q) at the mysql> prompt:

```
1 mysql> quit
2 Bye
3
4 c:\>
```

12.2.2 MySQL - Mac

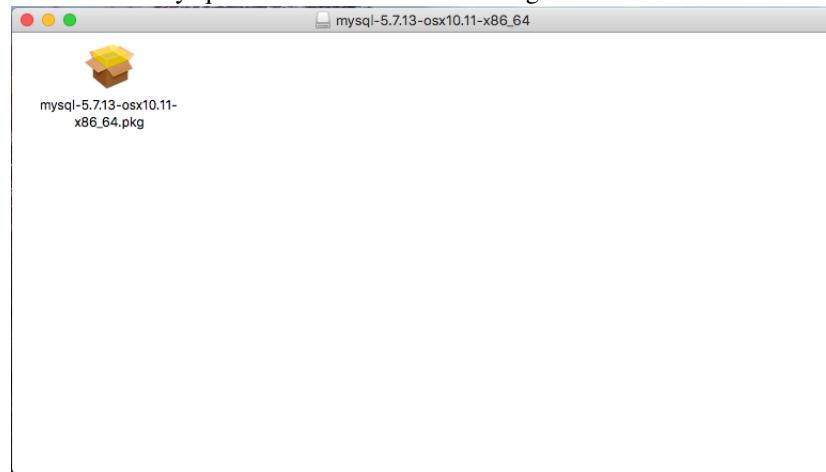
12.2.2.1 Download MySQL

- Download MySQL Community Server, Mac OS X 10.11 (x86, 64bit), DMG Archive (mysql-5.7.13-osx10.11-x86_64.dmg) from:
<http://dev.mysql.com/downloads/mysql>

12.2.2.2 Install MySQL

- Before proceeding with the installation, be sure to stop all running MySQL server instances by using either the MySQL Manager Application (on OS X Server), the preference pane, or mysqladmin shutdown on the command line.
- Install

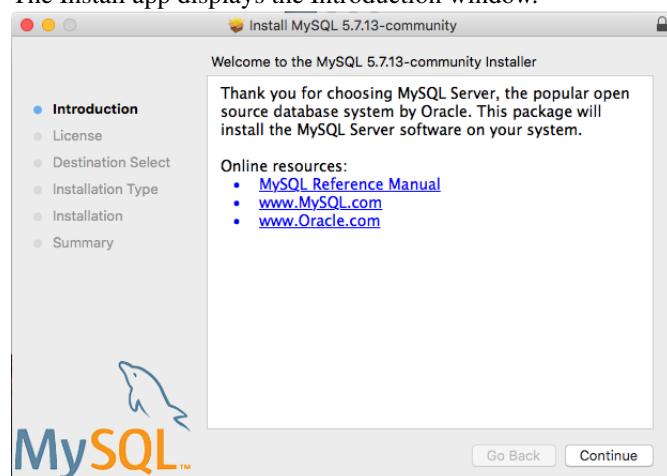
Double click mysql-5.7.13-osx10.11-x86_64.dmg file to launch it



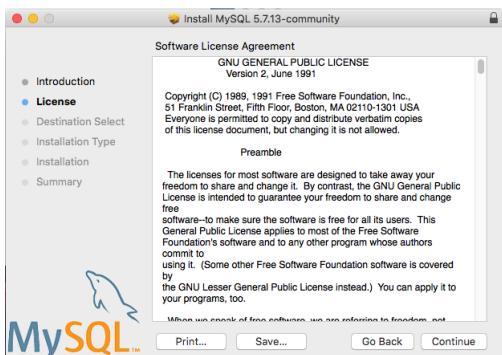
A Finder window appears containing an icon of an open box and the name of the .pkg file.

Double click the package icon to launch the Install app

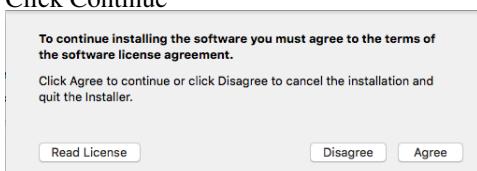
The Install app displays the Introduction window.



Click Continue

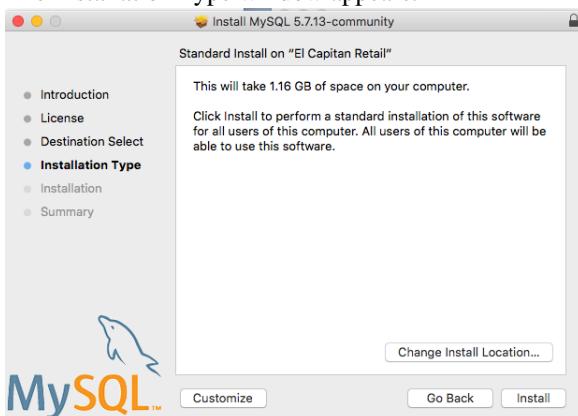


Click Continue

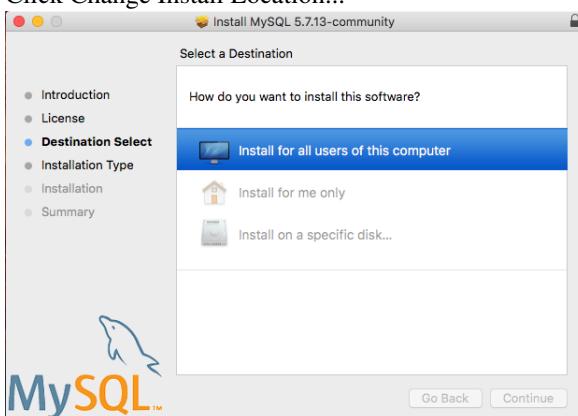


Click Agree

The Installation Type window appears.



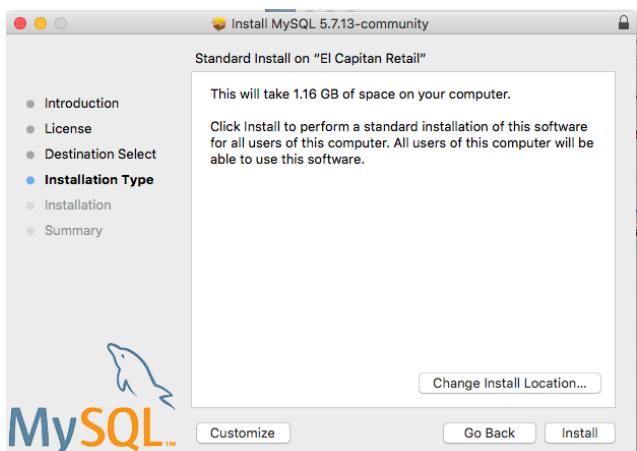
Click Change Install Location...



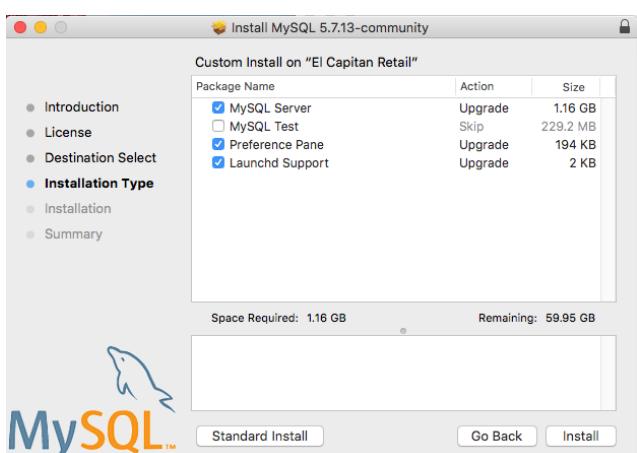
Click Install for all users of this computer



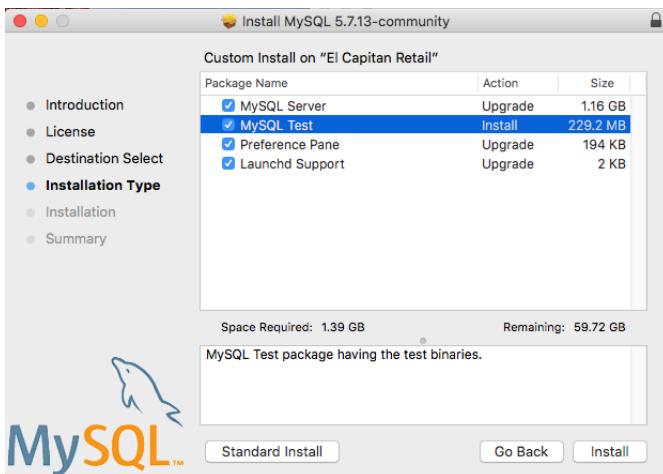
Click Continue



Click Customize

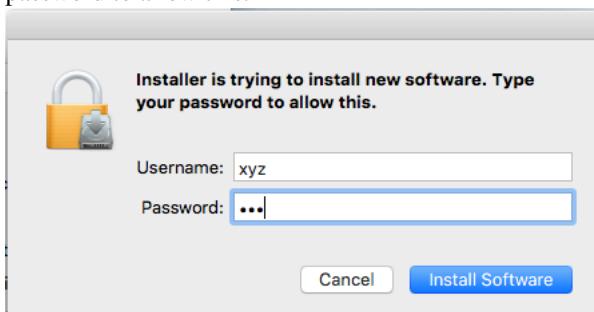


Select MySQL Test



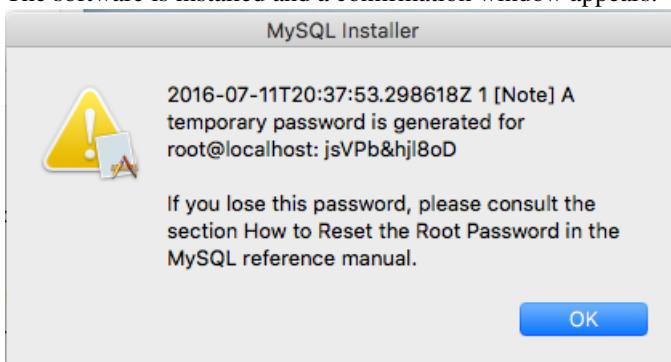
Click Install

A window appears that says "Installer is trying to install new software. Type your password to allow this."



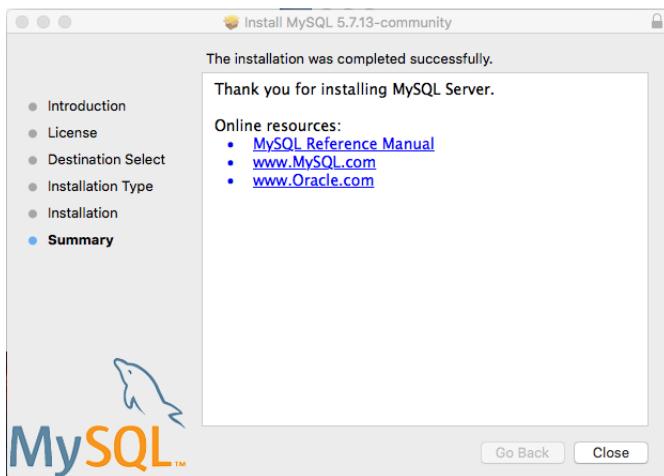
Enter the Administrator login and password and click Install Software.

The software is installed and a confirmation window appears.



Once the installation has been completed successfully, you will be provided with your temporary root password. This cannot be recovered, so you must save this password. Click OK

After logging into MySQL using this temporary password, MySQL will expire this password and require you to create a new password.

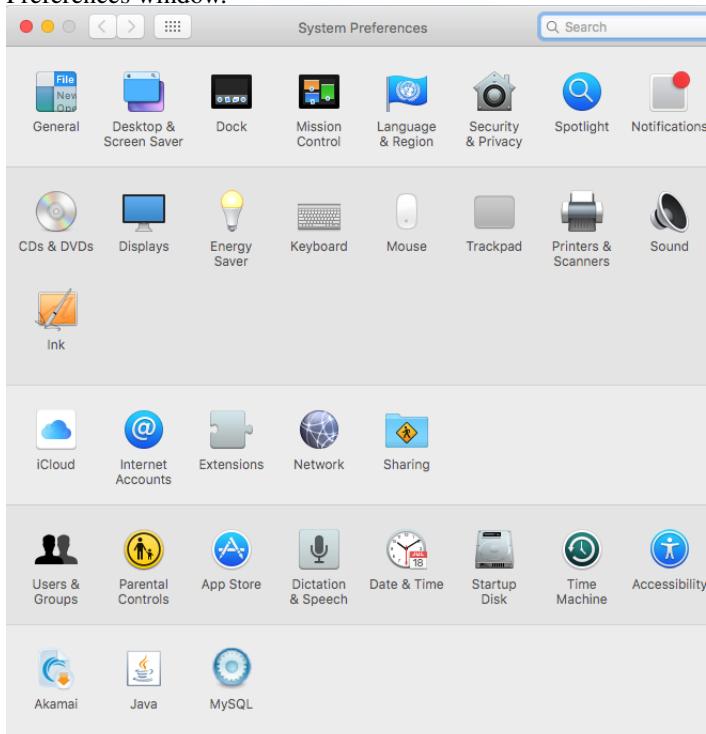


Click Close

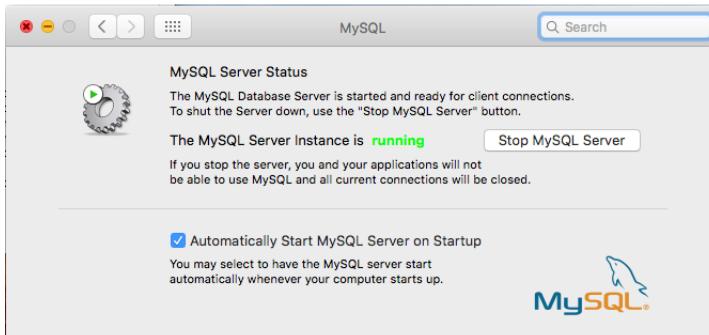
After the software is installed, delete the .dmg file if you want to save disk space.

12.2.2.3 MySQL Preference Pane

- The MySQL Installation Package includes a MySQL preference pane that enables you to start, stop, and control automated startup during boot of your MySQL installation.
- This preference pane is installed by default, and is listed under your system's System Preferences window.



Click MySQL



The MySQL Preference Pane shows the current status of the MySQL server, showing stopped (in red) if the server is not running and running (in green) if the server has already been started. The preference pane also shows the current setting for whether the MySQL server has been set to start automatically.

- To start the MySQL server using the preference pane:
Click Start MySQL Server. You may be prompted for the username and password of a user with administrator privileges to start the MySQL server.
- To stop the MySQL server using the preference pane:
Click Stop MySQL Server. You may be prompted for the username and password of a user with administrator privileges to stop the MySQL server.
- To automatically start the MySQL server when the system boots:
Check the check box next to Automatically Start MySQL Server on Startup.
- To disable automatic MySQL server startup when the system boots:
Uncheck the check box next to Automatically Start MySQL Server on Startup.
- You can close the System Preferences... window once you have completed your settings.

12.2.2.4 Uninstall MySQL

- Open a terminal window
- Use mysqldump to backup your databases to text files!
- Stop the database server

```

1 sudo rm /usr/local/mysql
2 sudo rm -rf /usr/local/mysql*
3 sudo rm -rf /Library/StartupItems/MySQLCOM
4 sudo rm -rf /Library/PreferencePanes/My*
5
6 edit /etc/hostconfig and remove the line MYSQLCOM==YES-
7
8 rm -rf ~/Library/PreferencePanes/My*
9 sudo rm -rf /Library/Receipts/mysql*
10 sudo rm -rf /Library/Receipts/MySQL*
11 sudo rm -rf /private/var/db/receipts/*mysql*
12 sudo rm -rf /var/db/receipts/com.mysql.mysql*
```

12.3 SQLDeveloper

12.3.1 Download SQLDeveloper

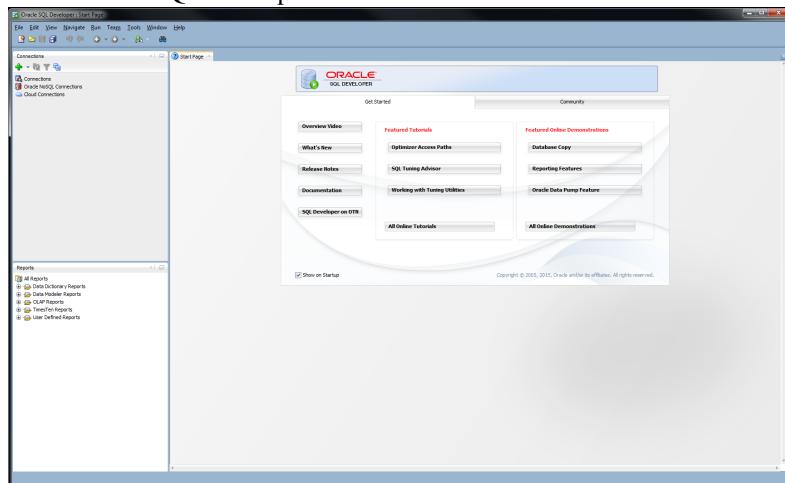
- Download SQLDeveloper from:
<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>
- Windows 64-bit with JDK 8 included
<http://download.oracle.com/otn/java/sqldeveloper/sqldeveloper-4.1.1.19.59-x64.zip>
- Windows 32-bit/64-bit
<http://download.oracle.com/otn/java/sqldeveloper/sqldeveloper-4.1.1.19.59-no-jre.zip>
- These two require JDK 8, if you prefer JDK 7, then download from:
<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/sqldev-downloads-403-2529784.html>
- Windows 64-bit - zip file includes the JDK 7
<http://download.oracle.com/otn/java/sqldeveloper/sqldeveloper-4.0.3.16.84-x64.zip>
- Windows 32/64-bit
<http://download.oracle.com/otn/java/sqldeveloper/sqldeveloper-4.0.3.16.84-no-jre.zip>
- Mac
<http://download.oracle.com/otn/java/sqldeveloper/sqldeveloper-4.1.3.20.78-macosx.app.zip>

12.3.2 Install SQLDeveloper

- Installing SQLDeveloper on Microsoft Windows Using a noinstall Zip Archive
 Extracting the Install Archive sqldeveloper-4.1.1.19.59-x64.zip to C:\opt
 It will create a directory sqldeveloper, so you will have C:\opt\sqldeveloper
- Add sqldeveloper directory to Path
 Add C:\opt\sqldeveloper to Windows Path.
`PATH=C:\opt\sqldeveloper;%PATH%`
- Installing SQLDeveloper on Mac, double the downloaded file, follow instructions, then move SQLDeveloper to Applications

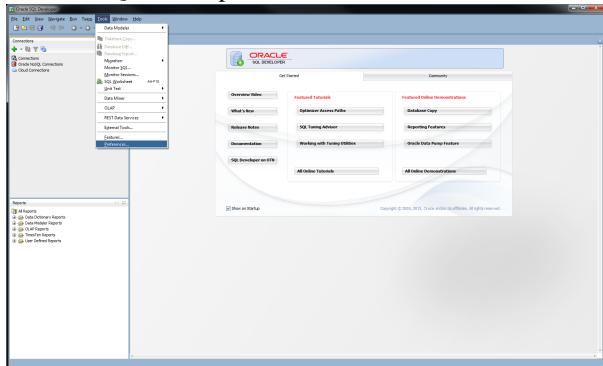
12.3.3 Start SQLDeveloper

- Double click SQLDeveloper

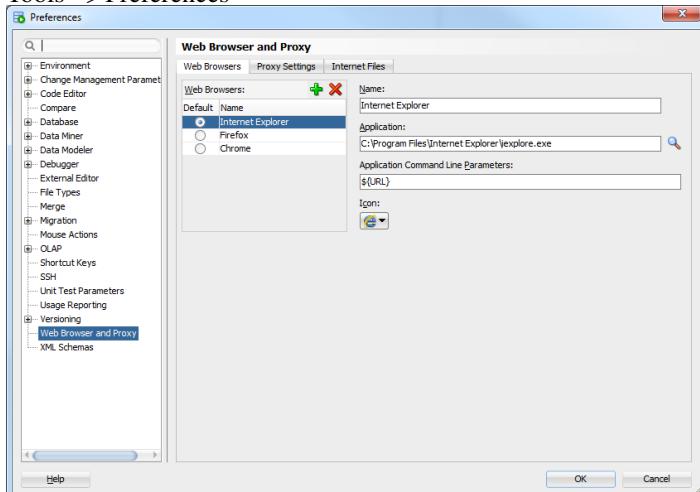


12.3.4 Configure Proxy

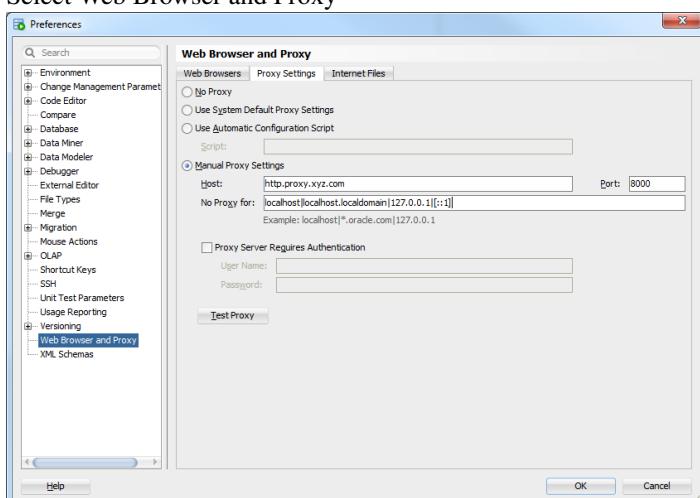
- Launch SQLDeveloper



Tools → Preferences



Select Web Browser and Proxy



Select Proxy Settings tab, select Manual Proxy Settings

Host: http.proxy.xyz.com
 Port: 8000
 Click Test Proxy, you should see Successful, then click OK.

12.3.5 Configure JDBC Driver in SQL Developer

12.3.5.1 Download JDBC Drivers

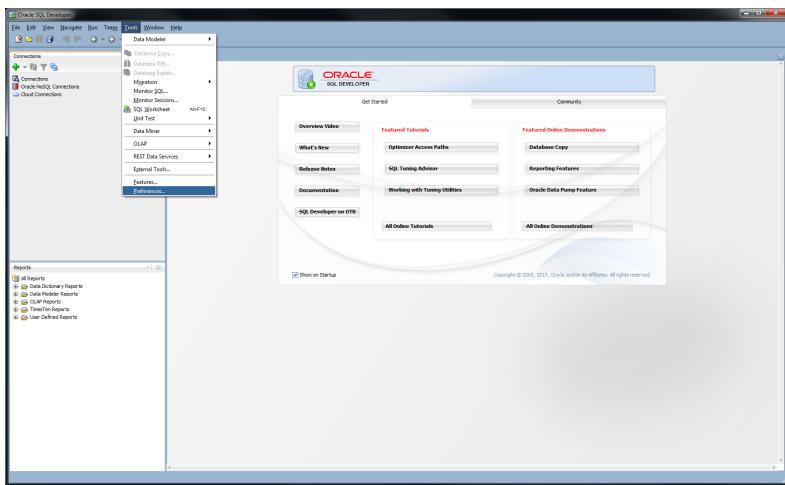
- JDBC is the Java Database Connectivity standard and it provides a mechanism for Java programs to connect to databases. To access databases using JDBC, you must use a JDBC driver. Database vendors offer JDBC drivers as free downloads.
<http://www.oracle.com/technetwork/database/migration/jdbc-migration-1923524.html>

SQL Developer supports the following JDBC drivers.

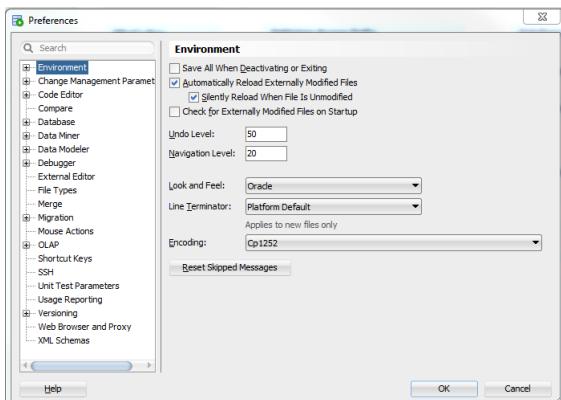
- IBM DB2: You need the binary driver jar files db2jcc.jar and db2jcc_license_cu.jar or db2jcc_license_cisuz.jar. Search for DB2 Universal JDBC Drivers or download it from:
<http://www-01.ibm.com/software/data/db2/linux-unix-windows/downloads.html>
<http://www-01.ibm.com/support/docview.wss?rs=4020&uid=swg21385217>
<http://www-01.ibm.com/support/docview.wss?uid=swg21363866>
Download: IBM Data Server Client for Windows (64bit)
ibm_data_server_client_winx64_V10.5.zip
Extracting and Install client, get db2jcc.jar and db2jcc_license_cu.jar or db2jcc_license_cisuz.jar from installation directory, copy to C:\opt\sqldeveloper\jdbc\lib
- Microsoft Access: No additional driver is required. Access uses the JDBC/ODBC bridge
- Microsoft SQL Server and Sybase: JTDS driver version 1.2. Download it from:
<http://sourceforge.net/projects/jtds/files/>
DO NOT download 1.3.1, click jtds: <http://sourceforge.net/projects/jtds/files/jtds/>
Click 1.2: <http://sourceforge.net/projects/jtds/files/jtds/1.2/>
Download jtds-1.2-dist.zip
Extracting jtds-1.2-dist.zip, get jtds-1.2.jar, copy to C:\opt\sqldeveloper\jdbc\lib
- MySQL: MySQL JDBC Driver, version 5.04. Download it from:
<http://dev.mysql.com/downloads/connector/j/5.0.html>
mysql-connector-java-5.0.8.zip
Extracting mysql-connector-java-5.0.8.zip, get mysql-connector-java-5.0.8-bin.jar, copy to C:\opt\sqldeveloper\jdbc\lib
- Teradata: Use Teradata JDBC Driver 12.0 or above. Both the Teradata JDBC Driver 12.0 and 13.0 use the jar files terajdbc4.jar and tdgssconfig.jar. Download it from:
<http://downloads.teradata.com/download>
<http://downloads.teradata.com/download/connectivity/jdbc-driver>
TeraJDBC_indep_indep.15.10.00.09.zip
Extracting TeraJDBC_indep_indep.15.10.00.09.zip, get terajdbc4.jar and tdgssconfig.jar, copy to C:\opt\sqldeveloper\jdbc\lib

12.3.5.2 Install a JDBC Driver in Oracle SQL Developer

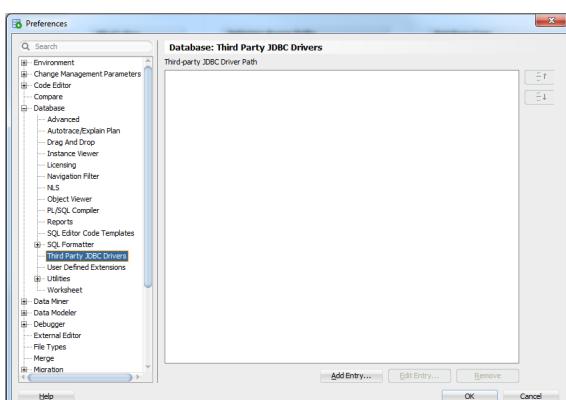
- Launch SQLDeveloper



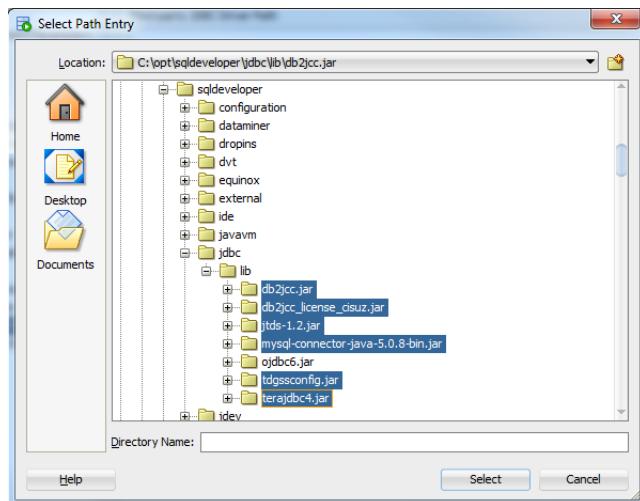
Tools → Preferences



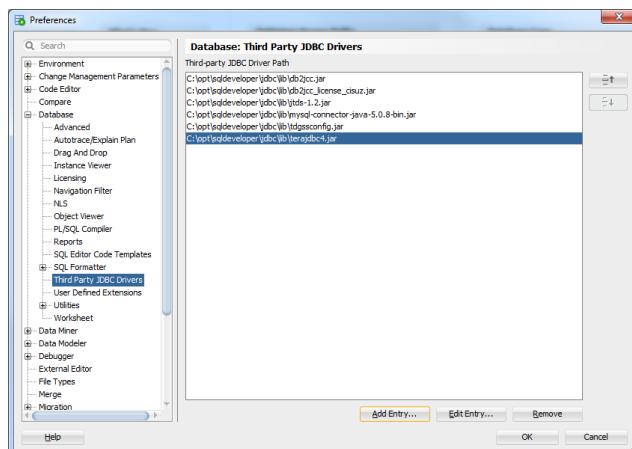
Expand Database, click Third Party JDBC Drivers



Click Add Entry...



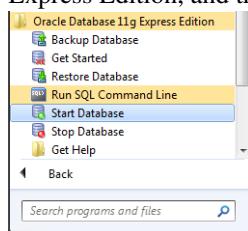
Browse to C:\opt\sqldeveloper\jdbc\lib, hold Ctrl key, select db2jcc.jar, db2jcc_license_cisuz.jar, jtds-1.2.jar, mysql-connector-java-5.0.8-bin.jar, tdgssconfig.jar and terajdbc4.jar
Click Select



Click OK

12.3.6 Connect to Oracle XE through SQL Developer

- Start OracleXE
- from the Start menu, select Programs (or All Programs), then Oracle Database 11g Express Edition, and then right click Start Database, Run as administrator.



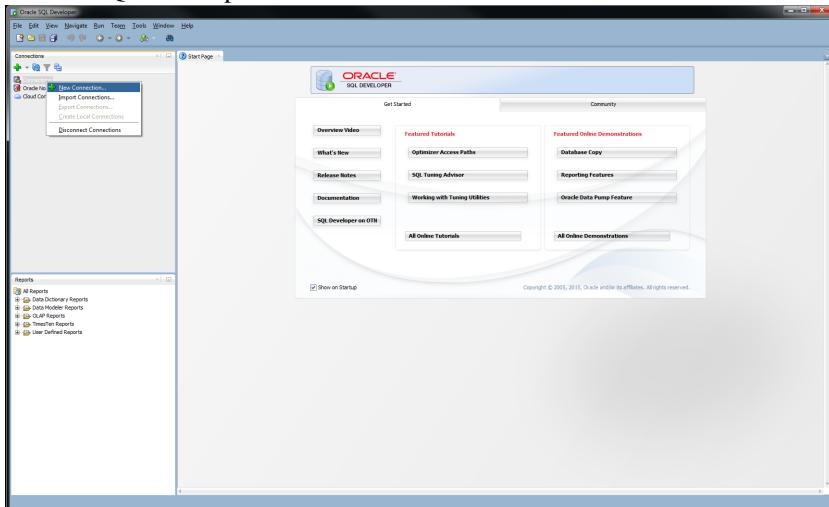
```

Administrator Start Database
The OracleServiceXE service is starting...
The OracleServiceXE service was started successfully.

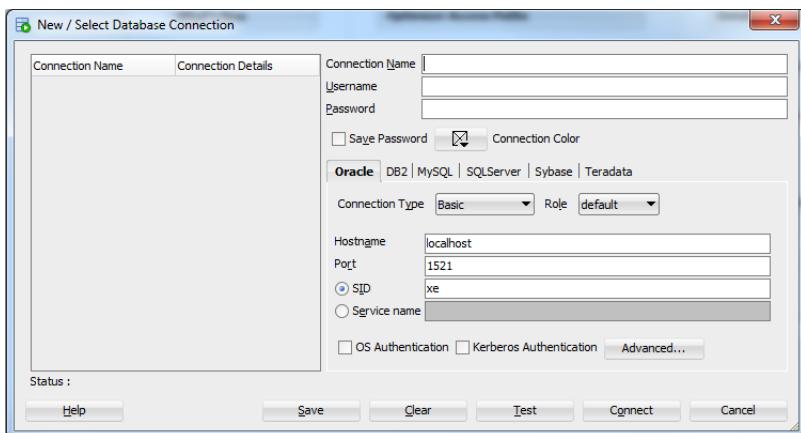
C:\opt\oraclexe\app\oracle\product\11.2.0\server\bin>

```

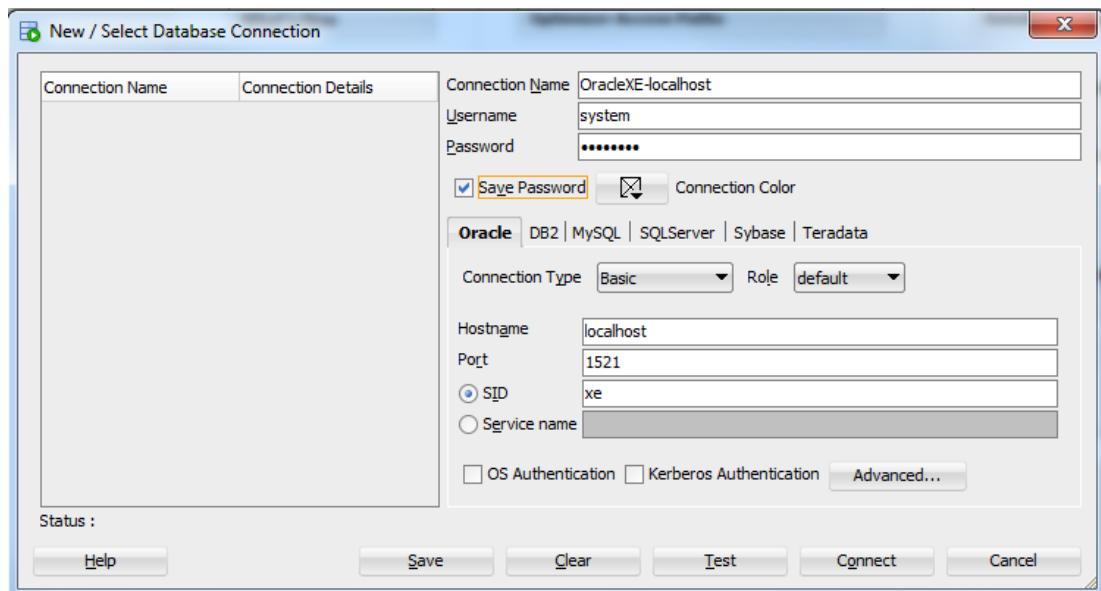
▪ Launch SQL Developer



Right click Connections ->New Connection...



You will see now we have Oracle, DB2, MySQL, Sybase, Teradata. If we do not add the JDBC drivers before, we only have Oracle option.



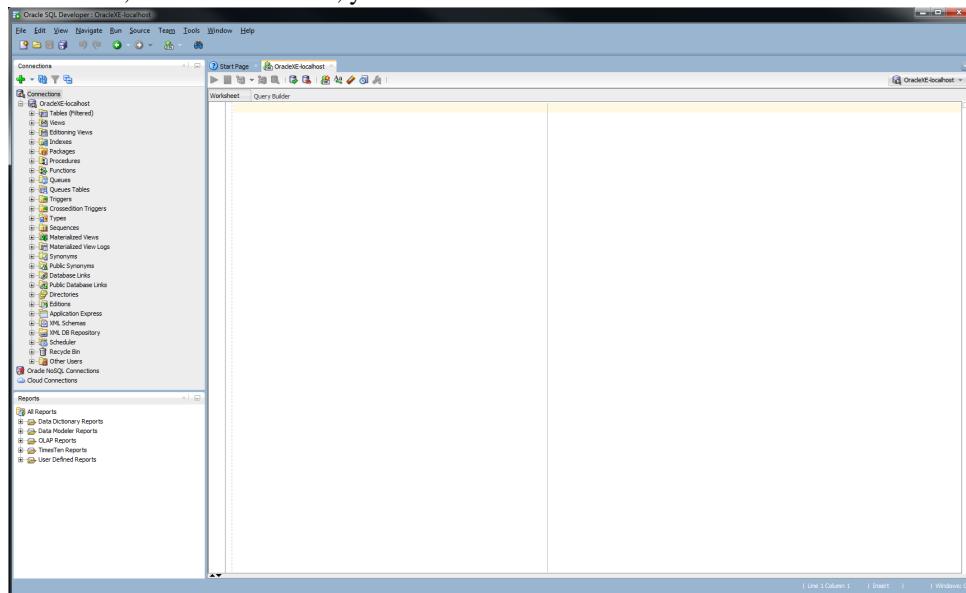
type in:

```
Connection Name: OracleXE-localhost
Username: system
Password: oraclexe
Save Password: checked
Hostname: localhost
Port: 1521
SID: xe
```

Click Test

You will see Status : Success

Click Save, then click Connect, you are now connected to OracleXE localhost!

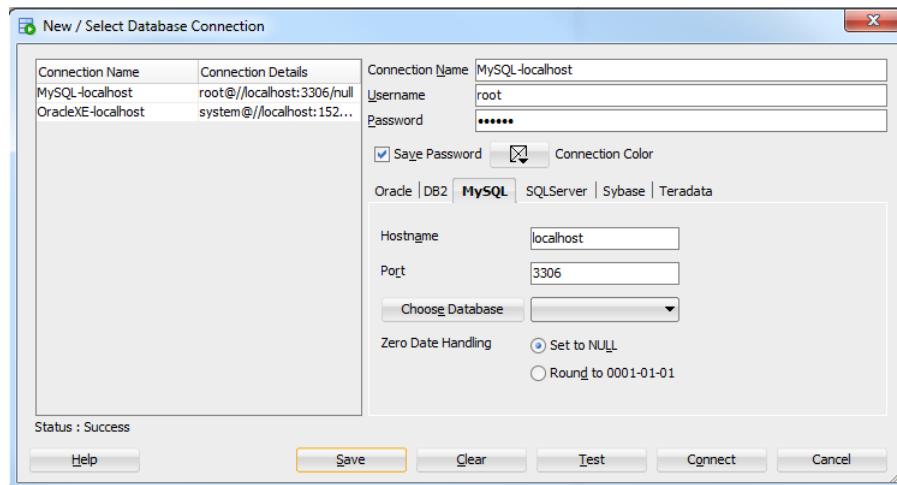


12.3.7 Connect to MySQL through SQL Developer

- Start MySQL

```
C:\> mysqld --console
```

- Launch SQL Developer, right click Connections → New Connection...



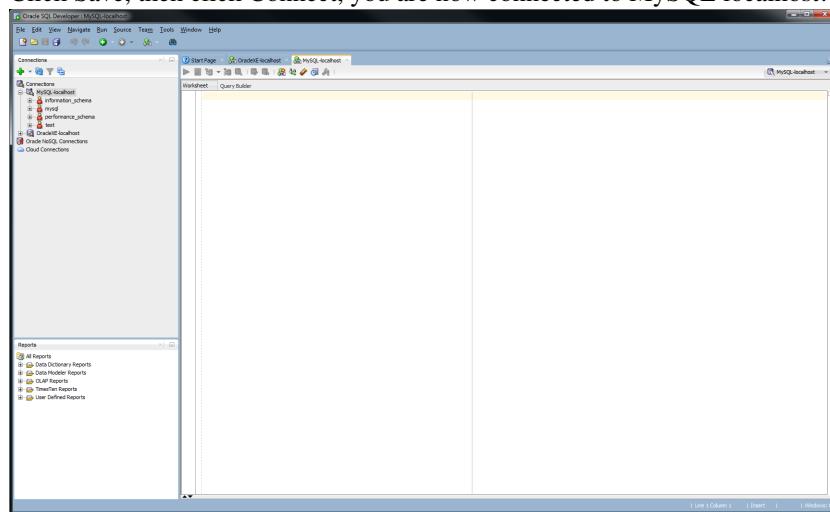
Select MySQL, type in:

```
Connection Name: MySQL-localhost
Username: root
Password: newpwd
Save Password: checked
Hostname: localhost
Port: 3306
```

Click Test

You will see Status : Success

Click Save, then click Connect, you are now connected to MySQL localhost!



12.4 HyperSQL (HSQLDB)

HSQLDB is a portable RDBMS implemented in pure java. It can be embedded with your application as well as can be used separately. It is very a small database that supports almost all features of the standard database system. It comes with small jar file that can be found in lib folder.

The HSQLDB jar package is located in the /lib directory of the ZIP package and contains several components and programs. Core components of jar file are : HyperSQL RDBMS Engine (HSQLDB), HyperSQL JDBC Driver, Database Manager, and Sql Tool.

12.4.1 Download HyperSQL

- Download HyperSQL from:
<http://hsqldb.org/>
<http://sourceforge.net/projects/hsqldb/files/>
[hsqldb-2.3.3.zip](http://sourceforge.net/projects/hsqldb/files/hsqldb-2.3.3.zip)

12.4.2 Install HyperSQL

- Installing HyperSQL on Microsoft Windows Using a noinstall Zip Archive
 Extracting the Install Archive hsqldb-2.3.3.zip to C:\opt
 It will create a directory hsqldb-2.3.3, so you will have C:\opt\hsqldb-2.3.3\hsqldb

12.4.3 Start HyperSQL

- This is the preferred way of running a database server and the fastest one. A proprietary communications protocol is used for this mode. A command similar to those used for running tools and described above is used for running the server. The following example of the command for starting the server starts the server with one (default) database with files named "mydb.*" and the public name of "testdb". The public name hides the file names from users.

```

1 c:>cd \opt\hsqldb-2.3.3\hsqldb\data
2 c:\opt\hsqldb-2.3.3\hsqldb\data>java -cp ..\lib\hsqldb.jar
   ↳ org.hsqldb.server.Server --database.0 file:mydb --dbname.0 testdb

```

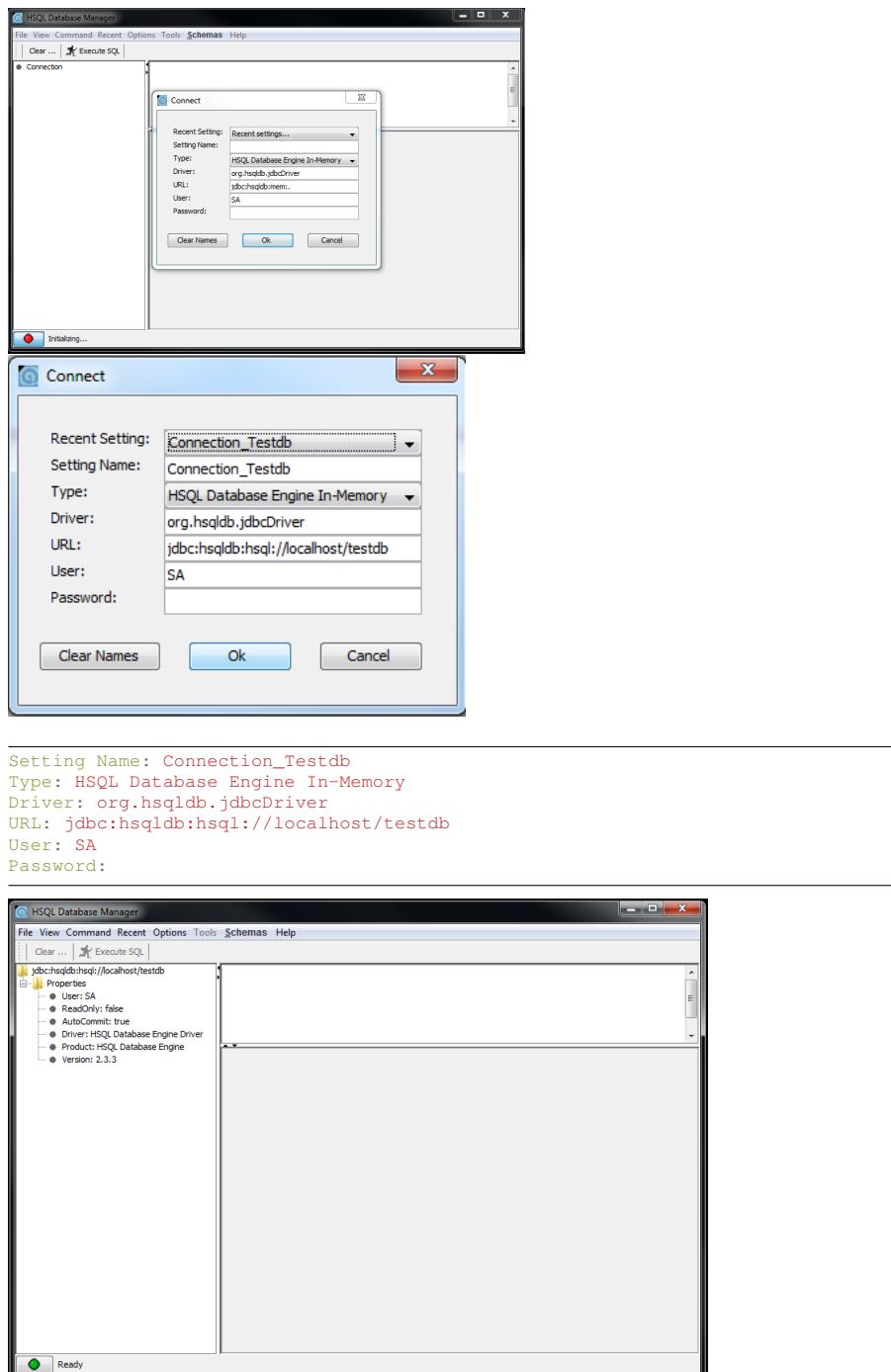
12.4.4 Connect to HyperSQL

- Running Database Access Tools

```

1 c:\opt\hsqldb-2.3.3\hsqldb\data>java -cp ..\lib\hsqldb.jar
   ↳ org.hsqldb.util.DatabaseManagerSwing

```



After connecting to your database, you can create tables using SQL statement and insert data into table. To view data from the table just execute select SQL statement for that table.

CHAPTER 13

JDBC

13.1 Quick Start

Some quick guides to show how JDBC interact with databases like MySQL, Oracle and HyperSQL.

13.1.1 Connect to MySQL

13.1.1.1 Create project using command line

```
1 c:\workspaces\eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=app-hello-jdbc-mysql
  ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

13.1.1.2 Modify pom.xml

- C:\workspaces\eclipse\my-projects\app-hello-jdbc-mysql\pom.xml

Use properties
Add mysql dependencies
Modify junit version

[†]change c:\workspaces\eclipse to /users/xyz/documents/workspace, then all the projects in this chapter will work in mac osx.

Add maven-compiler-plugin, set JDK version to 1.7

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4     http://maven.apache.org/maven-v4_0_0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <groupId>com.xyz</groupId>
7   <artifactId>app-hello-jdbc-mysql</artifactId>
8   <packaging>jar</packaging>
9   <version>1.0-SNAPSHOT</version>
10  <name>app-hello-jdbc-mysql</name>
11  <url>http://maven.apache.org</url>
12  <properties>
13    <java.version>1.7</java.version>
14    <junit.version>4.12</junit.version>
15    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
16    <mysql.version>5.1.37</mysql.version>
17  </properties>
18  <dependencies>
19    <dependency>
20      <groupId>mysql</groupId>
21      <artifactId>mysql-connector-java</artifactId>
22      <version>${mysql.version}</version>
23    </dependency>
24    <dependency>
25      <groupId>junit</groupId>
26      <artifactId>junit</artifactId>
27      <version>${junit.version}</version>
28      <scope>test</scope>
29    </dependency>
30  </dependencies>
31  <build>
32    <finalName>app-hello-jdbc-mysql</finalName>
33    <plugins>
34      <plugin>
35        <groupId>org.apache.maven.plugins</groupId>
36        <artifactId>maven-compiler-plugin</artifactId>
37        <version>${maven.compiler.plugin.version}</version>
38        <configuration>
39          <source>${java.version}</source>
40          <target>${java.version}</target>
41        </configuration>
42      </plugin>
43    </plugins>
44  </build>
45</project>
```

13.1.1.3 Make this project an eclipse project

```

1 c:\workspaces\workspace\my-projects\app-hello-jdbc-mysql>mvn eclipse:clean
  ↵ eclipse:eclipse
```

13.1.1.4 Import this project to eclipse

- Launch eclipse
File ->Import ->General ->Existing Projects into Workspace
Browse to app-hello-jdbc-mysql, and import it.

13.1.1.5 Add a java class

- src/main/java ->com.xyz ->JdbcMySqlConnect.java

```

1 package com.xyz;
2
```

```

3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 public class JdbcMySqlConnect {
8     public static void main(String[] argv) {
9         System.out.println("----- MySQL JDBC Connection -----");
10
11     try {
12         Class.forName("com.mysql.jdbc.Driver");
13     } catch (ClassNotFoundException e) {
14         System.out.println("MySQL JDBC Driver not found.");
15         e.printStackTrace();
16         return;
17     }
18
19     System.out.println("MySQL JDBC Driver Registered!");
20     Connection connection = null;
21
22     try {
23         connection =
24             DriverManager.getConnection("jdbc:mysql://localhost:3306",
25             "root", "newpwd");
26     } catch (SQLException e) {
27         System.out.println("Open Connection Failed! Check output console");
28         e.printStackTrace();
29         return;
30     }
31
32     if (connection != null) {
33         System.out.println("Open Connection Success!");
34     } else {
35         System.out.println("Open Connection Failed!");
36     }
37
38     try {
39         connection.close();
40         System.out.println("Close Connection Success!");
41     } catch (SQLException e) {
42         System.out.println("Close Connection Failed!");
43         e.printStackTrace();
44     }
45 }

```

13.1.1.6 Start MySQL

```
C:\> mysqld --console
```

13.1.1.7 Run App

Right click on project src/main/java ->com.xyz ->JdbcMySqlConnect

Run As ->Java Application

You will see in the console:

```

1 ----- MySQL JDBC Connection -----
2 MySQL JDBC Driver Registered!
3 Open Connection Success!
4 Close Connection Success!

```

13.1.2 Connect to Oracle

13.1.2.1 Download Oracle JDBC Driver

- Two ways to get the Oracle jdbc driver:
Download from Oracle.com, or Oracle database installed folder, for example, "ORACLE_HOME\jdbc\lib\ojdbc6.jar"
- Download Oracle JDBC Driver (JDBC Thin for All Platforms) from:
<http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>
Oracle Database 12c Release 1 (12.1.0.2) driver, ojdbc7.jar
Oracle Database 11g Release 2 (11.2.0.4) driver, ojdbc6.jar

13.1.2.2 Install Oracle JDBC Driver to Maven

- Install these drivers to your maven local repo

```

1 mvn install:install-file -Dfile=ojdbc7.jar -DgroupId=com.oracle
  ↵ -DartifactId=ojdbc7 -Dversion=12.1.0.2 -Dpackaging=jar
2 mvn install:install-file -Dfile=ojdbc6.jar -DgroupId=com.oracle
  ↵ -DartifactId=ojdbc6 -Dversion=11.2.0.4 -Dpackaging=jar

```

13.1.2.3 Create project using command line

```

1 c:\workspaces\eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↵ -DartifactId=app-hello-jdbc-oracle
  ↵ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

```

13.1.2.4 Modify pom.xml

- C:\workspaces\eclipse\my-projects\app-hello-jdbc-oracle\pom.xml

Use properties
Add oracle dependencies
Modify junit version
Add maven-compiler-plugin, set JDK version to 1.7

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↵ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  ↵ http://maven.apache.org/maven-v4_0_.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-jdbc-oracle</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-jdbc-oracle</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <java.version>1.7</java.version>
12    <junit.version>4.12</junit.version>
13    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
14    <oracle.version>12.1.0.2</oracle.version>
15  </properties>
16  <dependencies>
17    <dependency>
18      <groupId>com.oracle</groupId>
19      <artifactId>ojdbc7</artifactId>
20      <version>${oracle.version}</version>
21    </dependency>
22    <dependency>
23      <groupId>junit</groupId>
24      <artifactId>junit</artifactId>
25      <version>${junit.version}</version>
26      <scope>test</scope>
27    </dependency>

```

```

28   </dependencies>
29   <build>
30     <finalName>app-hello-jdbc-oracle</finalName>
31     <plugins>
32       <plugin>
33         <groupId>org.apache.maven.plugins</groupId>
34         <artifactId>maven-compiler-plugin</artifactId>
35         <version>${maven.compiler.plugin.version}</version>
36         <configuration>
37           <source>${java.version}</source>
38           <target>${java.version}</target>
39         </configuration>
40       </plugin>
41     </plugins>
42   </build>
43 </project>

```

13.1.2.5 Make this project an eclipse project

```

1 c:\workspaces\workspace\my-projects\app-hello-jdbc-oracle>mvn eclipse:clean
  → eclipse:eclipse

```

13.1.2.6 Import this project to eclipse

- Launch eclipse
File ->Import ->General ->Existing Projects into Workspace
Browse to app-hello-jdbc-oracle, and import it.

13.1.2.7 Add java classes

- src/main/java ->com.xyz ->JdbcOracleXEConnect.java

```

1 package com.xyz;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 public class JdbcOracleXEConnect {
8     public static void main(String[] args) {
9         System.out.println("----- Oracle XE JDBC Connection -----");
10
11        try {
12            Class.forName("oracle.jdbc.driver.OracleDriver");
13        } catch (ClassNotFoundException e) {
14            System.out.println("Oracle JDBC Driver not found.");
15            e.printStackTrace();
16            return;
17        }
18
19        System.out.println("Oracle JDBC Driver Registered!");
20        Connection connection = null;
21
22        try {
23            connection =
24                DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
25                "system", "oraclexe");
26        } catch (SQLException e) {
27            System.out.println("Open Connection Failed! Check output console");
28            e.printStackTrace();
29            return;
30        }
31
32        if (connection != null) {
33            System.out.println("Open Connection Success!");
34        } else {

```

```

33         System.out.println("Open Connection Failed!");
34     }
35
36     try {
37         connection.close();
38         System.out.println("Close Connection Success!");
39     } catch (SQLException e) {
40         System.out.println("Close Connection Failed!");
41         e.printStackTrace();
42         return;
43     }
44 }
45

```

src/main/java -> com.xyz -> JdbcOracleConnect.java

```

1 package com.xyz;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 public class JdbcOracleConnect {
8     public static void main(String[] argv) {
9         System.out.println("----- Oracle JDBC Connection -----");
10
11        try {
12            Class.forName("oracle.jdbc.driver.OracleDriver");
13        } catch (ClassNotFoundException e) {
14            System.out.println("Oracle JDBC Driver not found.");
15            e.printStackTrace();
16            return;
17        }
18
19        System.out.println("Oracle JDBC Driver Registered!");
20        Connection connection = null;
21
22        try {
23            connection =
24                → DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:orcl",
25                → "system", "password");
26        } catch (SQLException e) {
27            System.out.println("Open Connection Failed! Check output console");
28            e.printStackTrace();
29            return;
30        }
31
32        if (connection != null) {
33            System.out.println("Open Connection Success!");
34        } else {
35            System.out.println("Open Connection Failed!");
36        }
37
38        try {
39            connection.close();
40            System.out.println("Close Connection Success!");
41        } catch (SQLException e) {
42            System.out.println("Close Connection Failed!");
43            e.printStackTrace();
44            return;
45        }
46    }
47

```

13.1.2.8 Start Oracle

- Oracle XE

Your Oracle XE should already started automatically when reboot.

- Oracle 12c
Start Oracle 12c VM.

13.1.2.9 Run App

Right click on project src/main/java ->com.xyz ->JdbcOracleXEConnect

Run As ->Java Application

You will see in the console:

```

1 ----- Oracle XE JDBC Connection -----
2 Oracle JDBC Driver Registered!
3 Open Connection Success!
4 Close Connection Success!
```

Right click on project src/main/java ->com.xyz ->JdbcOracleConnect

Run As ->Java Application

You will see in the console:

```

1 ----- Oracle JDBC Connection -----
2 Oracle JDBC Driver Registered!
3 Open Connection Success!
4 Close Connection Success!
```

13.1.3 Connect to HyperSQL

13.1.3.1 Create project using command line

```

1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=app-hello-jdbc-hsql
  ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

13.1.3.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\app-hello-jdbc-hsql\pom.xml

Use properties

Add hsql dependencies

Modify junit version

Add maven-compiler-plugin, set JDK version to 1.7

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-jdbc-hsql</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-jdbc-hsql</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <java.version>1.7</java.version>
12    <junit.version>4.12</junit.version>
13    <hsql.version>2.3.3</hsql.version>
14    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
15  </properties>
16  <dependencies>
17    <dependency>
```

```

18      <groupId>org.hsqldb</groupId>
19      <artifactId>hsqldb</artifactId>
20      <version>${hsqldb.version}</version>
21  </dependency>
22  <dependency>
23      <groupId>junit</groupId>
24      <artifactId>junit</artifactId>
25      <version>${junit.version}</version>
26      <scope>test</scope>
27  </dependency>
28 </dependencies>
29 <build>
30     <finalName>app-hello-jdbc-hsql</finalName>
31     <plugins>
32         <plugin>
33             <groupId>org.apache.maven.plugins</groupId>
34             <artifactId>maven-compiler-plugin</artifactId>
35             <version>${maven.compiler.plugin.version}</version>
36             <configuration>
37                 <source>${java.version}</source>
38                 <target>${java.version}</target>
39             </configuration>
40         </plugin>
41     </plugins>
42 </build>
43 </project>

```

13.1.3.3 Make this project an eclipse project

```

1 c:\workspaces\workspace\my-projects\app-hello-jdbc-hsql>mvn eclipse:clean
  ↳ eclipse:eclipse

```

13.1.3.4 Import this project to eclipse

- Launch eclipse
File → Import → General → Existing Projects into Workspace
Browse to app-hello-jdbc-hsql, and import it.

13.1.3.5 Add a java class

- src/main/java → com.xyz → JdbcHsqlConnect.java
-

```

1 package com.xyz;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 public class JdbcHsqlConnect {
8     public static void main(String[] argv) {
9         System.out.println("----- HyperSQL JDBC Connection -----");
10
11         try {
12             Class.forName("org.hsqldb.jdbcDriver");
13         } catch (ClassNotFoundException e) {
14             System.out.println("HyperSQL JDBC Driver not found.");
15             e.printStackTrace();
16             return;
17         }
18
19         System.out.println("HyperSQL JDBC Driver Registered!");
20         Connection connection = null;
21
22         try {
23             connection =
24                 DriverManager.getConnection("jdbc:hsqldb:hsqldb://localhost/testdb",
25                     "SA", "");

```

```

24     } catch (SQLException e) {
25         System.out.println("Open Connection Failed! Check output console");
26         e.printStackTrace();
27         return;
28     }
29
30     if (connection != null) {
31         System.out.println("Open Connection Success!");
32     } else {
33         System.out.println("Open Connection Failed!");
34     }
35
36     try {
37         connection.close();
38         System.out.println("Close Connection Success!");
39     } catch (SQLException e) {
40         System.out.println("Close Connection Failed!");
41         e.printStackTrace();
42         return;
43     }
44 }
45 }
```

13.1.3.6 Start HyperSQL

```
c:>cd \opt\hsqldb-2.3.3\hsqldb\data
c:\opt\hsqldb-2.3.3\hsqldb\data>java -cp ..\lib\hsqldb.jar
→ org.hsqldb.server.Server --database.0 file:mydb --dbname.0 testdb
```

13.1.3.7 Run App

Right click on project src/main/java ->com.xyz ->JdbcHsqlConnect

Run As ->Java Application

You will see in the console:

```
1 ----- HyperSQL JDBC Connection -----
2 HyperSQL JDBC Driver Registered!
3 Open Connection Success!
4 Close Connection Success!
```

13.1.3.8 In memory database

- Let's create a in memory HSQL database.
- We do not need the HyperSQL database we started before, we can stop it.
- Add a java class src/main/java ->com.xyz ->JdbcHsqlMemConnect.java

```

1 package com.xyz;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 import org.hsqldb.server.Server;
8
9 public class JdbcHsqlMemConnect {
10     public static void main(String[] argv) {
11         System.out.println("----- HyperSQL Memory JDBC Connection
12             -----");
13
14         // stub to get in/out of embedded db
15         Server hsqlServer = null;
16
17         hsqlServer = new Server();
18         hsqlServer.setLogWriter(null);
```

```

18     hsqlServer.setSilent(true);
19     hsqlServer.setDatabaseName(0, "testmemdb");
20     hsqlServer.setDatabasePath(0, "mem:testmemdb");
21
22     hsqlServer.start();
23
24     // making a connection
25     try {
26         Class.forName("org.hsqldb.jdbcDriver");
27     } catch (ClassNotFoundException e) {
28         System.out.println("HyperSQL JDBC Driver not found.");
29         e.printStackTrace();
30         return;
31     }
32
33     System.out.println("HyperSQL JDBC Driver Registered!");
34     Connection connection = null;
35
36     try {
37         connection =
38             DriverManager.getConnection("jdbc:hsqldb:hsq://localhost/testmemdb",
39             "SA", "");
40     } catch (SQLException e) {
41         System.out.println("Open Connection Failed! Check output console");
42         e.printStackTrace();
43         return;
44     }
45
46     if (connection != null) {
47         System.out.println("Open Connection Success!");
48     } else {
49         System.out.println("Open Connection Failed!");
50     }
51
52     try {
53         connection.close();
54         System.out.println("Close Connection Success!");
55     } catch (SQLException e) {
56         System.out.println("Close Connection Failed!");
57         e.printStackTrace();
58     }
59 }
```

- Run App Right click on project src/main/java ->com.xyz ->JdbcHsqlMemConnect
Run As ->Java Application
You will see in the console:

```

1 ----- HyperSQL Memory JDBC Connection -----
2 HyperSQL JDBC Driver Registered!
3 Open Connection Success!
4 Close Connection Success!
```

13.2 Statement

- The "Statement" interface is used to execute a simple SQL statement with no parameters. For create, insert, update or delete statement, use "Statement.executeUpdate(sql)"; select query, use "Statement.executeQuery(sql)", general, use "Statement.execute(sql)".
- Let's use app-hello-jdbc-oracle project, drop a table, create a table, insert record(s), update a record, delete a record and read record(s).
- src/main/java ->com.xyz ->JdbcOracleStatement.java

```

1 package com.xyz;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.text.DateFormat;
9 import java.text.SimpleDateFormat;
10
11 public class JdbcOracleStatement {
12     private final String DB_DRIVER = "oracle.jdbc.driver.OracleDriver";
13     private final String DB_CONNECTION =
14         "jdbc:oracle:thin:@localhost:1522:orcl";
15     private final String DB_USER = "system";
16     private final String DB_PASSWORD = "password";
17     private final DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd
18         HH:mm:ss");
18     private Connection connection = null;
19     private Statement statement = null;
19
20     private void dropTable() throws SQLException {
21         String sql = "BEGIN EXECUTE IMMEDIATE 'DROP TABLE PERSON'; EXCEPTION
22             WHEN OTHERS THEN NULL; END;";
22         System.out.println(sql);
23         statement.execute(sql);
24         System.out.println("Table is dropped!\n");
25     }
26
27     private void createTable() throws SQLException {
28         String sql = "CREATE TABLE PERSON(PID NUMBER(10) NOT NULL, LASTNAME
29             VARCHAR(20) NOT NULL, FIRSTNAME VARCHAR(20) NOT NULL, "
30             + "CREATED_BY VARCHAR(20) NOT NULL, CREATED_DATE DATE NOT NULL,
31             PRIMARY KEY (PID) " + ")";
32         System.out.println(sql);
33         statement.execute(sql);
34         System.out.println("Table is created!\n");
35     }
34
36     private void insertRecord() throws SQLException {
37         String sql = "INSERT INTO PERSON (PID, LASTNAME, FIRSTNAME, CREATED_BY,
38             CREATED_DATE) "
39             + "VALUES (1, 'Smith', 'John', 'system', to_date('"
40             + getCurrentTimeStamp()
41             + "', 'yyyy/mm/dd hh24:mi:ss'))";
42         System.out.println(sql);
43         statement.executeUpdate(sql);
44         System.out.println("Record is inserted!\n");
45     }
45
46     private void updateRecord() throws SQLException {
47         String sql = "UPDATE PERSON SET LASTNAME = 'Doe' WHERE PID = 1";
48         System.out.println(sql);
49         statement.executeUpdate(sql);
50         System.out.println("Record is updated!\n");
51     }
51
52     private void deleteRecord() throws SQLException {
53         String sql = "DELETE PERSON WHERE PID = 1";
54         System.out.println(sql);
55         statement.executeUpdate(sql);
56         System.out.println("Record is deleted!\n");
57     }
57
58     private void listRecord() throws SQLException {
59         String sql = "SELECT PID, LASTNAME, FIRSTNAME FROM PERSON";
60         System.out.println(sql);
61         ResultSet rs = statement.executeQuery(sql);
62         while (rs.next()) {
63             String pid = rs.getString("PID");
64             String lastName = rs.getString("LASTNAME");
65             String firstName = rs.getString("FIRSTNAME");

```

```

66         System.out.println("PID: " + pid);
67         System.out.println("LASTNAME: " + lastName);
68         System.out.println("FIRSTNAME: " + firstName + "\n");
69     }
70 }
71
72 private void batchInsertRecord() throws SQLException {
73     String sql1 = "INSERT INTO PERSON(PID, LASTNAME, FIRSTNAME, CREATED_BY,
74         ↪ CREATED_DATE) "
75         + "VALUES(1, 'Smith', 'John', 'system', to_date('"
76         ↪ getCurrentTimeStamp()
77         + "', 'yyyy/mm/dd hh24:mi:ss'))";
78     String sql2 = "INSERT INTO PERSON(PID, LASTNAME, FIRSTNAME, CREATED_BY,
79         ↪ CREATED_DATE) "
80         + "VALUES(2, 'Doe', 'Jane', 'system', to_date('"
81         ↪ getCurrentTimeStamp()
82         + "', 'yyyy/mm/dd hh24:mi:ss'))";
83     String sql3 = "INSERT INTO PERSON(PID, LASTNAME, FIRSTNAME, CREATED_BY,
84         ↪ CREATED_DATE) "
85         + "VALUES(3, 'Brown', 'Charlie', 'system', to_date('"
86         ↪ getCurrentTimeStamp()
87         + "', 'yyyy/mm/dd hh24:mi:ss'))";
88
89     System.out.println(sql1);
90     System.out.println(sql2);
91     System.out.println(sql3);
92     connection.setAutoCommit(false);
93     statement.addBatch(sql1);
94     statement.addBatch(sql2);
95     statement.addBatch(sql3);
96     statement.executeBatch();
97     connection.commit();
98     System.out.println("Records are inserted!\n");
99 }
100
101 private Connection getDBConnection() throws ClassNotFoundException,
102     ↪ SQLException {
103     Class.forName(DB_DRIVER);
104     connection = DriverManager.getConnection(DB_CONNECTION, DB_USER,
105         ↪ DB_PASSWORD);
106     return connection;
107 }
108
109 private String getCurrentTimeStamp() {
110     java.util.Date today = new java.util.Date();
111     return dateFormat.format(today.getTime());
112 }
113
114 private void run() throws ClassNotFoundException, SQLException {
115     connection = getDBConnection();
116     statement = connection.createStatement();
117     dropTable();
118     createTable();
119     insertRecord();
120     listRecord();
121     updateRecord();
122     listRecord();
123     deleteRecord();
124     listRecord();
125     batchInsertRecord();
126     listRecord();
127     statement.close();
128     connection.close();
129 }
130
131 public static void main(String[] argv) {
132     JdbcOracleStatement jdbcOracleStatement = new JdbcOracleStatement();
133     try {
134         jdbcOracleStatement.run();
135     } catch (ClassNotFoundException e) {
136         e.printStackTrace();
137     } catch (SQLException e) {
138         e.printStackTrace();
139     }
140 }
```

```
131      }
132 }
```

Result

```

1 BEGIN EXECUTE IMMEDIATE 'DROP TABLE PERSON'; EXCEPTION WHEN OTHERS THEN NULL;
   → END;
2 Table is dropped!
3
4 CREATE TABLE PERSON(PID NUMBER(10) NOT NULL, LASTNAME VARCHAR(20) NOT NULL,
   → FIRSTNAME VARCHAR(20) NOT NULL, CREATED_BY VARCHAR(20) NOT NULL,
   → CREATED_DATE DATE NOT NULL, PRIMARY KEY (PID))
5 Table is created!
6
7 INSERT INTO PERSON (PID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE) VALUES
   → (1, 'Smith', 'John', 'system', to_date('2015/12/04 14:10:50', 'yyyy/mm/dd
   → hh24:mi:ss'))
8 Record is inserted!
9
10 SELECT PID, LASTNAME, FIRSTNAME FROM PERSON
11 PID: 1
12 LASTNAME: Smith
13 FIRSTNAME: John
14
15 UPDATE PERSON SET LASTNAME = 'Doe' WHERE PID = 1
16 Record is updated!
17
18 SELECT PID, LASTNAME, FIRSTNAME FROM PERSON
19 PID: 1
20 LASTNAME: Doe
21 FIRSTNAME: John
22
23 DELETE PERSON WHERE PID = 1
24 Record is deleted!
25
26 SELECT PID, LASTNAME, FIRSTNAME FROM PERSON
27 INSERT INTO PERSON(PID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE) VALUES(1,
   → 'Smith', 'John', 'system', to_date('2015/12/04 14:10:50', 'yyyy/mm/dd
   → hh24:mi:ss'))
28 INSERT INTO PERSON(PID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE) VALUES(2,
   → 'Doe', 'Jane', 'system', to_date('2015/12/04 14:10:50', 'yyyy/mm/dd
   → hh24:mi:ss'))
29 INSERT INTO PERSON(PID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE) VALUES(3,
   → 'Brown', 'Charlie', 'system', to_date('2015/12/04 14:10:50', 'yyyy/mm/dd
   → hh24:mi:ss'))
30 Records are inserted!
31
32 SELECT PID, LASTNAME, FIRSTNAME FROM PERSON
33 PID: 1
34 LASTNAME: Smith
35 FIRSTNAME: John
36
37 PID: 2
38 LASTNAME: Doe
39 FIRSTNAME: Jane
40
41 PID: 3
42 LASTNAME: Brown
43 FIRSTNAME: Charlie

```

13.3 PreparedStatement

- The "PreparedStatement" interface is extended "Statement", with extra feature to send a pre-compiled SQL statement with parameters. For create, insert, update or delete statement, uses "PreparedStatement.executeUpdate(sql)"; select query, uses

”PreparedStatement.executeQuery(sql)”。

- src/main/java ->com.xyz ->JdbcOraclePreparedStatement.java

```

1 package com.xyz;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.sql.Timestamp;
9
10 public class JdbcOraclePreparedStatement {
11     private final String DB_DRIVER = "oracle.jdbc.driver.OracleDriver";
12     private final String DB_CONNECTION =
13         "jdbc:oracle:thin:@localhost:1522:orcl";
14     private final String DB_USER = "system";
15     private final String DB_PASSWORD = "password";
16     private Connection connection = null;
17     private PreparedStatement statement = null;
18
19     private void dropTable() throws SQLException {
20         String sql = "BEGIN EXECUTE IMMEDIATE 'DROP TABLE PERSON'; EXCEPTION
21             WHEN OTHERS THEN NULL; END;";
22         System.out.println(sql);
23         statement = connection.prepareStatement(sql);
24         statement.execute();
25         System.out.println("Table is dropped!\n");
26     }
27
28     private void createTable() throws SQLException {
29         String sql = "CREATE TABLE PERSON(PID NUMBER(10) NOT NULL, LASTNAME
30             ← VARCHAR(20) NOT NULL, FIRSTNAME VARCHAR(20) NOT NULL,
31             + "CREATED_BY VARCHAR(20) NOT NULL, CREATED_DATE DATE NOT NULL,
32             ← PRIMARY KEY (PID) " + ")";
33         System.out.println(sql);
34         statement = connection.prepareStatement(sql);
35         statement.execute();
36         System.out.println("Table is created!\n");
37     }
38
39     private void insertRecord() throws SQLException {
40         String sql = "INSERT INTO PERSON (PID, LASTNAME, FIRSTNAME, CREATED_BY,
41             ← CREATED_DATE) VALUES (?, ?, ?, ?, ?)";
42         System.out.println(sql);
43         statement = connection.prepareStatement(sql);
44         statement.setInt(1, 1);
45         statement.setString(2, "Smith");
46         statement.setString(3, "John");
47         statement.setString(4, "system");
48         statement.setTimestamp(5, getCurrentTimeStamp());
49         statement.executeUpdate();
50         System.out.println("Record is inserted!\n");
51     }
52
53     private void updateRecord() throws SQLException {
54         String sql = "UPDATE PERSON SET LASTNAME = ? WHERE PID = ?";
55         System.out.println(sql);
56         statement = connection.prepareStatement(sql);
57         statement.setString(1, "Doe");
58         statement.setInt(2, 1);
59         statement.executeUpdate();
60         System.out.println("Record is updated!\n");
61     }
62
63     private void deleteRecord() throws SQLException {
64         String sql = "DELETE PERSON WHERE PID = ?";
65         System.out.println(sql);
66         statement = connection.prepareStatement(sql);
67         statement.setInt(1, 1);
68         statement.executeUpdate();
69     }
70 }
```

```

64     System.out.println("Record is deleted!\n");
65 }
66
67 private void listRecord() throws SQLException {
68     //     String sql = "SELECT PID, LASTNAME, FIRSTNAME FROM PERSON WHERE PID =
69     ↪    ?";
70     String sql = "SELECT PID, LASTNAME, FIRSTNAME FROM PERSON WHERE
71     ↪    FIRSTNAME LIKE ?";
72     System.out.println(sql);
73     statement = connection.prepareStatement(sql);
74     ↪    statement.setInt(1, 1);
75     statement.setString(1, "J%");
76     ResultSet rs = statement.executeQuery();
77     while (rs.next()) {
78         String pid = rs.getString("PID");
79         String lastName = rs.getString("LASTNAME");
80         String firstName = rs.getString("FIRSTNAME");
81         System.out.println("PID: " + pid);
82         System.out.println("LASTNAME: " + lastName);
83         System.out.println("FIRSTNAME: " + firstName + "\n");
84     }
85 }
86
87 private void batchInsertRecord() throws SQLException {
88     String sql = "INSERT INTO PERSON (PID, LASTNAME, FIRSTNAME, CREATED_BY,
89     ↪    CREATED_DATE) VALUES (?, ?, ?, ?, ?)";
90     System.out.println(sql);
91     statement = connection.prepareStatement(sql);
92     connection.setAutoCommit(false);
93
94     statement.setInt(1, 1);
95     statement.setString(2, "Smith");
96     statement.setString(3, "John");
97     statement.setString(4, "system");
98     statement.setTimestamp(5, getCurrentTimeStamp());
99     statement.addBatch();
100
101     statement.setInt(1, 2);
102     statement.setString(2, "Doe");
103     statement.setString(3, "Jane");
104     statement.setString(4, "system");
105     statement.setTimestamp(5, getCurrentTimeStamp());
106     statement.addBatch();
107
108     statement.setInt(1, 3);
109     statement.setString(2, "Brown");
110     statement.setString(3, "Charlie");
111     statement.setString(4, "system");
112     statement.setTimestamp(5, getCurrentTimeStamp());
113     statement.addBatch();
114
115     statement.executeBatch();
116     connection.commit();
117     System.out.println("Records are inserted!\n");
118 }
119
120 private Connection getDBConnection() throws ClassNotFoundException,
121     ↪    SQLException {
122     Class.forName(DB_DRIVER);
123     connection = DriverManager.getConnection(DB_CONNECTION, DB_USER,
124     ↪    DB_PASSWORD);
125     return connection;
126 }
127
128 private Timestamp getCurrentTimeStamp() {
129     java.util.Date today = new java.util.Date();
130     return new Timestamp(today.getTime());
131 }
```

```

132     insertRecord();
133     listRecord();
134     updateRecord();
135     listRecord();
136     deleteRecord();
137     listRecord();
138     batchInsertRecord();
139     listRecord();
140     statement.close();
141     connection.close();
142 }
143
144 public static void main(String[] argv) {
145     JdbcOraclePreparedStatement jdbcOraclePreparedStatement = new
146         ↪ JdbcOraclePreparedStatement();
147     try {
148         jdbcOraclePreparedStatement.run();
149     } catch (ClassNotFoundException e) {
150         e.printStackTrace();
151     } catch (SQLException e) {
152         e.printStackTrace();
153     }
154 }
```

Result

```

1 BEGIN EXECUTE IMMEDIATE 'DROP TABLE PERSON'; EXCEPTION WHEN OTHERS THEN NULL;
2   ↪ END;
3 Table is dropped!
4 CREATE TABLE PERSON(PID NUMBER(10) NOT NULL, LASTNAME VARCHAR(20) NOT NULL,
5   ↪ FIRSTNAME VARCHAR(20) NOT NULL, CREATED_BY VARCHAR(20) NOT NULL,
6   ↪ CREATED_DATE DATE NOT NULL, PRIMARY KEY (PID))
5 Table is created!
6
7 INSERT INTO PERSON (PID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE) VALUES
8   ↪ (?, ?, ?, ?, ?)
8 Record is inserted!
9
10 SELECT PID, LASTNAME, FIRSTNAME FROM PERSON WHERE FIRSTNAME LIKE ?
11 PID: 1
12 LASTNAME: Smith
13 FIRSTNAME: John
14
15 UPDATE PERSON SET LASTNAME = ? WHERE PID = ?
16 Record is updated!
17
18 SELECT PID, LASTNAME, FIRSTNAME FROM PERSON WHERE FIRSTNAME LIKE ?
19 PID: 1
20 LASTNAME: Doe
21 FIRSTNAME: John
22
23 DELETE PERSON WHERE PID = ?
24 Record is deleted!
25
26 SELECT PID, LASTNAME, FIRSTNAME FROM PERSON WHERE FIRSTNAME LIKE ?
27 INSERT INTO PERSON (PID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE) VALUES
28   ↪ (?, ?, ?, ?, ?)
28 Records are inserted!
29
30 SELECT PID, LASTNAME, FIRSTNAME FROM PERSON WHERE FIRSTNAME LIKE ?
31 PID: 1
32 LASTNAME: Smith
33 FIRSTNAME: John
34
35 PID: 2
36 LASTNAME: Doe
37 FIRSTNAME: Jane
```

13.4 Stored Procedure

- JDBC CallableStatement and Stored Procedure, IN, OUT, CURSOR examples.

- Stored Procedure IN parameter

Create Stored Procedure through SQL*Plus, or SQL Developer

```

1 CREATE OR REPLACE PROCEDURE insertPERSON(
2   p_pid IN PERSON.PID%TYPE,
3   p_lastname IN PERSON.LASTNAME%TYPE,
4   p_firstname IN PERSON.FIRSTNAME%TYPE,
5   p_createdby IN PERSON.CREATED_BY%TYPE,
6   p_date IN PERSON.CREATED_DATE%TYPE)
7 IS
8 BEGIN
9   INSERT INTO PERSON ("PID", "LASTNAME", "FIRSTNAME", "CREATED_BY",
10   ↪ "CREATED_DATE")
11   VALUES (p_pid, p_lastname, p_firstname, p_createdby, p_date);
12   COMMIT;
13 END;
14 /

```

- Stored Procedure OUT parameter

```

1 CREATE OR REPLACE PROCEDURE getPERSONPid(
2   p_pid IN PERSON.PID%TYPE,
3   o_lastname OUT PERSON.LASTNAME%TYPE,
4   o_firstname OUT PERSON.FIRSTNAME%TYPE,
5   o_createdby OUT PERSON.CREATED_BY%TYPE,
6   o_date OUT PERSON.CREATED_DATE%TYPE)
7 IS
8 BEGIN
9   SELECT LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE
10   INTO o_lastname, o_firstname, o_createdby, o_date
11   FROM PERSON WHERE PID = p_pid;
12 END;
13 /

```

- Stored Procedure CURSOR parameter

```

1 CREATE OR REPLACE PROCEDURE getPERSONCursor(
2   p_lastname IN PERSON.LASTNAME%TYPE,
3   c_person OUT SYS_REFCURSOR)
4 IS
5 BEGIN
6   OPEN c_person FOR
7     SELECT * FROM PERSON WHERE LASTNAME LIKE p_lastname || '%';
8 END;
9 /

```

- src/main/java -> com.xyz -> JdbcOracleCallableStatement.java

```

1 package com.xyz;
2
3 import java.sql.CallableStatement;
4 import java.sql.Connection;
5 import java.sql.Date;
6 import java.sql.DriverManager;
7 import java.sql.ResultSet;
8 import java.sql.SQLException;
9
10 import oracle.jdbc.OracleTypes;
11
12 public class JdbcOracleCallableStatement {
13   private final String DB_DRIVER = "oracle.jdbc.driver.OracleDriver";
14   private final String DB_CONNECTION =
15     "jdbc:oracle:thin:@localhost:1522:orcl";

```

```

15     private final String DB_USER = "system";
16     private final String DB_PASSWORD = "password";
17     private Connection connection = null;
18     private CallableStatement statement = null;
19
20     private void callStoredProcINParameter() throws SQLException {
21         String sql = "{call insertPERSON(?, ?, ?, ?, ?, ?)}";
22         System.out.println(sql);
23         statement = connection.prepareCall(sql);
24         statement.setInt(1, 101);
25         statement.setString(2, "Smith");
26         statement.setString(3, "Johnny");
27         statement.setString(4, "system");
28         statement.setDate(5, getCurrentDate());
29         statement.executeUpdate();
30         System.out.println("Record is inserted!\n");
31     }
32
33     private void callStoredProcOUTParameter() throws SQLException {
34         String sql = "{call getPERSONPid(?, ?, ?, ?, ?, ?)}";
35         System.out.println(sql);
36         statement = connection.prepareCall(sql);
37         statement.setInt(1, 101);
38         statement.registerOutParameter(2, java.sql.Types.VARCHAR);
39         statement.registerOutParameter(3, java.sql.Types.VARCHAR);
40         statement.registerOutParameter(4, java.sql.Types.VARCHAR);
41         statement.registerOutParameter(5, java.sql.Types.DATE);
42         statement.executeUpdate();
43
44         String lastName = statement.getString(2);
45         String firstName = statement.getString(3);
46         String createdBy = statement.getString(4);
47         Date createdDate = statement.getDate(5);
48         System.out.println("LastName: " + lastName);
49         System.out.println("FirstName: " + firstName);
50         System.out.println("CreatedBy: " + createdBy);
51         System.out.println("CreatedDate: " + createdDate + "\n");
52     }
53
54     private void callStoredProcCURSOR() throws SQLException {
55         String sql = "{call getPERSONCursor(?, ?)}";
56         System.out.println(sql);
57         statement = connection.prepareCall(sql);
58         statement.setString(1, "Smith");
59         statement.registerOutParameter(2, OracleTypes.CURSOR);
60         statement.execute();
61
62         ResultSet rs = (ResultSet) statement.getObject(2);
63         while (rs.next()) {
64             String pid = rs.getString("PID");
65             String lastName = rs.getString("LASTNAME");
66             String firstName = rs.getString("FIRSTNAME");
67             String createdBy = rs.getString("CREATED_BY");
68             String createdDate = rs.getString("CREATED_DATE");
69             System.out.println("PID: " + pid);
70             System.out.println("LastName: " + lastName);
71             System.out.println("FirstName: " + firstName);
72             System.out.println("CreatedBy: " + createdBy);
73             System.out.println("CreatedDate: " + createdDate + "\n");
74         }
75     }
76
77     private Connection getDBConnection() throws ClassNotFoundException,
78         SQLException {
79         Class.forName(DB_DRIVER);
80         connection = DriverManager.getConnection(DB_CONNECTION, DB_USER,
81             DB_PASSWORD);
82         return connection;
83     }
84
85     private java.sql.Date getCurrentDate() {
86         java.util.Date today = new java.util.Date();
87         return new java.sql.Date(today.getTime());
88     }

```

```

86      }
87
88  private void run() throws ClassNotFoundException, SQLException {
89      connection = getDBConnection();
90      callStoredProcINParameter();
91      callStoredProcOUTParameter();
92      callStoredProcCURSOR();
93      statement.close();
94      connection.close();
95  }
96
97  public static void main(String[] argv) {
98      JdbcOracleCallableStatement jdbcOracleCallableStatement = new
99          ↪ JdbcOracleCallableStatement();
100     try {
101         jdbcOracleCallableStatement.run();
102     } catch (ClassNotFoundException e) {
103         e.printStackTrace();
104     } catch (SQLException e) {
105         e.printStackTrace();
106     }
107 }

```

Result

```

1 {call insertPERSON(?, ?, ?, ?, ?, ?)}
2 Record is inserted!
3
4 {call getPERSONPid(?, ?, ?, ?, ?, ?)}
5 LastName: Smith
6 FirstName: Johnny
7 CreatedBy: system
8 CreatedDate: 2015-12-07
9
10 {call getPERSONCursor(?, ?)}
11 PID: 1
12 LastName: Smith
13 FirstName: John
14 CreatedBy: system
15 CreatedDate: 2015-12-04 16:44:16.0
16
17 PID: 101
18 LastName: Smith
19 FirstName: Johnny
20 CreatedBy: system
21 CreatedDate: 2015-12-07 11:07:03.0

```

13.5 Transaction

- JDBC transaction make sure SQL statements within a transaction block are all executed successful, if either one of the SQL statement within transaction block is failed, abort and rollback everything within the transaction block.

- src/main/java ->com.xyz ->JdbcOracleTransaction.java

```

1 package com.xyz;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.sql.Timestamp;
9

```

```

10  public class JdbcOracleTransaction {
11      private final String DB_DRIVER = "oracle.jdbc.driver.OracleDriver";
12      private final String DB_CONNECTION =
13          "jdbc:oracle:thin:@localhost:1522:orcl";
14      private final String DB_USER = "system";
15      private final String DB_PASSWORD = "password";
16      private Connection connection = null;
17      private PreparedStatement statement = null;
18
19      private void insertRecord() throws SQLException {
20          String sql = "INSERT INTO PERSON (PID, LASTNAME, FIRSTNAME, CREATED_BY,
21              CREATED_DATE) VALUES (?, ?, ?, ?, ?)";
22          System.out.println(sql);
23          statement = connection.prepareStatement(sql);
24          statement.setInt(1, 201);
25          statement.setString(2, "Smith");
26          statement.setString(3, "Johnathan");
27          statement.setString(4, "system");
28          statement.setTimestamp(5, getCurrentTimeStamp());
29          statement.executeUpdate();
30          System.out.println("Record is inserted!\n");
31      }
32
33      private void updateRecord() throws SQLException {
34          String sql = "UPDATE PERSON SET LASTNAME = ? WHERE PID = ?";
35          System.out.println(sql);
36          statement = connection.prepareStatement(sql);
37          statement.setString(1, "Doe");
38          statement.setInt(2, 201);
39          statement.executeUpdate();
40          System.out.println("Record is updated!\n");
41      }
42
43      private void listRecord() throws SQLException {
44          String sql = "SELECT PID, LASTNAME, FIRSTNAME FROM PERSON WHERE PID =
45              ?";
46          System.out.println(sql);
47          statement = connection.prepareStatement(sql);
48          statement.setInt(1, 201);
49          ResultSet rs = statement.executeQuery();
50          while (rs.next()) {
51              String pid = rs.getString("PID");
52              String lastName = rs.getString("LASTNAME");
53              String firstName = rs.getString("FIRSTNAME");
54              System.out.println("PID: " + pid);
55              System.out.println("LASTNAME: " + lastName);
56              System.out.println("FIRSTNAME: " + firstName + "\n");
57          }
58      }
59
60      private Connection getDBConnection() throws ClassNotFoundException,
61          SQLException {
62          Class.forName(DB_DRIVER);
63          connection = DriverManager.getConnection(DB_CONNECTION, DB_USER,
64              DB_PASSWORD);
65          return connection;
66      }
67
68      private Timestamp getCurrentTimeStamp() {
69          java.util.Date today = new java.util.Date();
70          return new Timestamp(today.getTime());
71      }
72
73      private void run() throws ClassNotFoundException, SQLException {
74          connection = getDBConnection();
75          try {
76              connection.setAutoCommit(false);
77              insertRecord();
78              listRecord();
79              updateRecord();
80              connection.commit();
81          } catch (SQLException e) {
82              connection.rollback();
83          }
84      }
85  }

```

```

78     }
79     connection.setAutoCommit(true);
80     listRecord();
81     statement.close();
82     connection.close();
83 }
84
85 public static void main(String[] argv) {
86     JdbcOracleTransaction jdbcOracleTransaction = new
87         JdbcOracleTransaction();
88     try {
89         jdbcOracleTransaction.run();
90     } catch (ClassNotFoundException e) {
91         e.printStackTrace();
92     } catch (SQLException e) {
93         e.printStackTrace();
94     }
95 }
```

Result

```

1 INSERT INTO PERSON (PID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE) VALUES
2     (?, ?, ?, ?, ?, ?)
3 Record is inserted!
4
5 SELECT PID, LASTNAME, FIRSTNAME FROM PERSON WHERE PID = ?
6 PID: 201
7 LASTNAME: Smith
8 FIRSTNAME: Johnathan
9
10 UPDATE PERSON SET LASTNAME = ? WHERE PID = ?
11 Record is updated!
12
13 SELECT PID, LASTNAME, FIRSTNAME FROM PERSON WHERE PID = ?
14 PID: 201
15 LASTNAME: Doe
16 FIRSTNAME: Johnathan
```

CHAPTER 14

SPRING DAO

14.1 JDBC

14.1.1 Create project using command line

```
1 c:\workspaces\workspace>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=app-hello-dao-jdbc
  ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

14.1.2 Modify pom.xml

- C:\workspaces\workspace\my-projects\app-hello-dao-jdbc\pom.xml
- Use properties
- Add oracle, spring dependencies
- Modify junit version
- Add maven-compiler-plugin, set JDK version to 1.7

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
```

[†]change c:\workspaces\workspace to /users/xyz/documents/workspace, then all the projects in this chapter will work in mac osx.

```

4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-dao-jdbc</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-dao-jdbc</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <java.version>1.7</java.version>
12    <javax.inject.version>1</javax.inject.version>
13    <junit.version>4.12</junit.version>
14    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
15    <oracle.version>12.1.0.2</oracle.version>
16    <spring.version>4.2.0.RELEASE</spring.version>
17  </properties>
18  <dependencies>
19    <dependency>
20      <groupId>javax.inject</groupId>
21      <artifactId>javax.inject</artifactId>
22      <version>${javax.inject.version}</version>
23    </dependency>
24    <dependency>
25      <groupId>junit</groupId>
26      <artifactId>junit</artifactId>
27      <version>${junit.version}</version>
28      <scope>test</scope>
29    </dependency>
30    <dependency>
31      <groupId>com.oracle</groupId>
32      <artifactId>ojdbc7</artifactId>
33      <version>${oracle.version}</version>
34    </dependency>
35    <dependency>
36      <groupId>org.springframework</groupId>
37      <artifactId>spring-context</artifactId>
38      <version>${spring.version}</version>
39    </dependency>
40    <dependency>
41      <groupId>org.springframework</groupId>
42      <artifactId>spring-core</artifactId>
43      <version>${spring.version}</version>
44    </dependency>
45    <dependency>
46      <groupId>org.springframework</groupId>
47      <artifactId>spring-jdbc</artifactId>
48      <version>${spring.version}</version>
49    </dependency>
50  </dependencies>
51  <build>
52    <finalName>app-hello-dao-jdbc</finalName>
53    <plugins>
54      <plugin>
55        <groupId>org.apache.maven.plugins</groupId>
56        <artifactId>maven-compiler-plugin</artifactId>
57        <version>${maven.compiler.plugin.version}</version>
58        <configuration>
59          <source>${java.version}</source>
60          <target>${java.version}</target>
61        </configuration>
62      </plugin>
63    </plugins>
64  </build>
65 </project>
```

14.1.3 Make this project an eclipse project

```

1 c:\workspaces\workspace\my-projects\app-hello-dao-jdbc>mvn eclipse:clean
  ↵ eclipse:eclipse

```

14.1.4 Import this project to eclipse

- Launch eclipse
File ->Import ->General ->Existing Projects into Workspace
Browse to app-hello-dao-jdbc, and import it.

14.1.5 Add a java model class

- src/main/java ->com.xyz.model ->Employee.java

```

1 package com.xyz.model;
2
3 import java.sql.Timestamp;
4
5 public class Employee {
6     int empId;
7     String lastName;
8     String firstName;
9     String createdBy;
10    Timestamp createdDate;
11
12    public int getEmpId() {
13        return empId;
14    }
15    public void setEmpId(int empId) {
16        this.empId = empId;
17    }
18    public String getLastname() {
19        return lastName;
20    }
21    public void setLastname(String lastName) {
22        this.lastName = lastName;
23    }
24    public String getFirstName() {
25        return firstName;
26    }
27    public void setFirstName(String firstName) {
28        this.firstName = firstName;
29    }
30    public String getCreatedBy() {
31        return createdBy;
32    }
33    public void setCreatedBy(String createdBy) {
34        this.createdBy = createdBy;
35    }
36    public Timestamp getCreatedDate() {
37        return createdDate;
38    }
39    public void setCreatedDate(Timestamp createdDate) {
40        this.createdDate = createdDate;
41    }
42 }
```

14.1.6 Add java dao classes

- src/main/java ->com.xyz.dao ->EmployeeDAO.java

```

1 package com.xyz.dao;
2
3 public interface EmployeeDAO {
4     public void dropTable();
5     public void createTable();
6     public void insertRecord();
7     public void updateRecord();
8     public void deleteRecord();
9     public void listRecord();
```

```

10     public void batchInsertRecords();
11     public void listRecords();
12 }

```

src/main/java -> com.xyz.dao -> EmployeeDAOImpl.java

```

1 package com.xyz.dao;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Timestamp;
8
9 import javax.inject.Inject;
10 import javax.inject.Named;
11 import javax.sql.DataSource;
12
13 @Named
14 public class EmployeeDAOImpl implements EmployeeDAO {
15     @Inject
16     private DataSource dataSource;
17     private Connection connection = null;
18     private PreparedStatement statement = null;
19
20     @Override
21     public void dropTable() {
22         try {
23             connection = dataSource.getConnection();
24             String sql = "BEGIN EXECUTE IMMEDIATE 'DROP TABLE EMPLOYEE';
25             ↪ EXCEPTION WHEN OTHERS THEN NULL; END;";
26             System.out.println(sql);
27             statement = connection.prepareStatement(sql);
28             statement.execute();
29             System.out.println("Table is dropped!\n");
30             statement.close();
31             connection.close();
32         } catch (SQLException e) {
33             e.printStackTrace();
34         }
35     }
36
37     @Override
38     public void createTable() {
39         try {
40             connection = dataSource.getConnection();
41             String sql = "CREATE TABLE EMPLOYEE (EMPID NUMBER(10) NOT NULL,
42             ↪ LASTNAME VARCHAR(20) NOT NULL, FIRSTNAME VARCHAR(20) NOT NULL,
43             ↪ "
44             + "CREATED_BY VARCHAR(20) NOT NULL, CREATED_DATE DATE NOT
45             ↪ NULL, PRIMARY KEY (EMPID)" + ")";
46             System.out.println(sql);
47             statement = connection.prepareStatement(sql);
48             statement.execute();
49             System.out.println("Table is created!\n");
50             statement.close();
51             connection.close();
52         } catch (SQLException e) {
53             e.printStackTrace();
54         }
55     }
56
57     @Override
58     public void insertRecord() {
59         try {
60             connection = dataSource.getConnection();
61             String sql = "INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME,
62             ↪ CREATED_BY, CREATED_DATE) VALUES (?, ?, ?, ?, ?)";
63             System.out.println(sql);
64             statement = connection.prepareStatement(sql);
65             statement.setInt(1, 1);
66             statement.setString(2, "Smith");

```

```

62         statement.setString(3, "John");
63         statement.setString(4, "system");
64         statement.setTimestamp(5, getCurrentTimeStamp());
65         statement.executeUpdate();
66         System.out.println("Record is inserted!\n");
67         statement.close();
68         connection.close();
69     } catch (SQLException e) {
70         e.printStackTrace();
71     }
72 }
73
74 @Override
75 public void updateRecord() {
76     try {
77         connection = dataSource.getConnection();
78         String sql = "UPDATE EMPLOYEE SET LASTNAME = ? WHERE EMPID = ?";
79         System.out.println(sql);
80         statement = connection.prepareStatement(sql);
81         statement.setString(1, "Doe");
82         statement.setInt(2, 1);
83         statement.executeUpdate();
84         System.out.println("Record is updated!\n");
85         statement.close();
86         connection.close();
87     } catch (SQLException e) {
88         e.printStackTrace();
89     }
90 }
91
92 @Override
93 public void deleteRecord() {
94     try {
95         connection = dataSource.getConnection();
96         String sql = "DELETE EMPLOYEE WHERE EMPID = ?";
97         System.out.println(sql);
98         statement = connection.prepareStatement(sql);
99         statement.setInt(1, 1);
100        statement.executeUpdate();
101        System.out.println("Record is deleted!\n");
102        statement.close();
103        connection.close();
104    } catch (SQLException e) {
105        e.printStackTrace();
106    }
107 }
108
109 @Override
110 public void listRecord() {
111     try {
112         connection = dataSource.getConnection();
113         String sql = "SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY,
114             ↪ CREATED_DATE FROM EMPLOYEE WHERE EMPID = ?";
115         System.out.println(sql);
116         statement = connection.prepareStatement(sql);
117         statement.setInt(1, 1);
118         ResultSet rs = statement.executeQuery();
119         while (rs.next()) {
120             int empid = rs.getInt("EMPID");
121             String lastName = rs.getString("LASTNAME");
122             String firstName = rs.getString("FIRSTNAME");
123             String createdBy = rs.getString("CREATED_BY");
124             Timestamp createdDate = rs.getTimestamp("CREATED_DATE");
125             System.out.println("EMPID: " + empid);
126             System.out.println("LASTNAME: " + lastName);
127             System.out.println("FIRSTNAME: " + firstName);
128             System.out.println("CREATED_BY: " + createdBy);
129             System.out.println("CREATED_DATE: " + createdDate + "\n");
130         }
131         statement.close();
132         connection.close();
133     } catch (SQLException e) {
134         e.printStackTrace();
135     }
136 }
```

```

134         }
135     }
136
137     @Override
138     public void batchInsertRecords() {
139         try {
140             connection = dataSource.getConnection();
141             String sql = "INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME,
142                             ↪ CREATED_BY, CREATED_DATE) VALUES (?, ?, ?, ?, ?)";
143             System.out.println(sql);
144             statement = connection.prepareStatement(sql);
145             connection.setAutoCommit(false);
146
147             statement.setInt(1, 1);
148             statement.setString(2, "Smith");
149             statement.setString(3, "John");
150             statement.setString(4, "system");
151             statement.setTimestamp(5, getCurrentTimeStamp());
152             statement.addBatch();
153
154             statement.setInt(1, 2);
155             statement.setString(2, "Doe");
156             statement.setString(3, "Jane");
157             statement.setString(4, "system");
158             statement.setTimestamp(5, getCurrentTimeStamp());
159             statement.addBatch();
160
161             statement.setInt(1, 3);
162             statement.setString(2, "Brown");
163             statement.setString(3, "Charlie");
164             statement.setString(4, "system");
165             statement.setTimestamp(5, getCurrentTimeStamp());
166             statement.addBatch();
167
168             statement.executeBatch();
169             connection.commit();
170             System.out.println("Records are inserted!\n");
171             statement.close();
172             connection.close();
173         } catch (SQLException e) {
174             e.printStackTrace();
175         }
176
177     @Override
178     public void listRecords() {
179         try {
180             connection = dataSource.getConnection();
181             String sql = "SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY,
182                             ↪ CREATED_DATE FROM EMPLOYEE WHERE FIRSTNAME LIKE ?";
183             System.out.println(sql);
184             statement = connection.prepareStatement(sql);
185             statement.setString(1, "J%");
186             ResultSet rs = statement.executeQuery();
187             while (rs.next()) {
188                 int empid = rs.getInt("EMPID");
189                 String lastName = rs.getString("LASTNAME");
190                 String firstName = rs.getString("FIRSTNAME");
191                 String createdBy = rs.getString("CREATED_BY");
192                 Timestamp createdDate = rs.getTimestamp("CREATED_DATE");
193                 System.out.println("EMPID: " + empid);
194                 System.out.println("LASTNAME: " + lastName);
195                 System.out.println("FIRSTNAME: " + firstName);
196                 System.out.println("CREATED_BY: " + createdBy);
197                 System.out.println("CREATED_DATE: " + createdDate + "\n");
198             }
199             statement.close();
200             connection.close();
201         } catch (SQLException e) {
202             e.printStackTrace();
203         }
204     }

```

```

205     private Timestamp getCurrentTimeStamp() {
206         java.util.Date today = new java.util.Date();
207         return new Timestamp(today.getTime());
208     }
209 }
```

14.1.7 Configure spring bean

- Create resources directory and add it to Build Path
src/main/resources
- Configuration metadata
src/main/resources/app-hello-dao-jdbc.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4          xmlns:context="http://www.springframework.org/schema/context"
5          xsi:schemaLocation="http://www.springframework.org/schema/beans
6              http://www.springframework.org/schema/beans/spring-beans.xsd
7              http://www.springframework.org/schema/context
8                  http://www.springframework.org/schema/context/spring-context.xsd">
9
10 <context:annotation-config />
11 <context:component-scan base-package="com.xyz" />
12 <bean id="employeeDAO" class="com.xyz.dao.EmployeeDAOImpl" />
13
14 <bean id="dataSource"
15     class="org.springframework.jdbc.datasource.DriverManagerDataSource">
16     <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />
17     <property name="url" value="jdbc:oracle:thin:@localhost:1522:orcl" />
18     <property name="username" value="system" />
19     <property name="password" value="password" />
20 </bean>
21
22 </beans>
```

14.1.8 Instantiating a container

- App
Type in the code list below to src/main/java -> com.xyz -> App.java

```

1 package com.xyz;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5
6 import com.xyz.dao.EmployeeDAO;
7
8 public class App {
9     public static void main(String[] args) {
10         ApplicationContext context = new
11             ClassPathXmlApplicationContext("app-hello-dao-jdbc.xml");
12         EmployeeDAO dao = (EmployeeDAO) context.getBean("employeeDAO");
13         dao.dropTable();
14         dao.createTable();
15         dao.insertRecord();
16         dao.listRecord();
17         dao.updateRecord();
18         dao.listRecord();
19         dao.deleteRecord();
20         dao.listRecord();
21         dao.batchInsertRecords();
22         dao.listRecords();
```

```

22         ((ClassPathXmlApplicationContext) context).close();
23     }
24 }
```

14.1.9 Run App

Right click on project src/main/java ->com.xyz->App

Run As ->Java Application

You will see in the console:

```

1 BEGIN EXECUTE IMMEDIATE 'DROP TABLE EMPLOYEE'; EXCEPTION WHEN OTHERS THEN NULL;
2   ↑ END;
3 Table is dropped!
4 CREATE TABLE EMPLOYEE(EMPID NUMBER(10) NOT NULL, LASTNAME VARCHAR(20) NOT NULL,
5   ↑ FIRSTNAME VARCHAR(20) NOT NULL, CREATED_BY VARCHAR(20) NOT NULL,
6   ↑ CREATED_DATE DATE NOT NULL, PRIMARY KEY (EMPID))
7 Table is created!
8
9 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
10   ↑ VALUES (?, ?, ?, ?, ?)
11 Record is inserted!
12
13 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
14   ↑ EMPID = ?
15 EMPID: 1
16 LASTNAME: Smith
17 FIRSTNAME: John
18 CREATED_BY: system
19 CREATED_DATE: 2015-12-17 16:47:35.0
20
21 UPDATE EMPLOYEE SET LASTNAME = ? WHERE EMPID = ?
22 Record is updated!
23
24 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
25   ↑ EMPID = ?
26 EMPID: 1
27 LASTNAME: Doe
28 FIRSTNAME: John
29 CREATED_BY: system
30 CREATED_DATE: 2015-12-17 16:47:35.0
31
32 DELETE EMPLOYEE WHERE EMPID = ?
33 Record is deleted!
34
35 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
36   ↑ EMPID = ?
37 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
38   ↑ VALUES (?, ?, ?, ?, ?)
39 Records are inserted!
40
41 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
42   ↑ FIRSTNAME LIKE ?
43 EMPID: 1
44 LASTNAME: Smith
45 FIRSTNAME: John
46 CREATED_BY: system
47 CREATED_DATE: 2015-12-17 16:47:36.0
48
49 EMPID: 2
50 LASTNAME: Doe
51 FIRSTNAME: Jane
52 CREATED_BY: system
53 CREATED_DATE: 2015-12-17 16:47:36.0
```

14.2 Spring Framework JDBC

14.2.1 JdbcTemplate

- Configuration metadata

src/main/resources/app-hello-dao-jdbc.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     <!-- xmlns:context="http://www.springframework.org/schema/context"
5       xsi:schemaLocation="http://www.springframework.org/schema/beans
6         http://www.springframework.org/schema/beans/spring-beans.xsd
7         http://www.springframework.org/schema/context
8           http://www.springframework.org/schema/context/spring-context.xsd"-->
9 
10 <context:annotation-config />
11 <context:component-scan base-package="com.xyz" />
12 
13 <bean id="employeeDAO" class="com.xyz.dao.EmployeeDAOImpl" />
14 
15 <bean id="employeeDAOJdbcTemplate"
16   <!-- class="com.xyz.dao.EmployeeDAOJdbcTemplateImpl" /-->
17 
18 <bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
19   <property name="dataSource" ref="dataSource" />
20 
21 <bean id="dataSource"
22   class="org.springframework.jdbc.datasource.DriverManagerDataSource">
23   <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />
24   <property name="url" value="jdbc:oracle:thin:@localhost:1522:orcl" />
25   <property name="username" value="system" />
26   <property name="password" value="password" />
27 
28 </beans>
```

- DAO Implementation

src/main/java -> com.xyz.dao -> EmployeeDAOJdbcTemplateImpl.java

```

1 package com.xyz.dao;
2 
3 import java.sql.PreparedStatement;
4 import java.sql.ResultSet;
5 import java.sql.SQLException;
6 import java.sql.Timestamp;
7 import java.util.ArrayList;
8 import java.util.List;
9 
10 import javax.inject.Inject;
11 import javax.inject.Named;
12 
13 import org.springframework.jdbc.core.BatchPreparedStatementSetter;
14 import org.springframework.jdbc.core.JdbcTemplate;
15 import org.springframework.jdbc.core.RowMapper;
16 
17 import com.xyz.model.Employee;
18 
19 @Named
20 public class EmployeeDAOJdbcTemplateImpl implements EmployeeDAO {
21   @Inject
22   private JdbcTemplate jdbcTemplate;
23 
24   @Override
25   public void dropTable() {
26     String sql = "BEGIN EXECUTE IMMEDIATE 'DROP TABLE EMPLOYEE'; EXCEPTION
27       WHEN OTHERS THEN NULL; END;";
```

```

27         System.out.println(sql);
28         jdbcTemplate.execute(sql);
29         System.out.println("Table is dropped!\n");
30     }
31
32     @Override
33     public void createTable() {
34         String sql = "CREATE TABLE EMPLOYEE (EMPID NUMBER(10) NOT NULL, LASTNAME
35             ↪ VARCHAR(20) NOT NULL, FIRSTNAME VARCHAR(20) NOT NULL, "
36             + "CREATED_BY VARCHAR(20) NOT NULL, CREATED_DATE DATE NOT NULL,
37             ↪ PRIMARY KEY (EMPID) " + ")";
38         System.out.println(sql);
39         jdbcTemplate.execute(sql);
40         System.out.println("Table is created!\n");
41     }
42
43     @Override
44     public void insertRecord() {
45         String sql = "INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME,
46             ↪ CREATED_BY, CREATED_DATE) VALUES (?, ?, ?, ?, ?)";
47         System.out.println(sql);
48         jdbcTemplate.update(sql, 1, "Smith", "John", "system",
49             ↪ getCurrentTimeStamp());
50         System.out.println("Record is inserted!\n");
51     }
52
53     @Override
54     public void updateRecord() {
55         String sql = "UPDATE EMPLOYEE SET LASTNAME = ? WHERE EMPID = ?";
56         System.out.println(sql);
57         jdbcTemplate.update(sql, "Doe", 1);
58         System.out.println("Record is updated!\n");
59     }
60
61     @Override
62     public void deleteRecord() {
63         String sql = "DELETE EMPLOYEE WHERE EMPID = ?";
64         System.out.println(sql);
65         jdbcTemplate.update(sql, 1);
66         System.out.println("Record is deleted!\n");
67     }
68
69     @Override
70     public void listRecord() {
71         int rowCount = jdbcTemplate.queryForObject("SELECT COUNT(*) FROM
72             ↪ EMPLOYEE", Integer.class);
73         System.out.println("EMPLOYEE Row Count: " + rowCount);
74         if (rowCount > 0) {
75             String sql = "SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY,
76                 ↪ CREATED_DATE FROM EMPLOYEE WHERE EMPID = ?";
77             System.out.println(sql);
78             Employee employee = jdbcTemplate.queryForObject(sql, new Object[] {
79                 ↪ 1 }, new RowMapper<Employee>() {
80                 public Employee mapRow(ResultSet rs, int rowNum) throws
81                     ↪ SQLException {
82                     Employee employee = new Employee();
83                     employee.setEmpId(rs.getInt("EMPID"));
84                     employee.setLastName(rs.getString("LASTNAME"));
85                     employee.setFirstName(rs.getString("FIRSTNAME"));
86                     employee.setCreatedBy(rs.getString("CREATED_BY"));
87                     employee.setCreatedDate(rs.getTimestamp("CREATED_DATE"));
88                     return employee;
89                 }
90             });
91             System.out.println("EMPID: " + employee.getEmpId());
92             System.out.println("LASTNAME: " + employee.getLastName());
93             System.out.println("FIRSTNAME: " + employee.getFirstName());
94             System.out.println("CREATED_BY: " + employee.getCreatedBy());
95             System.out.println("CREATED_DATE: " + employee.getCreatedDate() +
96                 ↪ "\n");
97         }
98     }
99 }
```

```

91  @Override
92  public void batchInsertRecords() {
93      Employee employee1 = new Employee();
94      employee1.setEmpId(1);
95      employee1.setLastName("Smith");
96      employee1.setFirstName("John");
97      employee1.setCreatedBy("system");
98      employee1.setCreatedDate(getCurrentTimeStamp());
99
100     Employee employee2 = new Employee();
101     employee2.setEmpId(2);
102     employee2.setLastName("Doe");
103     employee2.setFirstName("Jane");
104     employee2.setCreatedBy("system");
105     employee2.setCreatedDate(getCurrentTimeStamp());
106
107     Employee employee3 = new Employee();
108     employee3.setEmpId(3);
109     employee3.setLastName("Brown");
110     employee3.setFirstName("Charlie");
111     employee3.setCreatedBy("system");
112     employee3.setCreatedDate(getCurrentTimeStamp());
113
114     final List<Employee> employees = new ArrayList<Employee>();
115     employees.add(employee1);
116     employees.add(employee2);
117     employees.add(employee3);
118
119     String sql = "INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME,
120         ↪ CREATED_BY, CREATED_DATE) VALUES (?, ?, ?, ?, ?)";
121     System.out.println(sql);
122
123     jdbcTemplate.batchUpdate(sql, new BatchPreparedStatementSetter() {
124         @Override
125         public void setValues(PreparedStatement ps, int i) throws
126             SQLException {
127                 Employee employee = employees.get(i);
128                 ps.setInt(1, employee.getEmpId());
129                 ps.setString(2, employee.getLastName());
130                 ps.setString(3, employee.getFirstName());
131                 ps.setString(4, employee.getCreatedBy());
132                 ps.setTimestamp(5, employee.getCreatedDate());
133
134             @Override
135             public int getBatchSize() {
136                 return employees.size();
137             }
138         });
139
140         System.out.println("Records are inserted!\n");
141     }
142
143     @Override
144     public void listRecords() {
145         String sql = "SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY,
146             ↪ CREATED_DATE FROM EMPLOYEE WHERE FIRSTNAME LIKE ?";
147         System.out.println(sql);
148         List<Employee> employees = jdbcTemplate.query(sql, new Object[] { "%J%" },
149             new RowMapper<Employee>() {
150                 public Employee mapRow(ResultSet rs, int rowNum) throws SQLException
151                 {
152                     Employee employee = new Employee();
153                     employee.setEmpId(rs.getInt("EMPID"));
154                     employee.setLastName(rs.getString("LASTNAME"));
155                     employee.setFirstName(rs.getString("FIRSTNAME"));
156                     employee.setCreatedBy(rs.getString("CREATED_BY"));
157                     employee.setCreatedDate(rs.getTimestamp("CREATED_DATE"));
158                     return employee;
159                 }
160             });
161         for (Employee employee : employees) {
162             System.out.println("EMPID: " + employee.getEmpId());
163         }
164     }

```

```

159         System.out.println("LASTNAME: " + employee.getLastName());
160         System.out.println("FIRSTNAME: " + employee.getFirstName());
161         System.out.println("CREATED_BY: " + employee.getCreatedBy());
162         System.out.println("CREATED_DATE: " + employee.getCreatedDate() +
163             "\n");
164     }
165
166     private Timestamp getCurrentTimeStamp() {
167         java.util.Date today = new java.util.Date();
168         return new Timestamp(today.getTime());
169     }
170 }

```

■ App

src/main/java ->com.xyz->AppJdbcTemplate.java

```

1 package com.xyz;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5
6 import com.xyz.dao.EmployeeDAO;
7
8 public class AppJdbcTemplate {
9     public static void main(String[] args) {
10         ApplicationContext context = new
11             ClassPathXmlApplicationContext("app-hello-dao-jdbc.xml");
12         EmployeeDAO dao = (EmployeeDAO)
13             context.getBean("employeeDAOJdbcTemplate");
14         dao.dropTable();
15         dao.createTable();
16         dao.insertRecord();
17         dao.listRecord();
18         dao.updateRecord();
19         dao.listRecord();
20         dao.deleteRecord();
21         dao.listRecord();
22         dao.batchInsertRecords();
23         dao.listRecords();
24         ((ClassPathXmlApplicationContext) context).close();
25     }
26 }

```

■ Run App

Right click on project src/main/java ->com.xyz->AppJdbcTemplate

Run As ->Java Application

You will see in the console:

```

1 BEGIN EXECUTE IMMEDIATE 'DROP TABLE EMPLOYEE'; EXCEPTION WHEN OTHERS THEN NULL;
2     → END;
3 Table is dropped!
4
4 CREATE TABLE EMPLOYEE(EMPID NUMBER(10) NOT NULL, LASTNAME VARCHAR(20) NOT NULL,
5     → FIRSTNAME VARCHAR(20) NOT NULL, CREATED_BY VARCHAR(20) NOT NULL,
5     → CREATED_DATE DATE NOT NULL, PRIMARY KEY (EMPID))
5 Table is created!
6
7 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
8     → VALUES (?, ?, ?, ?, ?)
8 Record is inserted!
9
10 EMPLOYEE Row Count: 1
11 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
11     → EMPID = ?
12 EMPID: 1
13 LASTNAME: Smith
14 FIRSTNAME: John
15 CREATED_BY: system

```

```

16  CREATED_DATE: 2015-12-17 16:47:54.0
17
18 UPDATE EMPLOYEE SET LASTNAME = ? WHERE EMPID = ?
19 Record is updated!
20
21 EMPLOYEE Row Count: 1
22 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
   ↪   EMPID = ?
23 EMPID: 1
24 LASTNAME: Doe
25 FIRSTNAME: John
26 CREATED_BY: system
27 CREATED_DATE: 2015-12-17 16:47:54.0
28
29 DELETE EMPLOYEE WHERE EMPID = ?
30 Record is deleted!
31
32 EMPLOYEE Row Count: 0
33 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
   ↪   VALUES (?, ?, ?, ?, ?)
34 Records are inserted!
35
36 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
   ↪   FIRSTNAME LIKE ?
37 EMPID: 1
38 LASTNAME: Smith
39 FIRSTNAME: John
40 CREATED_BY: system
41 CREATED_DATE: 2015-12-17 16:47:55.0
42
43 EMPID: 2
44 LASTNAME: Doe
45 FIRSTNAME: Jane
46 CREATED_BY: system
47 CREATED_DATE: 2015-12-17 16:47:55.0

```

14.2.2 NamedParameterJdbcTemplate

- Configuration metadata

src/main/resources/app-hello-dao-jdbc.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     ↪   xmlns:context="http://www.springframework.org/schema/context"
5   xsi:schemaLocation="http://www.springframework.org/schema/beans
6       http://www.springframework.org/schema/beans/spring-beans.xsd
7       http://www.springframework.org/schema/context
8       http://www.springframework.org/schema/context/spring-context.xsd">
9
10 <context:annotation-config />
11
12 <context:component-scan base-package="com.xyz" />
13
14 <bean id="employeeDAO" class="com.xyz.dao.EmployeeDAOImpl" />
15
16 <bean id="employeeDAOJdbcTemplate"
17   ↪   class="com.xyz.dao.EmployeeDAOJdbcTemplateImpl" />
18
19 <bean id="employeeDAONamedParameterJdbcTemplate"
20   ↪   class="com.xyz.dao.EmployeeDAONamedParameterJdbcTemplateImpl" />
21
22 <bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
23   <property name="dataSource" ref="dataSource" />
24 </bean>

```

```

25   </bean>
26
27   <bean id="dataSource"
28     class="org.springframework.jdbc.datasource.DriverManagerDataSource">
29     <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />
30     <property name="url" value="jdbc:oracle:thin:@localhost:1522:orcl" />
31     <property name="username" value="system" />
32     <property name="password" value="password" />
33   </bean>
34
35 </beans>

```

- DAO Implementation

src/main/java -> com.xyz.dao -> EmployeeDAONamedParameterJdbcTemplateImpl.java

```

1 package com.xyz.dao;
2
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5 import java.sql.Timestamp;
6 import java.util.ArrayList;
7 import java.util.HashMap;
8 import java.util.List;
9 import java.util.Map;
10
11 import javax.inject.Inject;
12 import javax.inject.Named;
13
14 import org.springframework.jdbc.core.JdbcTemplate;
15 import org.springframework.jdbc.core.RowMapper;
16 import org.springframework.jdbc.core.namedparam.BeanPropertySqlParameterSource;
17 import org.springframework.jdbc.core.namedparam.MapSqlParameterSource;
18 import org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate;
19 import org.springframework.jdbc.core.namedparam.SqlParameterSource;
20
21 import com.xyz.model.Employee;
22
23 @Named
24 public class EmployeeDAONamedParameterJdbcTemplateImpl implements EmployeeDAO {
25     @Inject
26     private JdbcTemplate jdbcTemplate;
27     @Inject
28     private NamedParameterJdbcTemplate namedParameterJdbcTemplate;
29
30     @Override
31     public void dropTable() {
32         String sql = "BEGIN EXECUTE IMMEDIATE 'DROP TABLE EMPLOYEE'; EXCEPTION
33             WHEN OTHERS THEN NULL; END;";
34         System.out.println(sql);
35         SqlParameterSource parameters = new MapSqlParameterSource();
36         namedParameterJdbcTemplate.update(sql, parameters);
37         System.out.println("Table is dropped!\n");
38     }
39
40     @Override
41     public void createTable() {
42         String sql = "CREATE TABLE EMPLOYEE(EMPID NUMBER(10) NOT NULL, LASTNAME
43             → VARCHAR(20) NOT NULL, FIRSTNAME VARCHAR(20) NOT NULL, "
44             + "CREATED_BY VARCHAR(20) NOT NULL, CREATED_DATE DATE NOT NULL,
45             → PRIMARY KEY (EMPID) " + ")";
46         System.out.println(sql);
47         SqlParameterSource parameters = new MapSqlParameterSource();
48         namedParameterJdbcTemplate.update(sql, parameters);
49         System.out.println("Table is created!\n");
50     }
51
52     @Override
53     public void insertRecord() {
54         String sql = "INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME,
55             → CREATED_BY, CREATED_DATE) VALUES (:empId, :lastName, :firstName,
56             → :createdBy, :createdDate)";

```

```

52     System.out.println(sql);
53     Map<String, Object> parameters = new HashMap<String, Object>();
54     parameters.put("empId", new Integer(1));
55     parameters.put("lastName", "Smith");
56     parameters.put("firstName", "John");
57     parameters.put("createdBy", "system");
58     parameters.put("createdDate", getCurrentTimeStamp());
59     namedParameterJdbcTemplate.update(sql, parameters);
60     System.out.println("Record is inserted!\n");
61 }
62
63 @Override
64 public void updateRecord() {
65     String sql = "UPDATE EMPLOYEE SET LASTNAME = :lastName WHERE EMPID =
66         <:empId";
67     System.out.println(sql);
68     Map<String, Object> parameters = new HashMap<String, Object>();
69     parameters.put("lastName", "Doe");
70     parameters.put("empId", new Integer(1));
71     namedParameterJdbcTemplate.update(sql, parameters);
72     System.out.println("Record is updated!\n");
73 }
74
75 @Override
76 public void deleteRecord() {
77     String sql = "DELETE EMPLOYEE WHERE EMPID = :empId";
78     System.out.println(sql);
79     Map<String, Object> parameters = new HashMap<String, Object>();
80     parameters.put("empId", new Integer(1));
81     namedParameterJdbcTemplate.update(sql, parameters);
82     System.out.println("Record is deleted!\n");
83 }
84
85 @Override
86 public void listRecord() {
87     int rowCount = jdbcTemplate.queryForObject("SELECT COUNT(*) FROM
88         EMPLOYEE", Integer.class);
89     System.out.println("EMPLOYEE Row Count: " + rowCount);
90     if (rowCount > 0) {
91         String sql = "SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY,
92             <CREATED_DATE FROM EMPLOYEE WHERE EMPID = :empId";
93         SqlParameterSource parameters = new MapSqlParameterSource("empId",
94             < Integer.valueOf(1));
95         System.out.println(sql);
96         Employee employee = namedParameterJdbcTemplate.queryForObject(sql,
97             < parameters, new RowMapper<Employee>() {
98             public Employee mapRow(ResultSet rs, int rowNum) throws
99                 SQLException {
100                 Employee employee = new Employee();
101                 employee.setEmpId(rs.getInt("EMPID"));
102                 employee.setLastName(rs.getString("LASTNAME"));
103                 employee.setFirstName(rs.getString("FIRSTNAME"));
104                 employee.setCreatedBy(rs.getString("CREATED_BY"));
105                 employee.setCreatedDate(rs.getTimestamp("CREATED_DATE"));
106                 return employee;
107             }
108         });
109     }
110
111 @Override
112 public void batchInsertRecords() {
113     Employee employee1 = new Employee();
114     employee1.setEmpId(1);
115     employee1.setLastName("Smith");
116     employee1.setFirstName("John");
117     employee1.setCreatedBy("system");

```

```

118     employee1.setCreatedDate(getCurrentTimeStamp());
119
120     Employee employee2 = new Employee();
121     employee2.setEmpId(2);
122     employee2.setLastName("Doe");
123     employee2.setFirstName("Jane");
124     employee2.setCreatedBy("system");
125     employee2.setCreatedDate(getCurrentTimeStamp());
126
127     Employee employee3 = new Employee();
128     employee3.setEmpId(3);
129     employee3.setLastName("Brown");
130     employee3.setFirstName("Charlie");
131     employee3.setCreatedBy("system");
132     employee3.setCreatedDate(getCurrentTimeStamp());
133
134     final List<Employee> employees = new ArrayList<Employee>();
135     employees.add(employee1);
136     employees.add(employee2);
137     employees.add(employee3);
138
139     String sql = "INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME,
140         ↪ CREATED_BY, CREATED_DATE) VALUES (:empId, :lastName, :firstName,
141         ↪ :createdBy, :createdDate)";
142     System.out.println(sql);
143     List<SqlParameterSource> parameters = new
144         ↪ ArrayList<SqlParameterSource>();
145     for (Employee employee : employees) {
146         parameters.add(new BeanPropertySqlParameterSource(employee));
147     }
148     namedParameterJdbcTemplate.batchUpdate(sql, parameters.toArray(new
149         ↪ SqlParameterSource[0]));
150     System.out.println("Records are inserted!\n");
151
152     @Override
153     public void listRecords() {
154         String sql = "SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY,
155             ↪ CREATED_DATE FROM EMPLOYEE WHERE FIRSTNAME LIKE :firstName";
156         System.out.println(sql);
157         SqlParameterSource parameters = new MapSqlParameterSource("firstName",
158             ↪ "J%");
159         List<Employee> employees = namedParameterJdbcTemplate.query(sql,
160             ↪ parameters, new RowMapper<Employee>() {
161                 public Employee mapRow(ResultSet rs, int rowNum) throws SQLException
162                     ↪ {
163                         Employee employee = new Employee();
164                         employee.setEmpId(rs.getInt("EMPID"));
165                         employee.setLastName(rs.getString("LASTNAME"));
166                         employee.setFirstName(rs.getString("FIRSTNAME"));
167                         employee.setCreatedBy(rs.getString("CREATED_BY"));
168                         employee.setCreatedDate(rs.getTimestamp("CREATED_DATE"));
169                         return employee;
170                     }
171             });
172         for (Employee employee : employees) {
173             System.out.println("EMPID: " + employee.getEmpId());
174             System.out.println("LASTNAME: " + employee.getLastname());
175             System.out.println("FIRSTNAME: " + employee.getFirstname());
176             System.out.println("CREATED_BY: " + employee.getCreatedBy());
177             System.out.println("CREATED_DATE: " + employee.getCreatedDate() +
178                 ↪ "\n");
179         }
180     }
181
182     private Timestamp getCurrentTimeStamp() {
183         java.util.Date today = new java.util.Date();
184         return new Timestamp(today.getTime());
185     }
186 }

```

- App

src/main/java -> com.xyz -> AppNamedParameterJdbcTemplate.java

```

1 package com.xyz;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5
6 import com.xyz.dao.EmployeeDAO;
7
8 public class AppNamedParameterJdbcTemplate {
9     public static void main(String[] args) {
10         ApplicationContext context = new
11             ClassPathXmlApplicationContext("app-hello-dao-jdbc.xml");
12         EmployeeDAO dao = (EmployeeDAO)
13             context.getBean("employeeDAONamedParameterJdbcTemplate");
14         dao.dropTable();
15         dao.createTable();
16         dao.insertRecord();
17         dao.listRecord();
18         dao.updateRecord();
19         dao.listRecord();
20         dao.deleteRecord();
21         dao.listRecord();
22         dao.batchInsertRecords();
23         dao.listRecords();
24     }
}

```

- Run App

Right click on project src/main/java -> com.xyz -> AppNamedParameterJdbcTemplate

Run As -> Java Application

You will see in the console:

```

1 BEGIN EXECUTE IMMEDIATE 'DROP TABLE EMPLOYEE'; EXCEPTION WHEN OTHERS THEN NULL;
2   ↪ END;
3 Table is dropped!
4
4 CREATE TABLE EMPLOYEE(EMPID NUMBER(10) NOT NULL, LASTNAME VARCHAR(20) NOT NULL,
5   ↪ FIRSTNAME VARCHAR(20) NOT NULL, CREATED_BY VARCHAR(20) NOT NULL,
5   ↪ CREATED_DATE DATE NOT NULL, PRIMARY KEY (EMPID))
5 Table is created!
6
7 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
8   ↪ VALUES (:empId, :lastName, :firstName, :createdBy, :createdDate)
8 Record is inserted!
9
10 EMPLOYEE Row Count: 1
11 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
12   ↪ EMPID = :empId
12 EMPID: 1
13 LASTNAME: Smith
14 FIRSTNAME: John
15 CREATED_BY: system
16 CREATED_DATE: 2015-12-18 15:53:08.0
17
18 UPDATE EMPLOYEE SET LASTNAME = :lastName WHERE EMPID = :empId
19 Record is updated!
20
21 EMPLOYEE Row Count: 1
22 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
23   ↪ EMPID = :empId
23 EMPID: 1
24 LASTNAME: Doe
25 FIRSTNAME: John
26 CREATED_BY: system
27 CREATED_DATE: 2015-12-18 15:53:08.0
28
29 DELETE EMPLOYEE WHERE EMPID = :empId

```

```

30 Record is deleted!
31
32 EMPLOYEE Row Count: 0
33 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
   ↪ VALUES (:empId, :lastName, :firstName, :createdBy, :createdDate)
34 Records are inserted!
35
36 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
   ↪ FIRSTNAME LIKE :firstName
37 EMPID: 1
38 LASTNAME: Smith
39 FIRSTNAME: John
40 CREATED_BY: system
41 CREATED_DATE: 2015-12-18 15:53:09.0
42
43 EMPID: 2
44 LASTNAME: Doe
45 FIRSTNAME: Jane
46 CREATED_BY: system
47 CREATED_DATE: 2015-12-18 15:53:09.0

```

14.3 Embedded Database

14.3.1 Create project using command line

```

1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↪ -DartifactId=app-hello-dao-embedded
  ↪ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

```

14.3.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\app-hello-dao-embedded\pom.xml

Use properties

Add derby, hsql, h2, spring dependencies

Modify junit version

Add maven-compiler-plugin, set JDK version to 1.7

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↪ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  ↪ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-dao-embedded</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-dao-embedded</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <derby.version>10.12.1.1</derby.version>
12    <hsqldb.version>2.3.3</hsqldb.version>
13    <h2.version>1.4.190</h2.version>
14    <java.version>1.7</java.version>
15    <javax.inject.version>1</javax.inject.version>
16    <junit.version>4.12</junit.version>
17    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
18    <spring.version>4.2.0.RELEASE</spring.version>
19  </properties>
20  <dependencies>
21    <dependency>
22      <groupId>org.apache.derby</groupId>

```

```

23      <artifactId>derby</artifactId>
24      <version>${derby.version}</version>
25    </dependency>
26    <dependency>
27      <groupId>org.hsqldb</groupId>
28      <artifactId>hsqldb</artifactId>
29      <version>${hsqldb.version}</version>
30    </dependency>
31    <dependency>
32      <groupId>com.h2database</groupId>
33      <artifactId>h2</artifactId>
34      <version>${h2.version}</version>
35    </dependency>
36    <dependency>
37      <groupId>javax.inject</groupId>
38      <artifactId>javax.inject</artifactId>
39      <version>${javax.inject.version}</version>
40    </dependency>
41    <dependency>
42      <groupId>junit</groupId>
43      <artifactId>junit</artifactId>
44      <version>${junit.version}</version>
45      <scope>test</scope>
46    </dependency>
47    <dependency>
48      <groupId>org.springframework</groupId>
49      <artifactId>spring-context</artifactId>
50      <version>${spring.version}</version>
51    </dependency>
52    <dependency>
53      <groupId>org.springframework</groupId>
54      <artifactId>spring-core</artifactId>
55      <version>${spring.version}</version>
56    </dependency>
57    <dependency>
58      <groupId>org.springframework</groupId>
59      <artifactId>spring-jdbc</artifactId>
60      <version>${spring.version}</version>
61    </dependency>
62  </dependencies>
63  <build>
64    <finalName>app-hello-dao-embedded</finalName>
65    <plugins>
66      <plugin>
67        <groupId>org.apache.maven.plugins</groupId>
68        <artifactId>maven-compiler-plugin</artifactId>
69        <version>${maven.compiler.plugin.version}</version>
70        <configuration>
71          <source>${java.version}</source>
72          <target>${java.version}</target>
73        </configuration>
74      </plugin>
75    </plugins>
76  </build>
77</project>
```

14.3.3 Make this project an eclipse project

```

1 c:\workspaces\workspace\my-projects\app-hello-dao-embedded>mvn eclipse:clean
  ↵ eclipse:eclipse

```

14.3.4 Import this project to eclipse

- Launch eclipse
- File → Import → General → Existing Projects into Workspace
Browse to app-hello-dao-embedded, and import it.

14.3.5 Add a java model class

- src/main/java -> com.xyz.model -> Employee.java

```

1 package com.xyz.model;
2
3 import java.sql.Timestamp;
4
5 public class Employee {
6     int empId;
7     String lastName;
8     String firstName;
9     String createdBy;
10    Timestamp createdDate;
11
12    public int getEmpId() {
13        return empId;
14    }
15    public void setEmpId(int empId) {
16        this.empId = empId;
17    }
18    public String getLastname() {
19        return lastName;
20    }
21    public void setLastname(String lastName) {
22        this.lastName = lastName;
23    }
24    public String getFirstName() {
25        return firstName;
26    }
27    public void setFirstName(String firstName) {
28        this.firstName = firstName;
29    }
30    public String getCreatedBy() {
31        return createdBy;
32    }
33    public void setCreatedBy(String createdBy) {
34        this.createdBy = createdBy;
35    }
36    public Timestamp getCreatedDate() {
37        return createdDate;
38    }
39    public void setCreatedDate(Timestamp createdDate) {
40        this.createdDate = createdDate;
41    }
42}

```

14.3.6 Add java dao classes

- src/main/java -> com.xyz.dao -> EmployeeDAO.java

```

1 package com.xyz.dao;
2
3 public interface EmployeeDAO {
4     public void insertRecord();
5     public void updateRecord();
6     public void deleteRecord();
7     public void listRecord();
8     public void batchInsertRecords();
9     public void listRecords();
10}

```

src/main/java -> com.xyz.dao -> EmployeeDAOImpl.java

```

1 package com.xyz.dao;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;

```

```

5   import java.sql.ResultSet;
6   import java.sql.SQLException;
7   import java.sql.Timestamp;
8
9   import javax.inject.Inject;
10  import javax.inject.Named;
11  import javax.sql.DataSource;
12
13 @Named
14 public class EmployeeDAOImpl implements EmployeeDAO {
15     @Inject
16     private DataSource dataSource;
17     private Connection connection = null;
18     private PreparedStatement statement = null;
19
20     @Override
21     public void insertRecord() {
22         try {
23             connection = dataSource.getConnection();
24             String sql = "INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME,
25                         ↪ CREATED_BY, CREATED_DATE) VALUES (?, ?, ?, ?, ?, ?)";
26             System.out.println(sql);
27             statement = connection.prepareStatement(sql);
28             statement.setInt(1, 1);
29             statement.setString(2, "Smith");
30             statement.setString(3, "John");
31             statement.setString(4, "system");
32             statement.setTimestamp(5, getCurrentTimeStamp());
33             statement.executeUpdate();
34             System.out.println("Record is inserted!\n");
35             statement.close();
36             connection.close();
37         } catch (SQLException e) {
38             e.printStackTrace();
39         }
40     }
41
42     @Override
43     public void updateRecord() {
44         try {
45             connection = dataSource.getConnection();
46             String sql = "UPDATE EMPLOYEE SET LASTNAME = ? WHERE EMPID = ?";
47             System.out.println(sql);
48             statement = connection.prepareStatement(sql);
49             statement.setString(1, "Doe");
50             statement.setInt(2, 1);
51             statement.executeUpdate();
52             System.out.println("Record is updated!\n");
53             statement.close();
54             connection.close();
55         } catch (SQLException e) {
56             e.printStackTrace();
57         }
58     }
59
60     @Override
61     public void deleteRecord() {
62         try {
63             connection = dataSource.getConnection();
64             String sql = "DELETE FROM EMPLOYEE WHERE EMPID = ?";
65             System.out.println(sql);
66             statement = connection.prepareStatement(sql);
67             statement.setInt(1, 1);
68             statement.executeUpdate();
69             System.out.println("Record is deleted!\n");
70             statement.close();
71             connection.close();
72         } catch (SQLException e) {
73             e.printStackTrace();
74         }
75     }
76     @Override

```

```

77  public void listRecord() {
78      try {
79          connection = dataSource.getConnection();
80          String sql = "SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY,
81                         ↪ CREATED_DATE FROM EMPLOYEE WHERE EMPID = ?";
82          System.out.println(sql);
83          statement = connection.prepareStatement(sql);
84          statement.setInt(1, 1);
85          ResultSet rs = statement.executeQuery();
86          while (rs.next()) {
87              int empid = rs.getInt("EMPID");
88              String lastName = rs.getString("LASTNAME");
89              String firstName = rs.getString("FIRSTNAME");
90              String createdBy = rs.getString("CREATED_BY");
91              Timestamp createdDate = rs.getTimestamp("CREATED_DATE");
92              System.out.println("EMPID: " + empid);
93              System.out.println("LASTNAME: " + lastName);
94              System.out.println("FIRSTNAME: " + firstName);
95              System.out.println("CREATED_BY: " + createdBy);
96              System.out.println("CREATED_DATE: " + createdDate + "\n");
97          }
98          statement.close();
99          connection.close();
100     } catch (SQLException e) {
101         e.printStackTrace();
102     }
103 }
104 @Override
105 public void batchInsertRecords() {
106     try {
107         connection = dataSource.getConnection();
108         String sql = "INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME,
109                         ↪ CREATED_BY, CREATED_DATE) VALUES (?, ?, ?, ?, ?)";
110         System.out.println(sql);
111         statement = connection.prepareStatement(sql);
112         connection.setAutoCommit(false);
113
114         statement.setInt(1, 1);
115         statement.setString(2, "Smith");
116         statement.setString(3, "John");
117         statement.setString(4, "system");
118         statement.setTimestamp(5, getCurrentTimeStamp());
119         statement.addBatch();
120
121         statement.setInt(1, 2);
122         statement.setString(2, "Doe");
123         statement.setString(3, "Jane");
124         statement.setString(4, "system");
125         statement.setTimestamp(5, getCurrentTimeStamp());
126         statement.addBatch();
127
128         statement.setInt(1, 3);
129         statement.setString(2, "Brown");
130         statement.setString(3, "Charlie");
131         statement.setString(4, "system");
132         statement.setTimestamp(5, getCurrentTimeStamp());
133         statement.addBatch();
134
135         statement.executeBatch();
136         connection.commit();
137         System.out.println("Records are inserted!\n");
138         statement.close();
139     } catch (SQLException e) {
140         e.printStackTrace();
141     }
142 }
143
144 @Override
145 public void listRecords() {
146     try {
147         connection = dataSource.getConnection();

```

```

148     String sql = "SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY,
149             ↪ CREATED_DATE FROM EMPLOYEE WHERE FIRSTNAME LIKE ?";
150     System.out.println(sql);
151     statement = connection.prepareStatement(sql);
152     statement.setString(1, "J%");
153     ResultSet rs = statement.executeQuery();
154     while (rs.next()) {
155         int empid = rs.getInt("EMPID");
156         String lastName = rs.getString("LASTNAME");
157         String firstName = rs.getString("FIRSTNAME");
158         String createdBy = rs.getString("CREATED_BY");
159         Timestamp createdDate = rs.getTimestamp("CREATED_DATE");
160         System.out.println("EMPID: " + empid);
161         System.out.println("LASTNAME: " + lastName);
162         System.out.println("FIRSTNAME: " + firstName);
163         System.out.println("CREATED_BY: " + createdBy);
164         System.out.println("CREATED_DATE: " + createdDate + "\n");
165     }
166     statement.close();
167     connection.close();
168 } catch (SQLException e) {
169     e.printStackTrace();
170 }
171
172 private Timestamp getCurrentTimeStamp() {
173     java.util.Date today = new java.util.Date();
174     return new Timestamp(today.getTime());
175 }
176 }
```

14.3.7 Configure spring bean

- Create resources directory and add it to Build Path
src/main/resources
- Configuration metadata
src/main/resources/app-hello-dao-embedded.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       ↪ xmlns:context="http://www.springframework.org/schema/context"
5   xmlns:jdbc="http://www.springframework.org/schema/jdbc"
6   xsi:schemaLocation="http://www.springframework.org/schema/beans
7       http://www.springframework.org/schema/beans/spring-beans.xsd
8       http://www.springframework.org/schema/context
9       http://www.springframework.org/schema/context/spring-context.xsd
10      http://www.springframework.org/schema/jdbc
11      http://www.springframework.org/schema/jdbc/spring-jdbc.xsd">
12
13 <context:annotation-config />
14
15 <context:component-scan base-package="com.xyz" />
16
17 <bean id="employeeDAO" class="com.xyz.dao.EmployeeDAOImpl" />
18
19 <jdbc:embedded-database type="HSQL" id="dataSource">
20     <jdbc:script location="classpath:sql/create-db.sql" />
21     <jdbc:script location="classpath:sql/insert-data.sql" />
22 </jdbc:embedded-database>
23
24 <!--
25 <jdbc:embedded-database type="H2" id="dataSource">
26     <jdbc:script location="classpath:sql/create-db.sql" />
27     <jdbc:script location="classpath:sql/insert-data.sql" />
28 </jdbc:embedded-database>
29 <jdbc:embedded-database type="DERBY" id="dataSource">
```

```

30      <jdbc:script location="classpath:sql/create-db.sql" />
31      <jdbc:script location="classpath:sql/insert-data.sql" />
32  </jdbc:embedded-database>
33  -->
34
35  </beans>
```

14.3.8 SQL

- Create sql directory under resources directory
src/main/resources/sql
- Create Table
src/main/resources/sql/create-table.sql

```

1 CREATE TABLE EMPLOYEE (
2     EMPID INTEGER NOT NULL,
3     LASTNAME VARCHAR(20) NOT NULL,
4     FIRSTNAME VARCHAR(20) NOT NULL,
5     CREATED_BY VARCHAR(20) NOT NULL,
6     CREATED_DATE DATE NOT NULL,
7     PRIMARY KEY (EMPID)
8 );
```

src/main/resources/sql/insert-data.sql

```

1 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
2   ↪ VALUES (1001, 'Smith', 'John', 'system', CURRENT_DATE);
3 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
4   ↪ VALUES (1002, 'Doe', 'Jane', 'system', CURRENT_DATE);
5 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
6   ↪ VALUES (1003, 'Brown', 'Charlie', 'system', CURRENT_DATE);
```

14.3.9 Instantiating a container

- App
Type in the code list below to src/main/java ->com.xyz ->App.java

```

1 package com.xyz;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5
6 import com.xyz.dao.EmployeeDAO;
7
8 public class App {
9     public static void main(String[] args) {
10         ApplicationContext context = new
11             ClassPathXmlApplicationContext("app-hello-dao-embedded.xml");
12         EmployeeDAO dao = (EmployeeDAO) context.getBean("employeeDAO");
13         dao.insertRecord();
14         dao.listRecord();
15         dao.updateRecord();
16         dao.listRecord();
17         dao.deleteRecord();
18         dao.listRecord();
19         dao.batchInsertRecords();
20         dao.listRecords();
21         ((ClassPathXmlApplicationContext) context).close();
22     }
23 }
```

14.3.10 Run App - HSQL

Right click on project src/main/java ->com.xyz->App

Run As ->Java Application

You will see in the console:

```

1  INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
   ↪  VALUES (?, ?, ?, ?, ?)
2  Record is inserted!
3
4  SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
   ↪  EMPID = ?
5  EMPID: 1
6  LASTNAME: Smith
7  FIRSTNAME: John
8  CREATED_BY: system
9  CREATED_DATE: 2016-01-05 00:00:00.0
10
11 UPDATE EMPLOYEE SET LASTNAME = ? WHERE EMPID = ?
12 Record is updated!
13
14 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
   ↪  EMPID = ?
15 EMPID: 1
16 LASTNAME: Doe
17 FIRSTNAME: John
18 CREATED_BY: system
19 CREATED_DATE: 2016-01-05 00:00:00.0
20
21 DELETE FROM EMPLOYEE WHERE EMPID = ?
22 Record is deleted!
23
24 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
   ↪  EMPID = ?
25 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
   ↪  VALUES (?, ?, ?, ?, ?)
26 Records are inserted!
27
28 SELECT EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE FROM EMPLOYEE WHERE
   ↪  FIRSTNAME LIKE ?
29 EMPID: 1
30 LASTNAME: Smith
31 FIRSTNAME: John
32 CREATED_BY: system
33 CREATED_DATE: 2016-01-05 00:00:00.0
34
35 EMPID: 2
36 LASTNAME: Doe
37 FIRSTNAME: Jane
38 CREATED_BY: system
39 CREATED_DATE: 2016-01-05 00:00:00.0
40
41 EMPID: 1001
42 LASTNAME: Smith
43 FIRSTNAME: John
44 CREATED_BY: system
45 CREATED_DATE: 2016-01-05 00:00:00.0
46
47 EMPID: 1002
48 LASTNAME: Doe
49 FIRSTNAME: Jane
50 CREATED_BY: system
51 CREATED_DATE: 2016-01-05 00:00:00.0

```

14.3.11 Run App - H2 and Derby

Modify app-hello-dao-embedded.xml, use type H2 or DERBY, you can run it in H2 or DERBY.

CHAPTER 15

SPRING ORM

15.1 MyBatis

15.1.1 Create project using command line

```
1 c:\workspaces\workspace>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=app-hello-orm-mybatis
  ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

15.1.2 Modify pom.xml

- C:\workspaces\workspace\my-projects\app-hello-orm-mybatis\pom.xml
Use properties
Add hsql, mybatis-spring, spring dependencies
Modify junit version
Add maven-compiler-plugin, set JDK version to 1.7

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
```

[†]change c:\workspaces\workspace to /users/xyz/documents/workspace, then all the projects in this chapter will work in mac osx.

```

4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-orm-mybatis</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-orm-mybatis</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <hsqldb.version>2.3.3</hsqldb.version>
12    <java.version>1.7</java.version>
13    <javax.inject.version>1</javax.inject.version>
14    <junit.version>4.12</junit.version>
15    <maven.compiler.plugin.version>3.3.</maven.compiler.plugin.version>
16    <mybatis.version>3.3.0</mybatis.version>
17    <mybatis.spring.version>1.2.3</mybatis.spring.version>
18    <spring.version>4.2.0.RELEASE</spring.version>
19  </properties>
20  <dependencies>
21    <dependency>
22      <groupId>org.hsqldb</groupId>
23      <artifactId>hsqldb</artifactId>
24      <version>${hsqldb.version}</version>
25    </dependency>
26    <dependency>
27      <groupId>javax.inject</groupId>
28      <artifactId>javax.inject</artifactId>
29      <version>${javax.inject.version}</version>
30    </dependency>
31    <dependency>
32      <groupId>junit</groupId>
33      <artifactId>junit</artifactId>
34      <version>${junit.version}</version>
35      <scope>test</scope>
36    </dependency>
37    <dependency>
38      <groupId>org.mybatis</groupId>
39      <artifactId>mybatis</artifactId>
40      <version>${mybatis.version}</version>
41    </dependency>
42    <dependency>
43      <groupId>org.mybatis</groupId>
44      <artifactId>mybatis-spring</artifactId>
45      <version>${mybatis.spring.version}</version>
46    </dependency>
47    <dependency>
48      <groupId>org.springframework</groupId>
49      <artifactId>spring-context</artifactId>
50      <version>${spring.version}</version>
51    </dependency>
52    <dependency>
53      <groupId>org.springframework</groupId>
54      <artifactId>spring-core</artifactId>
55      <version>${spring.version}</version>
56    </dependency>
57    <dependency>
58      <groupId>org.springframework</groupId>
59      <artifactId>spring-jdbc</artifactId>
60      <version>${spring.version}</version>
61    </dependency>
62  </dependencies>
63  <build>
64    <finalName>app-hello-orm-mybatis</finalName>
65    <plugins>
66      <plugin>
67        <groupId>org.apache.maven.plugins</groupId>
68        <artifactId>maven-compiler-plugin</artifactId>
69        <version>${maven.compiler.plugin.version}</version>
70        <configuration>
71          <source>${java.version}</source>
72          <target>${java.version}</target>
73        </configuration>
74      </plugin>
75    </plugins>

```

```
76    </build>
77 </project>
```

15.1.3 Make this project an eclipse project

```
1 c:\workspaces\eclipse\my-projects\app-hello-orm-mybatis>mvn eclipse:clean
→ eclipse:eclipse
```

15.1.4 Import this project to eclipse

- Launch eclipse
File → Import → General → Existing Projects into Workspace
Browse to app-hello-orm-mybatis, and import it.

15.1.5 Add a java model class

- src/main/java → com.xyz.model → Employee.java

```
1 package com.xyz.model;
2
3 import java.sql.Timestamp;
4
5 public class Employee {
6     int empId;
7     String lastName;
8     String firstName;
9     String createdBy;
10    Timestamp createdDate;
11
12    public int getEmpId() {
13        return empId;
14    }
15    public void setEmpId(int empId) {
16        this.empId = empId;
17    }
18    public String getLastname() {
19        return lastName;
20    }
21    public void setLastname(String lastName) {
22        this.lastName = lastName;
23    }
24    public String getFirstname() {
25        return firstName;
26    }
27    public void setFirstname(String firstName) {
28        this.firstName = firstName;
29    }
30    public String getCreatedBy() {
31        return createdBy;
32    }
33    public void setCreatedBy(String createdBy) {
34        this.createdBy = createdBy;
35    }
36    public Timestamp getCreatedDate() {
37        return createdDate;
38    }
39    public void setCreatedDate(Timestamp createdDate) {
40        this.createdDate = createdDate;
41    }
42
43    @Override
44    public String toString() {
```

```

45         return "Employee [empId: " + empId + " lastName = " + lastName + "
46             ↵ firstName: " + firstName + " createdBy: " + createdBy + "
47             ↵ createdDate = " + createdDate + "]";
}

```

15.1.6 Add java dao classes

- src/main/java -> com.xyz.dao -> EmployeeDAO.java

```

1 package com.xyz.dao;
2
3 public interface EmployeeDAO {
4     public void getRecordNumbers();
5     public void listRecord();
6     public void listRecords();
7 }

```

src/main/java -> com.xyz.dao -> EmployeeDAOImpl.java

```

1 package com.xyz.dao;
2
3 import java.util.List;
4
5 import javax.inject.Inject;
6 import javax.inject.Named;
7
8 import org.mybatis.spring.SqlSessionTemplate;
9
10 import com.xyz.model.Employee;
11
12 @Named
13 public class EmployeeDAOImpl implements EmployeeDAO {
14     @Inject
15     private SqlSessionTemplate sqlSessionTemplate;
16
17     @Override
18     public void getRecordNumbers() {
19         int totalRecords = (Integer)
20             sqlSessionTemplate.selectOne("findEmployeeCount");
21         System.out.println("Total Employee Numbers: " + totalRecords);
22     }
23
24     @Override
25     public void listRecord() {
26         Employee employee = (Employee)
27             sqlSessionTemplate.selectOne("findEmployee", 1001);
28         System.out.println("findEmployee:");
29         System.out.println(employee.toString());
30     }
31
32     @Override
33     public void listRecords() {
34         List<Employee> employees =
35             sqlSessionTemplate.selectList("findEmployees", "J%");
36         System.out.println("findEmployees:");
37         for(Employee employee : employees) {
38             System.out.println(employee.toString());
39         }
40     }
41 }

```

15.1.7 Configure spring bean

- Create resources directory and add it to Build Path
src/main/resources

- Configuration metadata

src/main/resources/app-hello-orm-mybatis.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4          <!-- xmlns:context="http://www.springframework.org/schema/context"
5      xmlns:jdbc="http://www.springframework.org/schema/jdbc"
6      xsi:schemaLocation="http://www.springframework.org/schema/beans
7          http://www.springframework.org/schema/beans/spring-beans.xsd
8          http://www.springframework.org/schema/context
9          http://www.springframework.org/schema/context/spring-context.xsd
10         http://www.springframework.org/schema/jdbc
11         http://www.springframework.org/schema/jdbc/spring-jdbc.xsd">
12
13     <context:annotation-config />
14
15     <context:component-scan base-package="com.xyz" />
16
17     <bean id="employeeDAO" class="com.xyz.dao.EmployeeDAOImpl" />
18
19     <jdbc:embedded-database type="HSQL" id="dataSource">
20         <jdbc:script location="classpath:sql/create-db.sql" />
21         <jdbc:script location="classpath:sql/insert-data.sql" />
22     </jdbc:embedded-database>
23
24     <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
25         <property name="dataSource" ref="dataSource" />
26         <property name="configLocation" value="classpath:mybatis/mybatis-config.xml"
27             <!-- /-->
28     </bean>
29
30     <bean id="sqlSessionTemplate" class="org.mybatis.spring.SqlSessionTemplate">
31         <constructor-arg index="0" ref="sqlSessionFactory" />
32     </bean>
33
34 </beans>
```

src/main/resources/mybatis/mybatis-config.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE configuration
3      PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
4      "http://mybatis.org/dtd/mybatis-3-config.dtd">
5
6  <configuration>
7      <mappers>
8          <mapper resource="mybatis/mapper-hello.xml" />
9      </mappers>
10 </configuration>
```

src/main/resources/mybatis/mybatis-hello.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
3      <!-- "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
4  <mapper namespace="hello">
5
6      <select id="findEmployee" parameterType="java.lang.Integer"
7          <!-- resultType="com.xyz.model.Employee" statementType="PREPARED">
8          SELECT EMPID AS empId, LASTNAME AS lastName, FIRSTNAME AS firstName,
9              <!-- CREATED_BY AS createdBy, CREATED_DATE AS createdDate
10             FROM EMPLOYEE
11             WHERE EMPID = #{value}
12         </select>
13
14      <select id="findEmployees" parameterType="java.lang.String"
15          <!-- resultType="com.xyz.model.Employee" statementType="PREPARED">
16          SELECT EMPID AS empId, LASTNAME AS lastName, FIRSTNAME AS firstName,
17              <!-- CREATED_BY AS createdBy, CREATED_DATE AS createdDate
```

```

13      FROM EMPLOYEE
14      WHERE FIRSTNAME LIKE #{value}
15  </select>
16
17  <select id="findEmployeeCount" resultType="java.lang.Integer"
18      <!-- statementType="PREPARED">
19      SELECT COUNT(EMPID)
20      FROM EMPLOYEE
21  </select>
22
22 </mapper>

```

15.1.8 SQL

- Create sql directory under resources directory
src/main/resources/sql
- Create Table
src/main/resources/sql/create-table.sql

```

1 CREATE TABLE EMPLOYEE (
2     EMPID INTEGER NOT NULL,
3     LASTNAME VARCHAR(20) NOT NULL,
4     FIRSTNAME VARCHAR(20) NOT NULL,
5     CREATED_BY VARCHAR(20) NOT NULL,
6     CREATED_DATE DATE NOT NULL,
7     PRIMARY KEY (EMPID)
8 );

```

src/main/resources/sql/insert-data.sql

```

1 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
2     <-- VALUES (1001, 'Smith', 'John', 'system', CURRENT_DATE);
3 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
4     <-- VALUES (1002, 'Doe', 'Jane', 'system', CURRENT_DATE);
5 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
6     <-- VALUES (1003, 'Brown', 'Charlie', 'system', CURRENT_DATE);

```

15.1.9 Instantiating a container

- App
Type in the code list below to src/main/java -> com.xyz -> App.java

```

1 package com.xyz;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5
6 import com.xyz.dao.EmployeeDAO;
7
8 public class App {
9     public static void main(String[] args) {
10         ApplicationContext context = new
11             ClassPathXmlApplicationContext("app-hello-orm-mybatis.xml");
12         EmployeeDAO dao = (EmployeeDAO) context.getBean("employeeDAO");
13         dao.getRecordNumbers();
14         dao.listRecord();
15         dao.listRecords();
16     }
16 }

```

15.1.10 Run App

Right click on project src/main/java ->com.xyz ->App

Run As ->Java Application

You will see in the console:

```

1 Total Employee Numbers: 3
2 findEmployee:
3 Employee [empid: 1001 lastName = Smith firstName: John createdBy: system
   ↵   createdDate = 2016-01-11 00:00:00.0]
4 findEmployees:
5 Employee [empid: 1001 lastName = Smith firstName: John createdBy: system
   ↵   createdDate = 2016-01-11 00:00:00.0]
6 Employee [empid: 1002 lastName = Doe firstName: Jane createdBy: system
   ↵   createdDate = 2016-01-11 00:00:00.0]
```

15.2 Hibernate

15.2.1 Create project using command line

```

1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
   ↵   -DartifactId=app-hello-orm-hibernate
   ↵   -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

15.2.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\app-hello-orm-hibernate\pom.xml

Use properties

Add hsql, hibernate, spring dependencies

Modify junit version

Add maven-compiler-plugin, set JDK version to 1.7

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
   ↵   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
   ↵   http://maven.apache.org/maven-v4_0_.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-orm-hibernate</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-orm-hibernate</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <hsqldb.version>2.3.3</hsqldb.version>
12    <java.version>1.7</java.version>
13    <javax.inject.version>1</javax.inject.version>
14    <junit.version>4.12</junit.version>
15    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
16    <hibernate.version>5.0.6.Final</hibernate.version>
17    <spring.version>4.2.0.RELEASE</spring.version>
18  </properties>
19  <dependencies>
20    <dependency>
21      <groupId>org.hsqldb</groupId>
22      <artifactId>hsqldb</artifactId>
23      <version>${hsqldb.version}</version>
24    </dependency>
25    <dependency>
26      <groupId>javax.inject</groupId>
27      <artifactId>javax.inject</artifactId>
```

```

28      <version>${jaxb.inject.version}</version>
29    </dependency>
30    <dependency>
31      <groupId>junit</groupId>
32      <artifactId>junit</artifactId>
33      <version>${junit.version}</version>
34      <scope>test</scope>
35    </dependency>
36    <dependency>
37      <groupId>org.hibernate</groupId>
38      <artifactId>hibernate-core</artifactId>
39      <version>${hibernate.version}</version>
40    </dependency>
41    <dependency>
42      <groupId>org.springframework</groupId>
43      <artifactId>spring-context</artifactId>
44      <version>${spring.version}</version>
45    </dependency>
46    <dependency>
47      <groupId>org.springframework</groupId>
48      <artifactId>spring-core</artifactId>
49      <version>${spring.version}</version>
50    </dependency>
51    <dependency>
52      <groupId>org.springframework</groupId>
53      <artifactId>spring-jdbc</artifactId>
54      <version>${spring.version}</version>
55    </dependency>
56    <dependency>
57      <groupId>org.springframework</groupId>
58      <artifactId>spring-orm</artifactId>
59      <version>${spring.version}</version>
60    </dependency>
61    <dependency>
62      <groupId>org.springframework</groupId>
63      <artifactId>spring-tx</artifactId>
64      <version>${spring.version}</version>
65    </dependency>
66  </dependencies>
67  <build>
68    <finalName>app-hello-orm-hibernate</finalName>
69    <plugins>
70      <plugin>
71        <groupId>org.apache.maven.plugins</groupId>
72        <artifactId>maven-compiler-plugin</artifactId>
73        <version>${maven.compiler.plugin.version}</version>
74        <configuration>
75          <source>${java.version}</source>
76          <target>${java.version}</target>
77        </configuration>
78      </plugin>
79    </plugins>
80  </build>
81 </project>

```

15.2.3 Make this project an eclipse project

```

1 c:\workspaces\eclipse\my-projects\app-hello-orm-hibernate>mvn eclipse:clean
  ↵   eclipse:eclipse

```

15.2.4 Import this project to eclipse

- Launch eclipse
- File ->Import ->General ->Existing Projects into Workspace
Browse to app-hello-orm-hibernate, and import it.

15.2.5 Add a java model class

- src/main/java -> com.xyz.model -> Employee.java

```

1 package com.xyz.model;
2
3 import java.sql.Timestamp;
4
5 public class Employee {
6     int empId;
7     String lastName;
8     String firstName;
9     String createdBy;
10    Timestamp createdDate;
11
12    public int getEmpId() {
13        return empId;
14    }
15    public void setEmpId(int empId) {
16        this.empId = empId;
17    }
18    public String getLastname() {
19        return lastName;
20    }
21    public void setLastname(String lastName) {
22        this.lastName = lastName;
23    }
24    public String getFirstName() {
25        return firstName;
26    }
27    public void setFirstName(String firstName) {
28        this.firstName = firstName;
29    }
30    public String getCreatedBy() {
31        return createdBy;
32    }
33    public void setCreatedBy(String createdBy) {
34        this.createdBy = createdBy;
35    }
36    public Timestamp getCreatedDate() {
37        return createdDate;
38    }
39    public void setCreatedDate(Timestamp createdDate) {
40        this.createdDate = createdDate;
41    }
42
43    @Override
44    public String toString() {
45        return "Employee [empid: " + empId + " lastName = " + lastName + "
46        ↪   firstName: " + firstName + " createdBy: " + createdBy + "
47        ↪   createdDate = " + createdDate + "]";
48    }
49 }
```

15.2.6 Add java dao classes

- src/main/java -> com.xyz.dao -> EmployeeDAO.java

```

1 package com.xyz.dao;
2
3 import java.util.List;
4
5 import com.xyz.model.Employee;
6
7 public interface EmployeeDAO {
8     public void insertRecord(Employee employee);
9     public void updateRecord(Employee employee);
10    public void deleteRecord(int empId);
11    public Employee listRecord(int empId);
```

```

12     public List<Employee> listRecords() ;
13 }
```

src/main/java -> com.xyz.dao -> EmployeeDAOImpl.java

```

1 package com.xyz.dao;
2
3 import java.util.List;
4
5 import javax.inject.Inject;
6 import javax.inject.Named;
7
8 import org.hibernate.Session;
9 import org.hibernate.SessionFactory;
10 import org.springframework.transaction.annotation.Transactional;
11
12 import com.xyz.model.Employee;
13
14 @Named
15 public class EmployeeDAOImpl implements EmployeeDAO {
16     @Inject
17     private SessionFactory sessionFactory;
18
19     @Override
20     @Transactional
21     public void insertRecord(Employee employee) {
22         Session session = sessionFactory.getCurrentSession();
23         session.persist(employee);
24         System.out.println("\nInsert Record:");
25         System.out.println(employee.toString());
26     }
27
28     @Override
29     @Transactional
30     public void updateRecord(Employee employee) {
31         Session session = sessionFactory.getCurrentSession();
32         session.update(employee);
33         System.out.println("\nUpdate Record:");
34         System.out.println(employee.toString());
35     }
36
37     @Override
38     @Transactional
39     public void deleteRecord(int empId) {
40         Session session = sessionFactory.getCurrentSession();
41         Employee employee = (Employee) session.load(Employee.class, new
42             ↪ Integer(empId));
43         if (null != employee) {
44             session.delete(employee);
45             System.out.println("\nDelete Record:");
46             System.out.println(employee.toString());
47         }
48
49     @Override
50     @Transactional
51     public Employee listRecord(int empId) {
52         Session session = sessionFactory.getCurrentSession();
53         Employee employee = (Employee) session.load(Employee.class, new
54             ↪ Integer(empId));
55         System.out.println("\nList Record:");
56         System.out.println(employee.toString());
57         return employee;
58     }
59
60     @SuppressWarnings("unchecked")
61     @Override
62     @Transactional
63     public List<Employee> listRecords() {
64         Session session = sessionFactory.getCurrentSession();
65         List<Employee> employees = (List<Employee>) session.createQuery("FROM
66             ↪ Employee").list();
```

```

65     System.out.println("\nList Records:");
66     for (Employee employee : employees) {
67         System.out.println(employee.toString());
68     }
69     return employees;
70 }
71 }
```

15.2.7 Configure spring bean

- Create resources directory and add it to Build Path
src/main/resources
- Configuration metadata
src/main/resources/app-hello-orm-hibernate.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4          xsi:schemaLocation="http://www.springframework.org/schema/beans
5              http://www.springframework.org/schema/beans/spring-beans.xsd
6              http://www.springframework.org/schema/context
7                  http://www.springframework.org/schema/context/spring-context.xsd
8                  http://www.springframework.org/schema/jdbc
9                      http://www.springframework.org/schema/jdbc/spring-jdbc.xsd
10                     http://www.springframework.org/schema/tx
11                         http://www.springframework.org/schema/tx/spring-tx.xsd">
12
13     <context:annotation-config />
14
15     <context:component-scan base-package="com.xyz" />
16
17     <bean id="employeeDAO" class="com.xyz.dao.EmployeeDAOImpl" />
18
19     <jdbc:embedded-database type="HSQL" id="dataSource">
20         <jdbc:script location="classpath:sql/create-db.sql" />
21         <jdbc:script location="classpath:sql/insert-data.sql" />
22     </jdbc:embedded-database>
23
24     <bean id="sessionFactory"
25         class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">
26         <property name="dataSource" ref="dataSource" />
27         <property name="configLocation" value="classpath:hibernate.cfg.xml" />
28     </bean>
29
30     <tx:annotation-driven />
31     <bean id="transactionManager"
32         class="org.springframework.orm.hibernate5.HibernateTransactionManager">
33         <property name="sessionFactory" ref="sessionFactory" />
34     </bean>
35 </beans>
```

src/main/resources/hibernate.cfg.xml

```

1  <?xml version='1.0' encoding='utf-8'?>
2  <!DOCTYPE hibernate-configuration PUBLIC
3      "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
4      "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
5  <hibernate-configuration>
6      <session-factory>
7          <property name="dialect">org.hibernate.dialect.HSQLDialect</property>
8          <property name="show_sql">true</property>
9          <mapping resource="hibernate/Employee.hbm.xml" />
10     </session-factory>
11 </hibernate-configuration>
```

src/main/resources/hibernate/Employee.hbm.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-mapping PUBLIC
3   "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
4   "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
5 <hibernate-mapping package="com.xyz.model">
6   <class name="Employee" table="EMPLOYEE">
7     <id name="empId" column="EMPID" />
8     <property name="lastName" column="LASTNAME" />
9     <property name="firstName" column="FIRSTNAME" />
10    <property name="createdBy" column="CREATED_BY" />
11    <property name="createdDate" column="CREATED_DATE" />
12  </class>
13 </hibernate-mapping>
```

15.2.8 SQL

- Create sql directory under resources directory
src/main/resources/sql
- Create Table
src/main/resources/sql/create-table.sql

```

1 CREATE TABLE EMPLOYEE (
2   EMPID INTEGER NOT NULL,
3   LASTNAME VARCHAR(20) NOT NULL,
4   FIRSTNAME VARCHAR(20) NOT NULL,
5   CREATED_BY VARCHAR(20) NOT NULL,
6   CREATED_DATE DATE NOT NULL,
7   PRIMARY KEY (EMPID)
8 );
```

src/main/resources/sql/insert-data.sql

```

1 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
2   ↪ VALUES (1001, 'Smith', 'John', 'system', CURRENT_DATE);
3 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
4   ↪ VALUES (1002, 'Doe', 'Jane', 'system', CURRENT_DATE);
5 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
6   ↪ VALUES (1003, 'Brown', 'Charlie', 'system', CURRENT_DATE);
```

15.2.9 Instantiating a container

- App
- Type in the code list below to src/main/java -> com.xyz -> App.java

```

1 package com.xyz;
2
3 import java.sql.Timestamp;
4 import java.util.Date;
5
6 import org.springframework.context.ApplicationContext;
7 import org.springframework.context.support.ClassPathXmlApplicationContext;
8
9 import com.xyz.dao.EmployeeDAO;
10 import com.xyz.model.Employee;
11
12 public class App {
13   public static void main(String[] args) {
14     ApplicationContext context = new
15       ClassPathXmlApplicationContext("app-hello-orm-hibernate.xml");
```

```

15     EmployeeDAO dao = (EmployeeDAO) context.getBean("employeeDAO");
16
17     dao.listRecords();
18
19     Employee employee = new Employee();
20     employee.setEmpId(1);
21     employee.setLastName("Smith");
22     employee.setFirstName("John");
23     employee.setCreatedBy("system");
24     employee.setCreatedDate(new Timestamp((new Date()).getTime()));
25     dao.insertRecord(employee);
26     dao.listRecord(1);
27
28     employee.setLastName("Doe");
29     dao.updateRecord(employee);
30     dao.listRecord(1);
31
32     dao.listRecords();
33     dao.deleteRecord(1);
34     dao.listRecords();
35
36     ((ClassPathXmlApplicationContext) context).close();
37 }
38 }
```

15.2.10 Run App

Right click on project src/main/java ->com.xyz->App

Run As ->Java Application

You will see in the console:

```

1 Hibernate: select employee0_.EMPID as EMPID1_0_, employee0_.LASTNAME as
   ↪ LASTNAME2_0_0_, employee0_.FIRSTNAME as FIRSTNAM3_0_0_, employee0_.CREATED_BY
   ↪ as CREATED_4_0_0_, employee0_.CREATED_DATE as CREATED_5_0_0_ from EMPLOYEE
   ↪ employee0_
2
3 List Records:
4 Employee [empid: 1001 lastName = Smith firstName: John createdBy: system
   ↪ createdDate = 2016-01-12 00:00:00.0]
5 Employee [empid: 1002 lastName = Doe firstName: Jane createdBy: system
   ↪ createdDate = 2016-01-12 00:00:00.0]
6 Employee [empid: 1003 lastName = Brown firstName: Charlie createdBy: system
   ↪ createdDate = 2016-01-12 00:00:00.0]
7
8 Insert Record:
9 Employee [empid: 1 lastName = Smith firstName: John createdBy: system
   ↪ createdDate = 2016-01-12 16:23:43.333]
10 Hibernate: insert into EMPLOYEE (LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE,
   ↪ EMPID) values (?, ?, ?, ?, ?)
11
12 List Record:
13 Hibernate: select employee0_.EMPID as EMPID1_0_0_, employee0_.LASTNAME as
   ↪ LASTNAME2_0_0_0_, employee0_.FIRSTNAME as FIRSTNAM3_0_0_0_,
   ↪ employee0_.CREATED_BY as CREATED_4_0_0_0_, employee0_.CREATED_DATE as
   ↪ CREATED_5_0_0_0_ from EMPLOYEE employee0_ where employee0_.EMPID=?
14 Employee [empid: 1 lastName = Smith firstName: John createdBy: system
   ↪ createdDate = 2016-01-12 00:00:00.0]
15
16 Update Record:
17 Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
   ↪ = 2016-01-12 16:23:43.333]
18 Hibernate: update EMPLOYEE set LASTNAME=?, FIRSTNAME=?, CREATED_BY=?,
   ↪ CREATED_DATE=? where EMPID=?
19
20 List Record:
21 Hibernate: select employee0_.EMPID as EMPID1_0_0_0_, employee0_.LASTNAME as
   ↪ LASTNAME2_0_0_0_0_, employee0_.FIRSTNAME as FIRSTNAM3_0_0_0_0_,
   ↪ employee0_.CREATED_BY as CREATED_4_0_0_0_0_, employee0_.CREATED_DATE as
   ↪ CREATED_5_0_0_0_0_ from EMPLOYEE employee0_ where employee0_.EMPID=?
```

```

22 Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
   ↳ = 2016-01-12 00:00:00.0]
23 Hibernate: select employee0_.EMPID as EMPID1_0_, employee0_.LASTNAME as
   ↳ LASTNAME2_0_, employee0_.FIRSTNAME as FIRSTNAM3_0_, employee0_.CREATED_BY
   ↳ as CREATED_4_0_, employee0_.CREATED_DATE as CREATED_5_0_ from EMPLOYEE
   ↳ employee0_
24
25 List Records:
26 Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
   ↳ = 2016-01-12 00:00:00.0]
27 Employee [empid: 1001 lastName = Smith firstName: John createdBy: system
   ↳ createdDate = 2016-01-12 00:00:00.0]
28 Employee [empid: 1002 lastName = Doe firstName: Jane createdBy: system
   ↳ createdDate = 2016-01-12 00:00:00.0]
29 Employee [empid: 1003 lastName = Brown firstName: Charlie createdBy: system
   ↳ createdDate = 2016-01-12 00:00:00.0]
30 Hibernate: select employee0_.EMPID as EMPID1_0_, employee0_.LASTNAME as
   ↳ LASTNAME2_0_0_, employee0_.FIRSTNAME as FIRSTNAM3_0_0_,
   ↳ employee0_.CREATED_BY as CREATED_4_0_0_, employee0_.CREATED_DATE as
   ↳ CREATED_5_0_0_ from EMPLOYEE employee0_ where employee0_.EMPID=??
31
32 Delete Record:
33 Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
   ↳ = 2016-01-12 00:00:00.0]
34 Hibernate: delete from EMPLOYEE where EMPID=?
35 Hibernate: select employee0_.EMPID as EMPID1_0_, employee0_.LASTNAME as
   ↳ LASTNAME2_0_, employee0_.FIRSTNAME as FIRSTNAM3_0_, employee0_.CREATED_BY
   ↳ as CREATED_4_0_, employee0_.CREATED_DATE as CREATED_5_0_ from EMPLOYEE
   ↳ employee0_
36
37 List Records:
38 Employee [empid: 1001 lastName = Smith firstName: John createdBy: system
   ↳ createdDate = 2016-01-12 00:00:00.0]
39 Employee [empid: 1002 lastName = Doe firstName: Jane createdBy: system
   ↳ createdDate = 2016-01-12 00:00:00.0]
40 Employee [empid: 1003 lastName = Brown firstName: Charlie createdBy: system
   ↳ createdDate = 2016-01-12 00:00:00.0]

```

15.3 JPA

15.3.1 JPA

15.3.1.1 Create project using command line

```

1 c:\workspaces\workspace\my-projects>mvn archetype:generate -DgroupId=com.xyz
   ↳ -DartifactId=app-hello-orm-jpa
   ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

```

15.3.1.2 Modify pom.xml

- C:\workspaces\workspace\my-projects\app-hello-orm-jpa\pom.xml
- Use properties
- Add oracle, hibernate, jta dependencies
- Modify junit version
- Add maven-compiler-plugin, set JDK version to 1.7

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
   ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
   ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-orm-jpa</artifactId>

```

```

6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-orm-jpa</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <java.version>1.7</java.version>
12    <junit.version>4.12</junit.version>
13    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
14    <hibernate.version>5.0.6.Final</hibernate.version>
15    <oracle.version>12.1.0.2</oracle.version>
16    <jta.version>1.1</jta.version>
17  </properties>
18  <dependencies>
19    <dependency>
20      <groupId>javax.transaction</groupId>
21      <artifactId>jta</artifactId>
22      <version>${jta.version}</version>
23    </dependency>
24    <dependency>
25      <groupId>junit</groupId>
26      <artifactId>junit</artifactId>
27      <version>${junit.version}</version>
28      <scope>test</scope>
29    </dependency>
30    <dependency>
31      <groupId>org.hibernate</groupId>
32      <artifactId>hibernate-c3p0</artifactId>
33      <version>${hibernate.version}</version>
34    </dependency>
35    <dependency>
36      <groupId>org.hibernate</groupId>
37      <artifactId>hibernate-entitymanager</artifactId>
38      <version>${hibernate.version}</version>
39    </dependency>
40    <dependency>
41      <groupId>com.oracle</groupId>
42      <artifactId>ojdbc7</artifactId>
43      <version>${oracle.version}</version>
44    </dependency>
45  </dependencies>
46  <build>
47    <finalName>app-hello-orm-jpa</finalName>
48    <plugins>
49      <plugin>
50        <groupId>org.apache.maven.plugins</groupId>
51        <artifactId>maven-compiler-plugin</artifactId>
52        <version>${maven.compiler.plugin.version}</version>
53        <configuration>
54          <source>${java.version}</source>
55          <target>${java.version}</target>
56        </configuration>
57      </plugin>
58    </plugins>
59  </build>
60</project>
```

15.3.1.3 Make this project an eclipse project

```

1 c:\workspaces\workspace\my-projects\app-hello-orm-jpa>mvn eclipse:clean
  ↵ eclipse:eclipse

```

15.3.1.4 Import this project to eclipse

- Launch eclipse
- File → Import → General → Existing Projects into Workspace
Browse to app-hello-orm-jpa, and import it.

15.3.1.5 Add a java model class

- src/main/java -> com.xyz.model -> Employee.java

```

1 package com.xyz.model;
2
3 import java.sql.Timestamp;
4
5 import javax.persistence.Column;
6 import javax.persistence.Entity;
7 import javax.persistence.Id;
8 import javax.persistence.Table;
9
10 @Entity
11 @Table(name = "EMPLOYEE")
12 public class Employee {
13     @Id
14     @Column(name = "EMPIID")
15     int empId;
16     @Column(name = "LASTNAME")
17     String lastName;
18     @Column(name = "FIRSTNAME")
19     String firstName;
20     @Column(name = "CREATED_BY")
21     String createdBy;
22     @Column(name = "CREATED_DATE")
23     Timestamp createdDate;
24
25     public int getEmpId() {
26         return empId;
27     }
28     public void setEmpId(int empId) {
29         this.empId = empId;
30     }
31     public String getLastname() {
32         return lastName;
33     }
34     public void setLastname(String lastName) {
35         this.lastName = lastName;
36     }
37     public String getFirstname() {
38         return firstName;
39     }
40     public void setFirstname(String firstName) {
41         this.firstName = firstName;
42     }
43     public String getCreatedBy() {
44         return createdBy;
45     }
46     public void setCreatedBy(String createdBy) {
47         this.createdBy = createdBy;
48     }
49     public Timestamp getCreatedDate() {
50         return createdDate;
51     }
52     public void setCreatedDate(Timestamp createdDate) {
53         this.createdDate = createdDate;
54     }
55
56     @Override
57     public String toString() {
58         return "Employee [empid: " + empId + " lastName = " + lastName + "
59             ↵   firstName: " + firstName + " createdBy: " + createdBy + "
60             ↵   createdDate = " + createdDate + "]";
61     }
62 }
```

15.3.1.6 Add java dao classes

- src/main/java -> com.xyz.dao -> EmployeeDAO.java

```

1 package com.xyz.dao;
2
3 import java.util.List;
4
5 import com.xyz.model.Employee;
6
7 public interface EmployeeDAO {
8     public void insertRecord(Employee employee);
9     public void updateRecord(int empId, String lastName);
10    public void deleteRecord(int empId);
11    public Employee listRecord(int empId);
12    public List<Employee> listRecords();
13 }

```

src/main/java -> com.xyz.dao -> EmployeeDAOImpl.java

```

1 package com.xyz.dao;
2
3 import java.util.List;
4
5 import javax.persistence.EntityManager;
6
7 import com.xyz.model.Employee;
8
9 public class EmployeeDAOImpl implements EmployeeDAO {
10     private EntityManager entityManager;
11
12     public EmployeeDAOImpl(EntityManager entityManager) {
13         super();
14         this.entityManager = entityManager;
15     }
16
17     @Override
18     public void insertRecord(Employee employee) {
19         entityManager.getTransaction().begin();
20         entityManager.persist(employee);
21         entityManager.getTransaction().commit();
22         System.out.println("\nInsert Record:");
23         System.out.println(employee.toString());
24     }
25
26     @Override
27     public void updateRecord(int empId, String lastName) {
28         Employee employee = (Employee) entityManager.find(Employee.class, new
29             ↪ Integer(empId));
30         if (null != employee) {
31             entityManager.getTransaction().begin();
32             employee.setLastName(lastName);
33             entityManager.getTransaction().commit();
34             System.out.println("\nUpdate Record:");
35             System.out.println(employee.toString());
36         }
37     }
38
39     @Override
40     public void deleteRecord(int empId) {
41         Employee employee = (Employee) entityManager.find(Employee.class, new
42             ↪ Integer(empId));
43         if (null != employee) {
44             entityManager.getTransaction().begin();
45             entityManager.remove(employee);
46             entityManager.getTransaction().commit();
47             System.out.println("\nDelete Record:");
48             System.out.println(employee.toString());
49         }
50     }
51     @Override
52     public Employee listRecord(int empId) {
53         Employee employee = (Employee) entityManager.find(Employee.class, new
54             ↪ Integer(empId));

```

```

53         System.out.println("\nList Record:");
54         System.out.println(employee.toString());
55     }
56 }
57
58 @SuppressWarnings("unchecked")
59 @Override
60 public List<Employee> listRecords() {
61     List<Employee> employees = (List<Employee>)
62         entityManager.createQuery("FROM Employee").getResultList();
63     System.out.println("\nList Records:");
64     for (Employee employee : employees) {
65         System.out.println(employee.toString());
66     }
67     return employees;
68 }
```

src/main/java -> com.xyz.dao -> PersistenceManager.java

```

1 package com.xyz.dao;
2
3 import javax.persistence.EntityManager;
4 import javax.persistence.EntityManagerFactory;
5 import javax.persistence.Persistence;
6
7 public enum PersistenceManager {
8     INSTANCE;
9     private EntityManagerFactory emFactory;
10
11     private PersistenceManager() {
12         emFactory = Persistence.createEntityManagerFactory("orm-jpa");
13     }
14
15     public EntityManager getEntityManager() {
16         return emFactory.createEntityManager();
17     }
18
19     public void close() {
20         emFactory.close();
21     }
22 }
```

15.3.1.7 Configure JPA

- Create resources directory and add it to Build Path
src/main/resources
- Configuration persistence
src/main/resources/META-INF/persistence.xml

```

1 <persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
2     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
4         http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
5     version="2.1">
6
7     <persistence-unit name="orm-jpa" transaction-type="RESOURCE_LOCAL">
8         <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
9         <properties>
10             <property name="javax.persistence.jdbc.driver"
11                 value="oracle.jdbc.driver.OracleDriver" />
12             <property name="javax.persistence.jdbc.url"
13                 value="jdbc:oracle:thin:@localhost:1522:orcl" />
14             <property name="javax.persistence.jdbc.user" value="system" />
15             <property name="javax.persistence.jdbc.password" value="password" />
16
17             <property name="hibernate.show_sql" value="true" />
18             <property name="hibernate.format_sql" value="true" />
```

```

16   <property name="hibernate.dialect"
17     ↪   value="org.hibernate.dialect.Oracle10gDialect" />
18   <property name="hibernate.hbm2ddl.auto" value="create-drop" />
19
20   <property name="hibernate.c3p0.min_size" value="5" />
21   <property name="hibernate.c3p0.max_size" value="20" />
22   <property name="hibernate.c3p0.timeout" value="500" />
23   <property name="hibernate.c3p0.max_statements" value="50" />
24   <property name="hibernate.c3p0.idle_test_period" value="2000" />
25 </properties>
26 </persistence-unit>
27 </persistence>

```

15.3.1.8 Instantiating a container

- App

Type in the code list below to src/main/java -> com.xyz ->App.java

```

1 package com.xyz;
2
3 import java.sql.Timestamp;
4 import java.util.Date;
5
6 import javax.persistence.EntityManager;
7
8 import com.xyz.dao.EmployeeDAO;
9 import com.xyz.dao.EmployeeDAOImpl;
10 import com.xyz.dao.PersistenceManager;
11 import com.xyz.model.Employee;
12
13 public class App {
14     public static void main(String[] args) {
15         EntityManager entityManager =
16             ↪ PersistenceManager.INSTANCE.getEntityManager();
17
18         EmployeeDAO dao = new EmployeeDAOImpl(entityManager);
19
20         dao.listRecords();
21
22         Employee employee = new Employee();
23         employee.setEmpId(1);
24         employee.setLastName("Smith");
25         employee.setFirstName("John");
26         employee.setCreatedBy("system");
27         employee.setCreatedDate(new Timestamp((new Date()).getTime()));
28         dao.insertRecord(employee);
29         dao.listRecord(1);
30
31         dao.updateRecord(1, "Doe");
32         dao.listRecord(1);
33
34         Employee employee1001 = new Employee();
35         employee1001.setEmpId(1001);
36         employee1001.setLastName("Smith");
37         employee1001.setFirstName("John");
38         employee1001.setCreatedBy("system");
39         employee1001.setCreatedDate(new Timestamp((new Date()).getTime()));
40         dao.insertRecord(employee1001);
41
42         Employee employee1002 = new Employee();
43         employee1002.setEmpId(1002);
44         employee1002.setLastName("Doe");
45         employee1002.setFirstName("Jane");
46         employee1002.setCreatedBy("system");
47         employee1002.setCreatedDate(new Timestamp((new Date()).getTime()));
48         dao.insertRecord(employee1002);
49
50         Employee employee1003 = new Employee();
51         employee1003.setEmpId(1003);

```

```

51     employee1003.setLastName("Brown");
52     employee1003.setFirstName("Charlie");
53     employee1003.setCreatedBy("system");
54     employee1003.setCreatedDate(new Timestamp((new Date()).getTime()));
55     dao.insertRecord(employee1003);
56
57     dao.listRecords();
58
59     dao.deleteRecord(1);
60     dao.listRecords();
61
62     entityManager.close();
63     PersistenceManager.INSTANCE.close();
64 }
}

```

15.3.1.9 Run App

Right click on project src/main/java ->com.xyz->App

Run As ->Java Application

You will see in the console:

```

1 Hibernate:
2     drop table EMPLOYEE cascade constraints
3
4 Hibernate:
5     create table EMPLOYEE (
6         EMPID number(10,0) not null,
7         CREATED_BY varchar2(255 char),
8         CREATED_DATE timestamp,
9         FIRSTNAME varchar2(255 char),
10        LASTNAME varchar2(255 char),
11        primary key (EMPID)
12    )
13
14 Hibernate:
15     select
16         employee0_.EMPID as EMPID1_0_,
17         employee0_.CREATED_BY as CREATED_BY2_0_,
18         employee0_.CREATED_DATE as CREATED_DATE3_0_,
19         employee0_.FIRSTNAME as FIRSTNAME4_0_,
20         employee0_.LASTNAME as LASTNAME5_0_
21     from
22         EMPLOYEE employee0_
23
24 List Records:
25 Hibernate:
26     insert
27     into
28         EMPLOYEE
29             (CREATED_BY, CREATED_DATE, FIRSTNAME, LASTNAME, EMPID)
30     values
31         (?, ?, ?, ?, ?)
32
33 Insert Record:
34 Employee [empid: 1 lastName = Smith firstName: John createdBy: system
35             ↪ createdDate = 2016-01-22 11:40:01.653]
36
37 List Record:
38 Employee [empid: 1 lastName = Smith firstName: John createdBy: system
39             ↪ createdDate = 2016-01-22 11:40:01.653]
40 Hibernate:
41     update
42         EMPLOYEE
43     set
44         CREATED_BY=?,
45         CREATED_DATE=?,
46         FIRSTNAME=?,
47         LASTNAME=?
48     where

```

```

47      EMPID=?
48
49  Update Record:
50  Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
51      ↪   = 2016-01-22 11:40:01.653]
52
53  List Record:
54  Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
55      ↪   = 2016-01-22 11:40:01.653]
56  Hibernate:
57      insert
58      into
59          EMPLOYEE
60              (CREATED_BY, CREATED_DATE, FIRSTNAME, LASTNAME, EMPID)
61      values
62          (?, ?, ?, ?, ?)
63
64  Insert Record:
65  Employee [empid: 1001 lastName = Smith firstName: John createdBy: system
66      ↪   createdDate = 2016-01-22 11:40:01.704]
67  Hibernate:
68      insert
69      into
70          EMPLOYEE
71              (CREATED_BY, CREATED_DATE, FIRSTNAME, LASTNAME, EMPID)
72      values
73          (?, ?, ?, ?, ?)
74
75  Insert Record:
76  Employee [empid: 1002 lastName = Doe firstName: Jane createdBy: system
77      ↪   createdDate = 2016-01-22 11:40:01.707]
78  Hibernate:
79      insert
80      into
81          EMPLOYEE
82              (CREATED_BY, CREATED_DATE, FIRSTNAME, LASTNAME, EMPID)
83      values
84          (?, ?, ?, ?, ?)
85
86  Insert Record:
87  Employee [empid: 1003 lastName = Brown firstName: Charlie createdBy: system
88      ↪   createdDate = 2016-01-22 11:40:01.71]
89  Hibernate:
90      select
91          employee0_.EMPID as EMPID1_0_,
92          employee0_.CREATED_BY as CREATED_BY2_0_,
93          employee0_.CREATED_DATE as CREATED_DATE3_0_,
94          employee0_.FIRSTNAME as FIRSTNAME4_0_,
95          employee0_.LASTNAME as LASTNAME5_0_
96      from
97          EMPLOYEE employee0_
98
99  List Records:
100 Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
101    ↪   = 2016-01-22 11:40:01.653]
102 Employee [empid: 1001 lastName = Smith firstName: John createdBy: system
103    ↪   createdDate = 2016-01-22 11:40:01.704]
104 Employee [empid: 1002 lastName = Doe firstName: Jane createdBy: system
105    ↪   createdDate = 2016-01-22 11:40:01.707]
106 Employee [empid: 1003 lastName = Brown firstName: Charlie createdBy: system
107    ↪   createdDate = 2016-01-22 11:40:01.71]
108 Hibernate:
109     delete
110     from
111         EMPLOYEE
112     where
113         EMPID=?
114
115 Delete Record:
116 Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
117    ↪   = 2016-01-22 11:40:01.653]
118 Hibernate:
119     select

```

```

110     employee0_.EMPID as EMPID1_0_,
111     employee0_.CREATED_BY as CREATED_BY2_0_,
112     employee0_.CREATED_DATE as CREATED_DATE3_0_,
113     employee0_.FIRSTNAME as FIRSTNAME4_0_,
114     employee0_.LASTNAME as LASTNAME5_0_
115   from
116     EMPLOYEE employee0_
117
118 List Records:
119 Employee [empid: 1001 lastName = Smith firstName: John createdBy: system
120   ↪ createdDate = 2016-01-22 11:40:01.704]
121 Employee [empid: 1002 lastName = Doe firstName: Jane createdBy: system
122   ↪ createdDate = 2016-01-22 11:40:01.707]
123 Employee [empid: 1003 lastName = Brown firstName: Charlie createdBy: system
124   ↪ createdDate = 2016-01-22 11:40:01.71]
125
126 Hibernate:
127   drop table EMPLOYEE cascade constraints

```

15.3.2 JPA - Spring Data

15.3.2.1 Create project using command line

```

1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
2   ↪ -DartifactId=app-hello-orm-jpa-spring
3   ↪ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

```

15.3.2.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\app-hello-orm-jpa-spring\pom.xml

Use properties
 Add hsql, hibernate, jta, spring dependencies
 Modify junit version
 Add maven-compiler-plugin, set JDK version to 1.7

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   ↪ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4   ↪ http://maven.apache.org/maven-v4_0_.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <groupId>com.xyz</groupId>
7   <artifactId>app-hello-orm-jpa-spring</artifactId>
8   <packaging>jar</packaging>
9   <version>1.0-SNAPSHOT</version>
10  <name>app-hello-orm-jpa-spring</name>
11  <url>http://maven.apache.org</url>
12  <properties>
13    <hsql.version>2.3.3</hsql.version>
14    <java.version>1.7</java.version>
15    <junit.version>4.12</junit.version>
16    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
17    <hibernate.version>5.0.6.Final</hibernate.version>
18    <spring.version>4.2.0.RELEASE</spring.version>
19    <jta.version>1.1</jta.version>
20  </properties>
21  <dependencies>
22    <dependency>
23      <groupId>org.hsqldb</groupId>
24      <artifactId>hsqldb</artifactId>
25      <version>${hsql.version}</version>
26    </dependency>
27    <dependency>
28      <groupId>javax.transaction</groupId>
29      <artifactId>jta</artifactId>

```

```

28      <version>${jta.version}</version>
29  </dependency>
30  <dependency>
31      <groupId>junit</groupId>
32      <artifactId>junit</artifactId>
33      <version>${junit.version}</version>
34      <scope>test</scope>
35  </dependency>
36  <dependency>
37      <groupId>org.hibernate</groupId>
38      <artifactId>hibernate-entitymanager</artifactId>
39      <version>${hibernate.version}</version>
40  </dependency>
41  <dependency>
42      <groupId>org.springframework</groupId>
43      <artifactId>spring-context</artifactId>
44      <version>${spring.version}</version>
45  </dependency>
46  <dependency>
47      <groupId>org.springframework</groupId>
48      <artifactId>spring-core</artifactId>
49      <version>${spring.version}</version>
50  </dependency>
51  <dependency>
52      <groupId>org.springframework</groupId>
53      <artifactId>spring-jdbc</artifactId>
54      <version>${spring.version}</version>
55  </dependency>
56  <dependency>
57      <groupId>org.springframework</groupId>
58      <artifactId>spring-orm</artifactId>
59      <version>${spring.version}</version>
60  </dependency>
61  <dependency>
62      <groupId>org.springframework</groupId>
63      <artifactId>spring-tx</artifactId>
64      <version>${spring.version}</version>
65  </dependency>
66 </dependencies>
67 <build>
68     <finalName>app-hello-orm-jpa-spring</finalName>
69     <plugins>
70         <plugin>
71             <groupId>org.apache.maven.plugins</groupId>
72             <artifactId>maven-compiler-plugin</artifactId>
73             <version>${maven.compiler.plugin.version}</version>
74             <configuration>
75                 <source>${java.version}</source>
76                 <target>${java.version}</target>
77             </configuration>
78         </plugin>
79     </plugins>
80 </build>
81 </project>

```

15.3.2.3 Make this project an eclipse project

```

1 c:\workspaces\workspace\my-projects\app-hello-orm-jpa-spring>mvn eclipse:clean
  ↳ eclipse:eclipse

```

15.3.2.4 Import this project to eclipse

- Launch eclipse
- File -> Import -> General -> Existing Projects into Workspace
- Browse to app-hello-orm-jpa-spring, and import it.

15.3.2.5 Add a java model class

- src/main/java -> com.xyz.model -> Employee.java

```

1 package com.xyz.model;
2
3 import java.sql.Timestamp;
4
5 import javax.persistence.Column;
6 import javax.persistence.Entity;
7 import javax.persistence.Id;
8 import javax.persistence.Table;
9
10 @Entity
11 @Table(name = "EMPLOYEE")
12 public class Employee {
13     @Id
14     @Column(name = "EMPIID")
15     int empId;
16     @Column(name = "LASTNAME")
17     String lastName;
18     @Column(name = "FIRSTNAME")
19     String firstName;
20     @Column(name = "CREATED_BY")
21     String createdBy;
22     @Column(name = "CREATED_DATE")
23     Timestamp createdDate;
24
25     public int getEmpId() {
26         return empId;
27     }
28     public void setEmpId(int empId) {
29         this.empId = empId;
30     }
31     public String getLastname() {
32         return lastName;
33     }
34     public void setLastname(String lastName) {
35         this.lastName = lastName;
36     }
37     public String getFirstname() {
38         return firstName;
39     }
40     public void setFirstname(String firstName) {
41         this.firstName = firstName;
42     }
43     public String getCreatedBy() {
44         return createdBy;
45     }
46     public void setCreatedBy(String createdBy) {
47         this.createdBy = createdBy;
48     }
49     public Timestamp getCreatedDate() {
50         return createdDate;
51     }
52     public void setCreatedDate(Timestamp createdDate) {
53         this.createdDate = createdDate;
54     }
55
56     @Override
57     public String toString() {
58         return "Employee [empid: " + empId + " lastName = " + lastName + "
59             ← firstName: " + firstName + " createdBy: " + createdBy + "
60             ← createdDate = " + createdDate + "]";
61     }
62 }
```

15.3.2.6 Add java dao classes

- src/main/java -> com.xyz.dao -> EmployeeDAO.java

```

1 package com.xyz.dao;
2
```

```

3 import java.util.List;
4
5 import com.xyz.model.Employee;
6
7 public interface EmployeeDAO {
8     public void insertRecord(Employee employee);
9     public void updateRecord(int empId, String lastName);
10    public void deleteRecord(int empId);
11    public Employee listRecord(int empId);
12    public List<Employee> listRecords();
13 }

```

src/main/java ->com.xyz.dao ->EmployeeDAOImpl.java

```

1 package com.xyz.dao;
2
3 import java.util.List;
4
5 import javax.persistence.EntityManager;
6 import javax.persistence.PersistenceContext;
7
8 import org.springframework.stereotype.Repository;
9 import org.springframework.transaction.annotation.Propagation;
10 import org.springframework.transaction.annotation.Transactional;
11
12 import com.xyz.model.Employee;
13
14 @Repository("EmployeeDAOImpl")
15 @Transactional(propagation = Propagation.REQUIRED)
16 public class EmployeeDAOImpl implements EmployeeDAO {
17     @PersistenceContext
18     private EntityManager entityManager;
19
20     @Override
21     public void insertRecord(Employee employee) {
22         entityManager.persist(employee);
23         System.out.println("\nInsert Record:");
24         System.out.println(employee.toString());
25     }
26
27     @Override
28     public void updateRecord(int empId, String lastName) {
29         Employee employee = (Employee) entityManager.find(Employee.class, new
30             Integer(empId));
31         if (null != employee) {
32             employee.setLastName(lastName);
33             System.out.println("\nUpdate Record:");
34             System.out.println(employee.toString());
35         }
36     }
37
38     @Override
39     public void deleteRecord(int empId) {
40         Employee employee = (Employee) entityManager.find(Employee.class, new
41             Integer(empId));
42         if (null != employee) {
43             entityManager.remove(employee);
44             System.out.println("\nDelete Record:");
45             System.out.println(employee.toString());
46         }
47     }
48     @Override
49     public Employee listRecord(int empId) {
50         Employee employee = (Employee) entityManager.find(Employee.class, new
51             Integer(empId));
52         System.out.println("\nList Record:");
53         System.out.println(employee.toString());
54         return employee;
55     }
56     @SuppressWarnings("unchecked")

```

```

56     @Override
57     public List<Employee> listRecords() {
58         List<Employee> employees = (List<Employee>)
59             entityManager.createQuery("FROM Employee").getResultList();
60         System.out.println("\nList Records:");
61         for (Employee employee : employees) {
62             System.out.println(employee.toString());
63         }
64     }
65 }
```

15.3.2.7 Configure spring bean and JPA

- Create resources directory and add it to Build Path
src/main/resources
- Configuration metadata
src/main/resources/app-hello-orm-jpa-spring.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <beans xmlns="http://www.springframework.org/schema/beans"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4          xsi:schemaLocation="http://www.springframework.org/schema/beans
5              http://www.springframework.org/schema/beans/spring-beans.xsd
6              http://www.springframework.org/schema/context
7              http://www.springframework.org/schema/context/spring-context.xsd
8              http://www.springframework.org/schema/jdbc
9                  http://www.springframework.org/schema/jdbc/spring-jdbc.xsd
10             http://www.springframework.org/schema/tx
11                 http://www.springframework.org/schema/tx/spring-tx.xsd">
12
13     <context:annotation-config />
14
15     <context:component-scan base-package="com.xyz" />
16
17     <bean id="employeeDAO" class="com.xyz.dao.EmployeeDAOImpl" />
18
19     <jdbc:embedded-database type="HSQL" id="dataSource">
20         <jdbc:script location="classpath:sql/create-db.sql" />
21         <jdbc:script location="classpath:sql/insert-data.sql" />
22     </jdbc:embedded-database>
23
24     <bean id="jpaDialect"
25         class="org.springframework.orm.jpa.vendor.HibernateJpaDialect" />
26
27     <bean id="entityManagerFactory"
28         class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
29         <property name="persistenceUnitName" value="orm-jpa-spring" />
30         <property name="dataSource" ref="dataSource" />
31         <property name="persistenceXmlLocation" value="META-INF/persistence.xml" />
32         <property name="jpaVendorAdapter" ref="jpaVendorAdapter" />
33         <property name="jpaDialect" ref="jpaDialect" />
34         <property name="jpaProperties">
35             <props>
36                 <prop key="hibernate.hbm2ddl.auto">validate</prop>
37                 <prop key="hibernate.dialect">org.hibernate.dialect.HSQLDialect</prop>
38             </props>
39         </property>
40     </bean>
41
42     <bean id="jpaVendorAdapter"
43         class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
44     </bean>
45
46     <bean id="txManager"
47         class="org.springframework.orm.jpa.JpaTransactionManager">
```

```

45   <property name="entityManagerFactory" ref="entityManagerFactory" />
46   <property name="dataSource" ref="dataSource" />
47   <property name="jpaDialect" ref="jpaDialect" />
48 </bean>
49
50 <tx:annotation-driven transaction-manager="txManager" />
51
52 </beans>
```

src/main/resources/META-INF/persistence.xml

```

1 <persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
4   ↪ http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
4   version="2.1">
5
6   <persistence-unit name="orm-jpa-spring" transaction-type="RESOURCE_LOCAL">
7     <class>com.xyz.model.Employee</class>
8   </persistence-unit>
9
10 </persistence>
```

15.3.2.8 SQL

- Create sql directory under resources directory
src/main/resources/sql
- Create Table
src/main/resources/sql/create-table.sql

```

1 CREATE TABLE EMPLOYEE (
2   EMPID INTEGER NOT NULL,
3   LASTNAME VARCHAR(20) NOT NULL,
4   FIRSTNAME VARCHAR(20) NOT NULL,
5   CREATED_BY VARCHAR(20) NOT NULL,
6   CREATED_DATE TIMESTAMP NOT NULL,
7   PRIMARY KEY (EMPID)
8 );
```

src/main/resources/sql/insert-data.sql

```

1 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
2   ↪ VALUES (1001, 'Smith', 'John', 'system', CURRENT_DATE);
2 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
3   ↪ VALUES (1002, 'Doe', 'Jane', 'system', CURRENT_DATE);
3 INSERT INTO EMPLOYEE (EMPID, LASTNAME, FIRSTNAME, CREATED_BY, CREATED_DATE)
4   ↪ VALUES (1003, 'Brown', 'Charlie', 'system', CURRENT_DATE);
```

15.3.2.9 Instantiating a container

- App
Type in the code list below to src/main/java -> com.xyz -> App.java

```

1 package com.xyz;
2
3 import java.sql.Timestamp;
4 import java.util.Date;
5
6 import org.springframework.context.ApplicationContext;
7 import org.springframework.context.support.ClassPathXmlApplicationContext;
8
9 import com.xyz.dao.EmployeeDAO;
10 import com.xyz.model.Employee;
11
```

```

12  public class App {
13      public static void main(String[] args) {
14          ApplicationContext context = new
15              ClassPathXmlApplicationContext("app-hello-orm-jpa-spring.xml");
16          EmployeeDAO dao = (EmployeeDAO) context.getBean("employeeDAO");
17
18          dao.listRecords();
19
20          Employee employee = new Employee();
21          employee.setEmpId(1);
22          employee.setLastName("Smith");
23          employee.setFirstName("John");
24          employee.setCreatedBy("system");
25          employee.setCreatedDate(new Timestamp((new Date()).getTime()));
26          dao.insertRecord(employee);
27          dao.listRecord(1);
28
29          dao.updateRecord(1, "Doe");
30          dao.listRecord(1);
31
32          dao.listRecords();
33          dao.deleteRecord(1);
34          dao.listRecords();
35
36      }
37  }

```

15.3.2.10 Run App

Right click on project src/main/java ->com.xyz->App

Run As ->Java Application

You will see in the console:

```

1 List Records:
2 Employee [empid: 1001 lastName = Smith firstName: John createdBy: system
2   ↪   createdDate = 2016-01-21 00:00:00.0]
3 Employee [empid: 1002 lastName = Doe firstName: Jane createdBy: system
3   ↪   createdDate = 2016-01-21 00:00:00.0]
4 Employee [empid: 1003 lastName = Brown firstName: Charlie createdBy: system
4   ↪   createdDate = 2016-01-21 00:00:00.0]
5
6 Insert Record:
7 Employee [empid: 1 lastName = Smith firstName: John createdBy: system
7   ↪   createdDate = 2016-01-21 17:11:22.965]
8
9 List Record:
10 Employee [empid: 1 lastName = Smith firstName: John createdBy: system
10   ↪   createdDate = 2016-01-21 17:11:22.965]
11
12 Update Record:
13 Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
13   ↪   = 2016-01-21 17:11:22.965]
14
15 List Record:
16 Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
16   ↪   = 2016-01-21 17:11:22.965]
17
18 List Records:
19 Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
19   ↪   = 2016-01-21 17:11:22.965]
20 Employee [empid: 1001 lastName = Smith firstName: John createdBy: system
20   ↪   createdDate = 2016-01-21 00:00:00.0]
21 Employee [empid: 1002 lastName = Doe firstName: Jane createdBy: system
21   ↪   createdDate = 2016-01-21 00:00:00.0]
22 Employee [empid: 1003 lastName = Brown firstName: Charlie createdBy: system
22   ↪   createdDate = 2016-01-21 00:00:00.0]
23
24 Delete Record:
25 Employee [empid: 1 lastName = Doe firstName: John createdBy: system createdDate
25   ↪   = 2016-01-21 17:11:22.965]

```

```
26
27 List Records:
28 Employee [empid: 1001 lastName = Smith firstName: John createdBy: system
29   ↳ createdDate = 2016-01-21 00:00:00.0]
30 Employee [empid: 1002 lastName = Doe firstName: Jane createdBy: system
31   ↳ createdDate = 2016-01-21 00:00:00.0]
32 Employee [empid: 1003 lastName = Brown firstName: Charlie createdBy: system
33   ↳ createdDate = 2016-01-21 00:00:00.0]
```

PART V

ADVANCED PROJECTS

CHAPTER 16

WEB SERVICE

16.1 REST

16.1.1 REST Service - Using Jersey

16.1.1.1 Create project using command line

```
1 c:\workspaces\eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↵ -DartifactId=hello-rest-service
  ↵ -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false
```

16.1.1.2 Modify pom.xml

- C:\workspaces\eclipse\my-projects\hello-rest-service\pom.xml
- Use properties
- Add jersey, json dependencies
- Modify junit version
- Add maven-compiler-plugin, set JDK version to 1.7

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↵   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  ↵   http://maven.apache.org/maven-v4_0_0.xsd">
```

[†]change c:\workspaces\eclipse to /users/xyz/documents/workspace, then all the projects in this chapter will work in mac osx.

```

3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>hello-rest-service</artifactId>
6   <packaging>war</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>hello-rest-service Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <jersey.version>1.19</jersey.version>
12    <json.version>20151123</json.version>
13    <java.version>1.7</java.version>
14    <junit.version>4.12</junit.version>
15    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
16  </properties>
17  <dependencies>
18    <dependency>
19      <groupId>com.sun.jersey</groupId>
20      <artifactId>jersey-bundle</artifactId>
21      <version>${jersey.version}</version>
22    </dependency>
23    <dependency>
24      <groupId>com.sun.jersey</groupId>
25      <artifactId>jersey-core</artifactId>
26      <version>${jersey.version}</version>
27    </dependency>
28    <dependency>
29      <groupId>org.json</groupId>
30      <artifactId>json</artifactId>
31      <version>${json.version}</version>
32    </dependency>
33    <dependency>
34      <groupId>junit</groupId>
35      <artifactId>junit</artifactId>
36      <version>${junit.version}</version>
37      <scope>test</scope>
38    </dependency>
39  </dependencies>
40  <build>
41    <finalName>hello-rest-service</finalName>
42    <plugins>
43      <plugin>
44        <groupId>org.apache.maven.plugins</groupId>
45        <artifactId>maven-compiler-plugin</artifactId>
46        <version>${maven.compiler.plugin.version}</version>
47        <configuration>
48          <source>${java.version}</source>
49          <target>${java.version}</target>
50        </configuration>
51      </plugin>
52    </plugins>
53  </build>
54</project>
```

16.1.1.3 Make this project an eclipse project

```
1 c:\workspaces\workspace\my-projects\hello-rest-service>mvn eclipse:clean
→ eclipse:eclipse -Dwtpversion=2.0
```

16.1.1.4 Import this project to eclipse

- Launch eclipse
File → Import → General → Existing Projects into Workspace
Browse to hello-rest-service, and import it.

16.1.1.5 Service Classes

- Create java directory and add it to Build Path
src/main/java

- Create new java service classes
src/main/java/com/xyz/service/CtoFService.java

```

1 package com.xyz.service;
2
3 import javax.ws.rs.GET;
4 import javax.ws.rs.Path;
5 import javax.ws.rs.PathParam;
6 import javax.ws.rs.Produces;
7
8 @Path("/ctofservice")
9 public class CtoFService {
10     @GET
11     @Produces("application/xml")
12     public String convertCtoF() {
13         Double fahrenheit;
14         Double celsius = 36.8;
15         fahrenheit = ((celsius * 9) / 5) + 32;
16
17         String result = "@Produces(\"application/xml\") Output: \n\nC to F
18             ↪ Converter Output: \n\n" + fahrenheit;
19         return "<ctofservice>" + "<celsius>" + celsius + "</celsius>" +
20             ↪ "<ctofoutput>" + result + "</ctofoutput>" + "</ctofservice>";
21     }
22
23     @Path("{c}")
24     @GET
25     @Produces("application/xml")
26     public String convertCtoFfromInput(@PathParam("c") Double c) {
27         Double fahrenheit;
28         Double celsius = c;
29         fahrenheit = ((celsius * 9) / 5) + 32;
30
31         String result = "@Produces(\"application/xml\") Output: \n\nC to F
32             ↪ Converter Output: \n\n" + fahrenheit;
33         return "<ctofservice>" + "<celsius>" + celsius + "</celsius>" +
34             ↪ "<ctofoutput>" + result + "</ctofoutput>" + "</ctofservice>";
35     }
36 }
```

src/main/java/com/xyz/service/FtoCService.java

```

1 package com.xyz.service;
2
3 import javax.ws.rs.GET;
4 import javax.ws.rs.Path;
5 import javax.ws.rs.PathParam;
6 import javax.ws.rs.Produces;
7 import javax.ws.rs.core.Response;
8
9 import org.json.JSONException;
10 import org.json.JSONObject;
11
12 @Path("/ftocservice")
13 public class FtoCService {
14     @GET
15     @Produces("application/json")
16     public Response convertFtoC() throws JSONException {
17         JSONObject jsonObject = new JSONObject();
18         Double fahrenheit = 98.24;
19         Double celsius;
20         celsius = (fahrenheit - 32) * 5 / 9;
21         jsonObject.put("F Value", fahrenheit);
22         jsonObject.put("C Value", celsius);
23
24         String result = "@Produces(\"application/json\") Output: \n\nF to C
25             ↪ Converter Output: \n\n" + jsonObject;
```

```

25         return Response.status(200).entity(result).build();
26     }
27
28     @Path("{f}")
29     @GET
30     @Produces("application/json")
31     public Response convertFtoCfromInput(@PathParam("f") float f) throws
32         JSONException {
33         JSONObject jsonObject = new JSONObject();
34         float celsius;
35         celsius = (f - 32) * 5 / 9;
36         jsonObject.put("F Value", f);
37         jsonObject.put("C Value", celsius);
38
39         String result = "@Produces(\"application/json\") Output: \n\nF to C
40             ↪ Converter Output: \n\n" + jsonObject;
41         return Response.status(200).entity(result).build();
42     }
43 }
```

16.1.1.6 Integrate web application - web.xml

- src/main/webapp/WEB-INF/web.xml

```

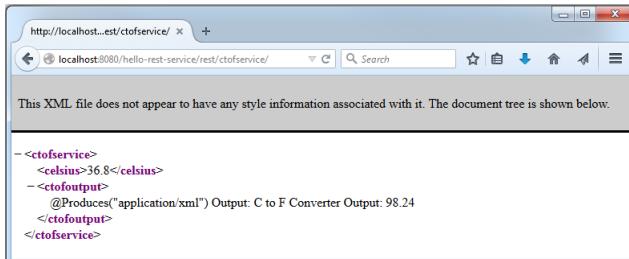
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
3   ↪ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
5   ↪ http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
6   version="2.4">
7
8   <display-name>Hello Rest Service</display-name>
9
10  <servlet>
11    <servlet-name>Hello Rest Service Web Application</servlet-name>
12
13    ↪ <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
14    <load-on-startup>1</load-on-startup>
15  </servlet>
16
17  <servlet-mapping>
18    <servlet-name>Hello Rest Service Web Application</servlet-name>
19    <url-pattern>/rest/*</url-pattern>
20  </servlet-mapping>
21
22 </web-app>
```

16.1.1.7 Build the project - command line

```
c:\workspaces\workspace\my-projects\hello-rest-service>mvn install
```

16.1.1.8 Run

- Add project to Tomcat Server
Go to Servers Tab
Right click Tomcat v7.0 Server at localhost [Stopped] → Add and Remove...
Move hello-rest-service from Available panel (left side) to Configured panel (right side)
- Start Tomcat Server
<http://localhost:8080/hello-rest-service/rest/ctofservice/>



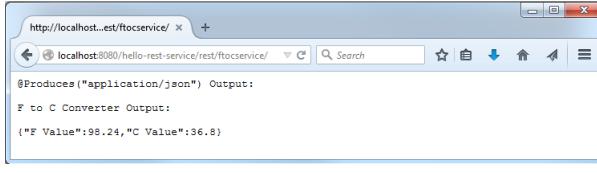
```
<ctofservice>
  <celsius>36.8</celsius>
  - <ctofoutput>
    @Produces("application/xml") Output: C to F Converter Output: 98.24
  </ctofoutput>
</ctofservice>
```

<http://localhost:8080/hello-rest-service/rest/ctofservice/35>



```
<ctofservice>
  <celsius>35.0</celsius>
  - <ctofoutput>
    @Produces("application/xml") Output: C to F Converter Output: 95.0
  </ctofoutput>
</ctofservice>
```

<http://localhost:8080/hello-rest-service/rest/ftocservice/>



```
@Produces("application/json") Output:
F to C Converter Output:
{"F Value":98.24,"C Value":36.8}
```

<http://localhost:8080/hello-rest-service/rest/ftocservice/98>



```
@Produces("application/json") Output:
F to C Converter Output:
{"F Value":98,"C Value":36.6666793823242}
```

<http://localhost:8080/hello-rest-service/rest/ctofservice/>

```
<ctofservice>
  <celsius>36.8</celsius>
  - <ctofoutput>
    @Produces("application/xml") Output: C to F Converter Output: 98.24
  </ctofoutput>
</ctofservice>
```

<http://localhost:8080/hello-rest-service/rest/ctofservice/35>

```
<ctofservice>
  <celsius>35.0</celsius>
  - <ctofoutput>
    @Produces("application/xml") Output: C to F Converter Output: 95.0
  </ctofoutput>
</ctofservice>
```

<http://localhost:8080/hello-rest-service/rest/ftocservice/>

```
@Produces("application/json") Output:
F to C Converter Output:
{"F Value":98.24,"C Value":36.8}
```

```
http://localhost:8080/hello-rest-service/rest/ftocservice/98
@Produces("application/json") Output:
F to C Converter Output:
{"F Value":98,"C Value":36.66666793823242}
```

16.1.2 REST Client - Using Java URL

16.1.2.1 Create project using command line

```
1 c:\workspaces\eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
   ↳ -DartifactId=app-hello-rest-client-url
   ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

16.1.2.2 Modify pom.xml

- C:\workspaces\eclipse\my-projects\app-hello-rest-client-url\pom.xml

Use properties

Add commons-logging, log4j, slf4j dependencies

Modify junit version

Add maven-compiler-plugin, set JDK version to 1.7

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
   ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
   ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-rest-client-url</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-rest-client-url</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <commons-logging.version>1.2</commons-logging.version>
12    <java.version>1.7</java.version>
13    <junit.version>4.12</junit.version>
14    <log4j.version>1.2.17</log4j.version>
15    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
16    <slf4j.version>1.7.14</slf4j.version>
17  </properties>
18  <dependencies>
19    <dependency>
20      <groupId>commons-logging</groupId>
21      <artifactId>commons-logging</artifactId>
22      <version>${commons-logging.version}</version>
23    </dependency>
24    <dependency>
25      <groupId>junit</groupId>
26      <artifactId>junit</artifactId>
27      <version>${junit.version}</version>
28      <scope>test</scope>
29    </dependency>
30    <dependency>
31      <groupId>log4j</groupId>
32      <artifactId>log4j</artifactId>
33      <version>${log4j.version}</version>
34    </dependency>
35    <dependency>
36      <groupId>org.slf4j</groupId>
37      <artifactId>slf4j-api</artifactId>
38      <version>${slf4j.version}</version>
```

```

39      </dependency>
40      <dependency>
41          <groupId>org.slf4j</groupId>
42          <artifactId>slf4j-log4j12</artifactId>
43          <version>${slf4j.version}</version>
44      </dependency>
45  </dependencies>
46  <build>
47      <finalName>app-hello-rest-client-url</finalName>
48      <plugins>
49          <plugin>
50              <groupId>org.apache.maven.plugins</groupId>
51              <artifactId>maven-compiler-plugin</artifactId>
52              <version>${maven.compiler.plugin.version}</version>
53              <configuration>
54                  <source>${java.version}</source>
55                  <target>${java.version}</target>
56              </configuration>
57          </plugin>
58      </plugins>
59  </build>
60 </project>

```

16.1.2.3 Make this project an eclipse project

```

1 c:\workspaces\eclipse\my-projects\app-hello-rest-client-url>mvn eclipse:clean
  ↳ eclipse:eclipse

```

16.1.2.4 Import this project to eclipse

- Launch eclipse
File ->Import ->General ->Existing Projects into Workspace
Browse to app-hello-rest-client-url, and import it.

16.1.2.5 Configure log properties

- Create resources directory and add it to Build Path
src/main/resources
src/main/resources/log4j.properties

```

1 # Root logger option
2 log4j.rootLogger=INFO, file, stdout
3
4 # Direct log messages to a log file
5 log4j.appender.file=org.apache.log4j.RollingFileAppender
6 log4j.appender.file.File=system.out.log
7 log4j.appender.file.MaxFileSize=10MB
8 log4j.appender.file.MaxBackupIndex=10
9 log4j.appender.file.layout=org.apache.log4j.PatternLayout
10 log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
   ↳ %c{1}:%L - %m%
11
12 # Direct log messages to stdout
13 log4j.appender.stdout=org.apache.log4j.ConsoleAppender
14 log4j.appender.stdout.Target=System.out
15 log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
16 log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
   ↳ %c{1}:%L - %m%

```

16.1.2.6 App class

- Type in the code list below to src/main/java ->com.xyz ->App.java

```

1 package com.xyz;
2
3 import java.io.BufferedReader;
4 import java.io.InputStreamReader;
5 import java.net.URL;
6 import java.net.URLConnection;
7 import java.nio.charset.Charset;
8
9 import org.apache.commons.logging.Log;
10 import org.apache.commons.logging.LogFactory;
11
12 public class App {
13     private static final Log log = LogFactory.getLog(App.class);
14
15     public App() {
16     }
17
18     public String connect(String myURL) {
19         log.info("Requested URL: " + myURL);
20         StringBuilder sb = new StringBuilder();
21         URLConnection urlConn = null;
22         InputStreamReader in = null;
23         try {
24             URL url = new URL(myURL);
25             urlConn = url.openConnection();
26             if (urlConn != null)
27                 urlConn.setReadTimeout(60 * 1000);
28             if (urlConn != null && urlConn.getInputStream() != null) {
29                 in = new InputStreamReader(urlConn.getInputStream(),
30                     Charset.defaultCharset());
31                 BufferedReader bufferedReader = new BufferedReader(in);
32                 if (bufferedReader != null) {
33                     int cp;
34                     while ((cp = bufferedReader.read()) != -1) {
35                         sb.append((char) cp);
36                     }
37                     bufferedReader.close();
38                 }
39                 in.close();
40             } catch (Exception e) {
41                 throw new RuntimeException("Exception while calling URL:" + myURL,
42                     e);
43             }
44         return sb.toString();
45     }
46
47     public static void main(String[] args) {
48         App client = new App();
49
50         log.info(client.connect("http://localhost:8080/hello-rest-service/rest/ctofservice/35"));
51         log.info(client.connect("http://localhost:8080/hello-rest-service/rest/ftocservice/98"));
52     }
}

```

16.1.2.7 Run App

Make sure REST service is running.

Then right click on project src/main/java ->com.xyz ->App

Run As ->Java Application

You will see in the console and in the log file (system.out.log):

```

2016-01-25 17:23:57 INFO App:19 - Requested URL:
  ↳ http://localhost:8080/hello-rest-service/rest/ctofservice/35
2016-01-25 17:23:57 INFO App:49 -
  ↳ <ctofservice><celsius>35.0</celsius><ctofoutput>@Produces("application/xml")
  ↳ Output:

```

C to F Converter Output:

```
95.0</ctofoutput></ctofservice>
2016-01-25 17:23:57 INFO App:19 - Requested URL:
→ http://localhost:8080/hello-rest-service/rest/ftocservice/98
2016-01-25 17:23:57 INFO App:50 - @Produces("application/json") Output:
F to C Converter Output:
{"F Value":98, "C Value":36.66666793823242}
```

16.1.3 REST Client - Using Apache HTTP Client

16.1.3.1 Create project using command line

```
1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
→ -DartifactId=app-hello-rest-client-httpclient
→ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

16.1.3.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\app-hello-rest-client-httpclient\pom.xml

Use properties

Add httpclient, commons-logging, log4j, slf4j dependencies

Modify junit version

Add maven-compiler-plugin, set JDK version to 1.7

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
→   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
→     http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-rest-client-httpclient</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-rest-client-httpclient</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <commons-logging.version>1.2</commons-logging.version>
12    <httpclient.version>4.5.1</httpclient.version>
13    <httpcore.version>4.4.4</httpcore.version>
14    <java.version>1.7</java.version>
15    <junit.version>4.12</junit.version>
16    <log4j.version>1.2.17</log4j.version>
17    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
18    <slf4j.version>1.7.14</slf4j.version>
19  </properties>
20  <dependencies>
21    <dependency>
22      <groupId>commons-logging</groupId>
23      <artifactId>commons-logging</artifactId>
24      <version>${commons-logging.version}</version>
25    </dependency>
26    <dependency>
27      <groupId>org.apache.httpcomponents</groupId>
28      <artifactId>httpclient</artifactId>
29      <version>${httpclient.version}</version>
30    </dependency>
31    <dependency>
32      <groupId>org.apache.httpcomponents</groupId>
33      <artifactId>httpcore</artifactId>
34      <version>${httpcore.version}</version>
```

```

35      </dependency>
36      <dependency>
37          <groupId>junit</groupId>
38          <artifactId>junit</artifactId>
39          <version>${junit.version}</version>
40          <scope>test</scope>
41      </dependency>
42      <dependency>
43          <groupId>log4j</groupId>
44          <artifactId>log4j</artifactId>
45          <version>${log4j.version}</version>
46      </dependency>
47      <dependency>
48          <groupId>org.slf4j</groupId>
49          <artifactId>slf4j-api</artifactId>
50          <version>${slf4j.version}</version>
51      </dependency>
52      <dependency>
53          <groupId>org.slf4j</groupId>
54          <artifactId>slf4j-log4j12</artifactId>
55          <version>${slf4j.version}</version>
56      </dependency>
57  </dependencies>
58  <build>
59      <finalName>app-hello-rest-client-httpclient</finalName>
60      <plugins>
61          <plugin>
62              <groupId>org.apache.maven.plugins</groupId>
63              <artifactId>maven-compiler-plugin</artifactId>
64              <version>${maven.compiler.plugin.version}</version>
65              <configuration>
66                  <source>${java.version}</source>
67                  <target>${java.version}</target>
68              </configuration>
69          </plugin>
70      </plugins>
71  </build>
72</project>
```

16.1.3.3 Make this project an eclipse project

```

1 c:\workspaces\workspace\my-projects\app-hello-rest-client->mvn
  ↳ eclipse:clean eclipse:eclipse
```

16.1.3.4 Import this project to eclipse

- Launch eclipse
File -> Import -> General -> Existing Projects into Workspace
Browse to app-hello-rest-client-httpclient, and import it.

16.1.3.5 Configure log properties

- Create resources directory and add it to Build Path
src/main/resources
src/main/resources/log4j.properties
-

```

1 # Root logger option
2 log4j.rootLogger=INFO, file, stdout
3
4 # Direct log messages to a log file
5 log4j.appender.file=org.apache.log4j.RollingFileAppender
6 log4j.appender.file.File=system.out.log
7 log4j.appender.file.MaxFileSize=10MB
8 log4j.appender.file.MaxBackupIndex=10
9 log4j.appender.file.layout=org.apache.log4j.PatternLayout
```

```

10 log4j.appenders.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
   ↪ %c{1}:%L - %m%n
11
12 # Direct log messages to stdout
13 log4j.appenders.stdout=org.apache.log4j.ConsoleAppender
14 log4j.appenders.stdout.Target=System.out
15 log4j.appenders.stdout.layout=org.apache.log4j.PatternLayout
16 log4j.appenders.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
   ↪ %c{1}:%L - %m%n

```

16.1.3.6 App class

- Type in the code list below to src/main/java -> com.xyz -> App.java

```

1 package com.xyz;
2
3 import java.io.BufferedReader;
4 import java.io.InputStreamReader;
5
6 import org.apache.commons.logging.Log;
7 import org.apache.commons.logging.LogFactory;
8 import org.apache.http.client.methods.CloseableHttpResponse;
9 import org.apache.http.client.methods.HttpGet;
10 import org.apache.http.impl.client.CloseableHttpClient;
11 import org.apache.http.impl.client.HttpClients;
12
13 public class App {
14     private static final Log log = LogFactory.getLog(App.class);
15
16     public App() {
17     }
18
19     public void connect(String url) {
20         try {
21             // Create default HTTP Client
22             CloseableHttpClient httpClient = HttpClients.createDefault();
23             // Create new getRequest with below mentioned URL
24             HttpGet getRequest = new HttpGet(url);
25             // Add additional header to request which accepts application/xml
26             ↪ data
27             getRequest.addHeader("accept", "application/xml");
28             // Execute your request and catch response
29             CloseableHttpResponse response = httpClient.execute(getRequest);
23             // Check for HTTP response code: 200 = success
24             if (response.getStatusLine().getStatusCode() != 200) {
25                 throw new RuntimeException("Failed : HTTP error code : " +
26                   ↪ response.getStatusLine().getStatusCode());
27             }
28             // Get-Capture Complete application/xml body response
29             BufferedReader br = new BufferedReader(new
30             ↪ InputStreamReader((response.getEntity().getContent())));
31             String output;
32             log.info("=====Output=====");
33
34             // Simply iterate through XML response and show on console.
35             while ((output = br.readLine()) != null) {
36                 log.info(output);
37             }
38
39             response.close();
40             httpClient.close();
41         } catch (Exception e) {
42             log.error(e.getMessage());
43         }
44     }
45
46     public static void main(String[] args) {
47         App client = new App();
48
49         ↪ client.connect("http://localhost:8080/hello-rest-service/rest/ctofservice/35");
50     }
51 }
52

```

16.1.3.7 Run App

Make sure REST service is running.

Then right click on project src/main/java ->com.xyz->App

Run As ->Java Application

You will see in the console and in the log file (system.out.log):

```
2016-01-25 18:02:33 INFO App:36 - =====Output=====
2016-01-25 18:02:33 INFO App:40 -
→ <ctofservice><celsius>35.0</celsius><ctofoutput>@Produces("application/xml")
→ Output:
2016-01-25 18:02:33 INFO App:40 -
2016-01-25 18:02:33 INFO App:40 - C to F Converter Output:
2016-01-25 18:02:33 INFO App:40 -
2016-01-25 18:02:33 INFO App:40 - 95.0</ctofoutput></ctofservice>
```

16.1.4 REST Client - Using Jersey Client

16.1.4.1 Create project using command line

```
1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
→ -DartifactId=app-hello-rest-client-jersey
→ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

16.1.4.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\app-hello-rest-client-jersey\pom.xml

Use properties

Add jersey, commons-logging, log4j, slf4j dependencies

Modify junit version

Add maven-compiler-plugin, set JDK version to 1.7

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
→   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
→     http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>app-hello-rest-client-jersey</artifactId>
6   <packaging>jar</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>app-hello-rest-client-jersey</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <commons-logging.version>1.2</commons-logging.version>
12    <java.version>1.7</java.version>
13    <jersey.version>1.19</jersey.version>
14    <junit.version>4.12</junit.version>
15    <log4j.version>1.2.17</log4j.version>
16    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
17    <slf4j.version>1.7.14</slf4j.version>
18  </properties>
19  <dependencies>
20    <dependency>
21      <groupId>commons-logging</groupId>
22      <artifactId>commons-logging</artifactId>
23      <version>${commons-logging.version}</version>
24    </dependency>
```

```

25   <dependency>
26     <groupId>com.sun.jersey</groupId>
27     <artifactId>jersey-client</artifactId>
28     <version>${jersey.version}</version>
29   </dependency>
30   <dependency>
31     <groupId>junit</groupId>
32     <artifactId>junit</artifactId>
33     <version>${junit.version}</version>
34     <scope>test</scope>
35   </dependency>
36   <dependency>
37     <groupId>log4j</groupId>
38     <artifactId>log4j</artifactId>
39     <version>${log4j.version}</version>
40   </dependency>
41   <dependency>
42     <groupId>org.slf4j</groupId>
43     <artifactId>slf4j-api</artifactId>
44     <version>${slf4j.version}</version>
45   </dependency>
46   <dependency>
47     <groupId>org.slf4j</groupId>
48     <artifactId>slf4j-log4j12</artifactId>
49     <version>${slf4j.version}</version>
50   </dependency>
51 </dependencies>
52 <build>
53   <finalName>app-hello-rest-client-jersey</finalName>
54   <plugins>
55     <plugin>
56       <groupId>org.apache.maven.plugins</groupId>
57       <artifactId>maven-compiler-plugin</artifactId>
58       <version>${maven.compiler.plugin.version}</version>
59       <configuration>
60         <source>${java.version}</source>
61         <target>${java.version}</target>
62       </configuration>
63     </plugin>
64   </plugins>
65 </build>
66 </project>

```

16.1.4.3 Make this project an eclipse project

```

1 c:\workspaces\workspace\my-projects\app-hello-rest-client-jersey>mvn eclipse:clean
  ↵  eclipse:eclipse

```

16.1.4.4 Import this project to eclipse

- Launch eclipse
File → Import → General → Existing Projects into Workspace
Browse to app-hello-rest-client-jersey, and import it.

16.1.4.5 Configure log properties

- Create resources directory and add it to Build Path
src/main/resources
src/main/resources/log4j.properties
-

```

1 # Root logger option
2 log4j.rootLogger=INFO, file, stdout
3
4 # Direct log messages to a log file
5 log4j.appenders.file=org.apache.log4j.RollingFileAppender

```

```

6  log4j.appender.file.File=system.out.log
7  log4j.appender.file.MaxFileSize=10MB
8  log4j.appender.file.MaxBackupIndex=10
9  log4j.appender.file.layout=org.apache.log4j.PatternLayout
10 log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
    ↳ %c{1}:%L - %m%n
11
12 # Direct log messages to stdout
13 log4j.appender.stdout=org.apache.log4j.ConsoleAppender
14 log4j.appender.stdout.Target=System.out
15 log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
16 log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
    ↳ %c{1}:%L - %m%n

```

16.1.4.6 App class

- Type in the code list below to src/main/java ->com.xyz ->App.java

```

1 package com.xyz;
2
3 import org.apache.commons.logging.Log;
4 import org.apache.commons.logging.LogFactory;
5
6 import com.sun.jersey.api.client.Client;
7 import com.sun.jersey.api.client.ClientResponse;
8 import com.sun.jersey.api.client.WebResource;
9
10 public class App {
11     private static final Log log = LogFactory.getLog(App.class);
12
13     public App() {
14     }
15
16     public void connect(String url, String format) {
17         try {
18             Client client = Client.create();
19             WebResource webResource = client.resource(url);
20             ClientResponse response =
21                 → webResource.accept(format).get(ClientResponse.class);
22             if (response.getStatus() != 200) {
23                 → throw new RuntimeException("Failed : HTTP error code : " +
24                     → response.getStatus());
25         }
26
27         String output = response.getEntity(String.class);
28         log.info("\n=====Response=====");
29         log.info(output);
30     } catch (Exception e) {
31         e.printStackTrace();
32     }
33
34     public static void main(String[] args) {
35         App client = new App();
36
37         → client.connect("http://localhost:8080/hello-rest-service/rest/ftocservice/90",
38             → "application/json");
39
40         → client.connect("http://localhost:8080/hello-rest-service/rest/ctofservice/40",
41             → "application/xml");
42     }
43 }

```

16.1.4.7 Run App

Make sure REST service is running.

Then right click on project src/main/java ->com.xyz ->App

Run As ->Java Application

You will see in the console and in the log file (system.out.log):

```

2016-01-27 11:59:13 INFO App:26 -
=====
2016-01-27 11:59:13 INFO App:27 - @Produces("application/json") Output:
F to C Converter Output:
{"F Value":90,"C Value":32.2222137451172}
2016-01-27 11:59:14 INFO App:26 -
=====
2016-01-27 11:59:14 INFO App:27 -
→ <ctoservice><celsius>40.0</celsius><ctofoutput>@Produces("application/xml")
→ Output:
C to F Converter Output:
104.0</ctofoutput></ctoservice>

```

16.1.5 REST Service - Using Jersey2 - Hello

16.1.5.1 Create project using command line

```

1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
   → -DartifactId=hello-jersey2-service
   → -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false

```

16.1.5.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\hello-jersey2-service\pom.xml

Use properties
 Add jersey2 dependency
 Modify junit version
 Add maven-compiler-plugin, set JDK version to 1.7

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
   → xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
   → http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>hello-jersey2-service</artifactId>
6   <packaging>war</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>hello-jersey2-service Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <jersey.version>2.22.1</jersey.version>
12    <java.version>1.7</java.version>
13    <junit.version>4.12</junit.version>
14    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
15  </properties>
16  <dependencies>
17    <dependency>
18      <groupId>org.glassfish.jersey.containers</groupId>
19      <artifactId>jersey-container-servlet</artifactId>
20      <version>${jersey.version}</version>
21    </dependency>
22    <dependency>
23      <groupId>junit</groupId>
24      <artifactId>junit</artifactId>
25      <version>${junit.version}</version>
26      <scope>test</scope>
27    </dependency>
28  </dependencies>

```

```

29   <build>
30     <finalName>hello-jersey2-service</finalName>
31     <plugins>
32       <plugin>
33         <groupId>org.apache.maven.plugins</groupId>
34         <artifactId>maven-compiler-plugin</artifactId>
35         <version>${maven.compiler.plugin.version}</version>
36         <configuration>
37           <source>${java.version}</source>
38           <target>${java.version}</target>
39         </configuration>
40       </plugin>
41     </plugins>
42   </build>
43 </project>
```

16.1.5.3 Make this project an eclipse project

```

1 c:\workspaces\eclipse\my-projects\hello-jersey2-service>mvn eclipse:clean
  ↳ eclipse:eclipse -Dwtpversion=2.0
```

16.1.5.4 Import this project to eclipse

- Launch eclipse
File → Import → General → Existing Projects into Workspace
Browse to hello-jersey2-service, and import it.

16.1.5.5 Service Class

- Create java directory and add it to Build Path
src/main/java
- Create new java service class
src/main/java/com/xyz/service/HelloService.java

```

1 package com.xyz.service;
2
3 import javax.ws.rs.GET;
4 import javax.ws.rs.Path;
5 import javax.ws.rs.Produces;
6 import javax.ws.rs.core.MediaType;
7
8 /**
9  * Plain old Java Object it does not extend as class or implements an interface
10 * The class registers its methods for the HTTP GET request using the @GET
11 * annotation.
12 * Using the @Produces annotation, it defines that it can deliver several MIME
13 * types, text, XML and HTML.
14 * The browser requests per default the HTML MIME type.
15 * Sets the path to base URL + /hello
16 */
17 @Path("/hello")
18 public class HelloService {
19     // This method is called if TEXT_PLAIN is request
20     @GET
21     @Produces(MediaType.TEXT_PLAIN)
22     public String sayPlainTextHello() {
23         return "Hello Jersey";
24     }
25     // This method is called if XML is request
26     @GET
27     @Produces(MediaType.TEXT_XML)
```

```

28     public String sayXMLHello() {
29         return "<?xml version=\"1.0\"?>" + "<hello> Hello Jersey" + "</hello>";
30     }
31
32     // This method is called if HTML is request
33     @GET
34     @Produces(MediaType.TEXT_HTML)
35     public String sayHTMLHello() {
36         return "<html> " + "<title>" + "Hello Jersey" + "</title>" +
37             "<body><h1>" + "Hello Jersey" + "</body></h1>" + "</html> ";
38     }
}

```

16.1.5.6 Integrate web application - web.xml

- src/main/webapp/WEB-INF/web.xml

```

1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
5          http://java.sun.com/xml/ns/j2ee/web-app_3_0.xsd" id="WebApp_ID"
6      version="3.0">
7
8      <display-name>Hello Jersey2 Service</display-name>
9
10     <servlet>
11         <servlet-name>Jersey REST Service</servlet-name>
12         <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
13         <!-- Register resources and providers under com.xyz package. -->
14         <init-param>
15             <param-name>jersey.config.server.provider.packages</param-name>
16             <param-value>com.xyz;org.codehaus.jackson.jaxrs</param-value>
17         </init-param>
18         <load-on-startup>1</load-on-startup>
19     </servlet>
20
21     <servlet-mapping>
22         <servlet-name>Jersey REST Service</servlet-name>
23         <url-pattern>/rest/*</url-pattern>
24     </servlet-mapping>
25
26 </web-app>

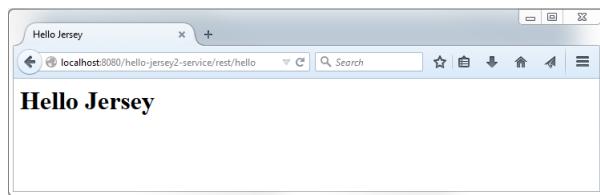
```

16.1.5.7 Build the project - command line

```
c:\workspaces\workspace\my-projects\hello-jersey2-service>mvn install
```

16.1.5.8 Run

- Add project to Tomcat Server
Go to Servers Tab
Right click Tomcat v7.0 Server at localhost [Stopped] ->Add and Remove...
Move hello-jersey2-service from Available panel (left side) to Configured panel (right side)
- Start Tomcat Server
<http://localhost:8080/hello-jersey2-service/rest/hello>



16.1.6 REST Client - Using Jersey2 - Hello

16.1.6.1 Create project using command line

```
1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↵ -DartifactId=app-hello-jersey2-client
  ↵ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

16.1.6.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\app-hello-jersey2-client\pom.xml

Use properties
 Add jersey, commons-logging, log4j, slf4j dependencies
 Modify junit version
 Add maven-compiler-plugin, set JDK version to 1.7

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.xyz</groupId>
  <artifactId>app-hello-jersey2-client</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>app-hello-jersey2-client</name>
  <url>http://maven.apache.org</url>
  <properties>
    <commons-logging.version>1.2</commons-logging.version>
    <java.version>1.7</java.version>
    <jersey.version>2.22.1</jersey.version>
    <junit.version>4.12</junit.version>
    <log4j.version>1.2.17</log4j.version>
    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
    <slf4j.version>1.7.14</slf4j.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>commons-logging</groupId>
      <artifactId>commons-logging</artifactId>
      <version>${commons-logging.version}</version>
    </dependency>
    <dependency>
      <groupId>org.glassfish.jersey.core</groupId>
      <artifactId>jersey-client</artifactId>
      <version>${jersey.version}</version>
    </dependency>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>${junit.version}</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>log4j</groupId>
```

```

38      <artifactId>log4j</artifactId>
39      <version>${log4j.version}</version>
40    </dependency>
41    <dependency>
42      <groupId>org.slf4j</groupId>
43      <artifactId>slf4j-api</artifactId>
44      <version>${slf4j.version}</version>
45    </dependency>
46    <dependency>
47      <groupId>org.slf4j</groupId>
48      <artifactId>slf4j-log4j12</artifactId>
49      <version>${slf4j.version}</version>
50    </dependency>
51  </dependencies>
52 <build>
53   <finalName>app-hello-jersey2-client</finalName>
54   <plugins>
55     <plugin>
56       <groupId>org.apache.maven.plugins</groupId>
57       <artifactId>maven-compiler-plugin</artifactId>
58       <version>${maven.compiler.plugin.version}</version>
59       <configuration>
60         <source>${java.version}</source>
61         <target>${java.version}</target>
62       </configuration>
63     </plugin>
64   </plugins>
65 </build>
66 </project>
```

16.1.6.3 Make this project an eclipse project

```

1 c:\workspaces\eclipse\my-projects\app-hello-jersey2-client>mvn eclipse:clean
  ↪ eclipse:eclipse
```

16.1.6.4 Import this project to eclipse

- Launch eclipse
File ->Import ->General ->Existing Projects into Workspace
Browse to app-hello-jersey2-client, and import it.

16.1.6.5 Configure log properties

- Create resources directory and add it to Build Path
src/main/resources
src/main/resources/log4j.properties

```

1 # Root logger option
2 log4j.rootLogger=INFO, file, stdout
3
4 # Direct log messages to a log file
5 log4j.appender.file=org.apache.log4j.RollingFileAppender
6 log4j.appender.file.File=system.out.log
7 log4j.appender.file.MaxFileSize=10MB
8 log4j.appender.file.MaxBackupIndex=10
9 log4j.appender.file.layout=org.apache.log4j.PatternLayout
10 log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
   ↪ %c{1}:%L - %m%n
11
12 # Direct log messages to stdout
13 log4j.appender.stdout=org.apache.log4j.ConsoleAppender
14 log4j.appender.stdout.Target=System.out
15 log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
16 log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
   ↪ %c{1}:%L - %m%n
```

16.1.6.6 App class

- Type in the code list below to src/main/java ->com.xyz->App.java

```

1 package com.xyz;
2
3 import java.net.URI;
4
5 import javax.ws.rs.client.Client;
6 import javax.ws.rs.client.ClientBuilder;
7 import javax.ws.rs.client.WebTarget;
8 import javax.ws.rs.core.MediaType;
9 import javax.ws.rs.core.Response;
10 import javax.ws.rs.core.UriBuilder;
11
12 import org.apache.commons.logging.Log;
13 import org.apache.commons.logging.LogFactory;
14 import org.glassfish.jersey.client.ClientConfig;
15
16 public class App {
17     private static final Log log = LogFactory.getLog(App.class);
18
19     public App() {
20     }
21
22     public void connect() {
23         ClientConfig config = new ClientConfig();
24         Client client = ClientBuilder.newClient(config);
25
26         URI baseURI =
27             UriBuilder.fromUri("http://localhost:8080/hello-jersey2-service").build();
28         WebTarget target = client.target(baseURI);
29
30         // Fluent interfaces
31         ↪ log.info(target.path("rest").path("hello").request().accept(MediaType.TEXT_PLAIN).get(Response.class)
32             .toString());
33
34         // Get plain text
35         ↪ log.info(target.path("rest").path("hello").request().accept(MediaType.TEXT_PLAIN).get(String.class));
36
37         // Get XML
38         ↪ log.info(target.path("rest").path("hello").request().accept(MediaType.TEXT_XML).get(String.class));
39
40         // The HTML
41         ↪ log.info(target.path("rest").path("hello").request().accept(MediaType.TEXT_HTML).get(String.class));
42     }
43
44     public static void main(String[] args) {
45         App client = new App();
46         client.connect();
47     }
48 }
```

16.1.6.7 Run App

Make sure REST service is running.

Then right click on project src/main/java ->com.xyz->App

Run As ->Java Application

You will see in the console and in the log file (system.out.log):

```

2016-01-28 17:43:33 INFO  App:30 -
    ↪ InboundJaxrsResponse{context=ClientResponse{method=GET,
    ↪ uri=http://localhost:8080/hello-jersey2-service/rest/hello, status=200,
    ↪ reason=OK}}
2016-01-28 17:43:33 INFO  App:33 - Hello Jersey
```

```
2016-01-28 17:43:33 INFO App:36 - <?xml version="1.0"?><hello> Hello
  ↳ Jersey</hello>
2016-01-28 17:43:33 INFO App:39 - <html> <title>Hello
  ↳ Jersey</title><body><hi>Hello Jersey</body></hi></html>
```

16.1.7 REST Service - Using Jersey2 - Employee

16.1.7.1 Create project using command line

```
1 c:\workspaces\eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=hello-jersey2-service-employee
  ↳ -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false
```

16.1.7.2 Modify pom.xml

- C:\workspaces\eclipse\my-projects\hello-jersey2-service-employee\pom.xml

Use properties
 Add jersey2, json dependency
 Modify junit version
 Add maven-compiler-plugin, set JDK version to 1.7

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>hello-jersey2-service-employee</artifactId>
6   <packaging>war</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>hello-jersey2-service-employee Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <java.version>1.7</java.version>
12    <jersey.version>2.22.1</jersey.version>
13    <json.version>20151123</json.version>
14    <junit.version>4.12</junit.version>
15    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
16  </properties>
17  <dependencies>
18    <dependency>
19      <groupId>org.glassfish.jersey.containers</groupId>
20      <artifactId>jersey-container-servlet</artifactId>
21      <version>${jersey.version}</version>
22    </dependency>
23    <dependency>
24      <groupId>org.glassfish.jersey.media</groupId>
25      <artifactId>jersey-media-json-jackson</artifactId>
26      <version>${jersey.version}</version>
27    </dependency>
28    <dependency>
29      <groupId>org.json</groupId>
30      <artifactId>json</artifactId>
31      <version>${json.version}</version>
32    </dependency>
33    <dependency>
34      <groupId>junit</groupId>
35      <artifactId>junit</artifactId>
36      <version>${junit.version}</version>
37      <scope>test</scope>
38    </dependency>
39  </dependencies>
40  <build>
41    <finalName>hello-jersey2-service-employee</finalName>
```

```

42   <plugins>
43     <plugin>
44       <groupId>org.apache.maven.plugins</groupId>
45       <artifactId>maven-compiler-plugin</artifactId>
46       <version>${maven.compiler.plugin.version}</version>
47       <configuration>
48         <source>${java.version}</source>
49         <target>${java.version}</target>
50       </configuration>
51     </plugin>
52   </plugins>
53 </build>
54 </project>

```

16.1.7.3 Make this project an eclipse project

```

1 c:\workspaces\eclipse\my-projects\hello-jersey2-service-employee>mvn
  → eclipse:clean eclipse:eclipse -Dwtpversion=2.0

```

16.1.7.4 Import this project to eclipse

- Launch eclipse
File ->Import ->General ->Existing Projects into Workspace
Browse to hello-jersey2-service-employee, and import it.

16.1.7.5 Model Class

- Create java directory and add it to Build Path
src/main/java
- Create new java model class
src/main/java/com/xyz/model/Employee.java

```

1 package com.xyz.model;
2
3 import javax.xml.bind.annotation.XmlRootElement;
4
5 @XmlRootElement
6 public class Employee {
7     int empId;
8     String lastName;
9     String firstName;
10
11    public int getEmpId() {
12        return empId;
13    }
14    public void setEmpId(int empId) {
15        this.empId = empId;
16    }
17    public String getLastname() {
18        return lastName;
19    }
20    public void setLastName(String lastName) {
21        this.lastName = lastName;
22    }
23    public String getFirstName() {
24        return firstName;
25    }
26    public void setFirstName(String firstName) {
27        this.firstName = firstName;
28    }
29
30    @Override
31    public String toString() {

```

```

32         return "Employee [empid: " + empId + " lastName = " + lastName + "
33             ↪   firstName: " + firstName + "]";
34     }

```

16.1.7.6 Service Class

- Create new java service class
src/main/java/com/xyz/service/EmployeeService.java

```

1 package com.xyz.service;
2
3 import javax.ws.rs.GET;
4 import javax.ws.rs.Path;
5 import javax.ws.rs.Produces;
6 import javax.ws.rs.core.MediaType;
7
8 import com.xyz.model.Employee;
9
10 @Path("/employee")
11 public class EmployeeService {
12     public EmployeeService() {
13     }
14
15     // This method is called if XML is request
16     @GET
17     @Produces({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
18     public Employee getXML() {
19         Employee employee = new Employee();
20         employee.setEmpId(1);
21         employee.setLastName("Smith");
22         employee.setFirstName("John");
23         return employee;
24     }
25
26     // This can be used to test the integration with the browser
27     @GET
28     @Produces({ MediaType.TEXT_XML })
29     public Employee getHTML() {
30         Employee employee = new Employee();
31         employee.setEmpId(1);
32         employee.setLastName("Smith");
33         employee.setFirstName("John");
34         return employee;
35     }
36
37     // This method is called if TEXT_PLAIN is request
38     @GET
39     @Produces(MediaType.TEXT_PLAIN)
40     public String getPlainText() {
41         Employee employee = new Employee();
42         employee.setEmpId(1);
43         employee.setLastName("Smith");
44         employee.setFirstName("John");
45         return employee.toString();
46     }
47 }

```

16.1.7.7 Integrate web application - web.xml

- src/main/webapp/WEB-INF/web.xml

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
3   ↪   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
5       ↪   http://java.sun.com/xml/ns/j2ee/web-app_3_0.xsd" id="WebApp_ID"
6       ↪   version="3.0">

```

```

4   <display-name>Hello Jersey2 Service</display-name>
5
6   <servlet>
7     <servlet-name>Jersey REST Service</servlet-name>
8     <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
9     <!-- Register resources and providers under com.xyz package. -->
10    <init-param>
11      <param-name>jersey.config.server.provider.packages</param-name>
12      <param-value>com.xyz;org.codehaus.jackson.jaxrs</param-value>
13    </init-param>
14    <load-on-startup>1</load-on-startup>
15  </servlet>
16
17
18  <servlet-mapping>
19    <servlet-name>Jersey REST Service</servlet-name>
20    <url-pattern>/rest/*</url-pattern>
21  </servlet-mapping>
22
23 </web-app>

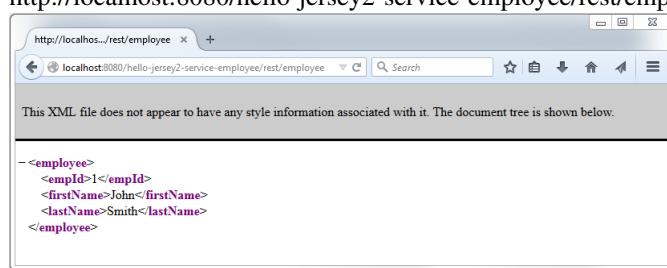
```

16.1.7.8 Build the project - command line

```
c:\workspaces\workspace\my-projects\hello-jersey2-service-employee>mvn install
```

16.1.7.9 Run

- Add project to Tomcat Server
Go to Servers Tab
Right click Tomcat v7.0 Server at localhost [Stopped] ->Add and Remove...
Move hello-jersey2-service-employee from Available panel (left side) to Configured panel (right side)
- Start Tomcat Server
<http://localhost:8080/hello-jersey2-service-employee/rest/employee>



16.1.8 REST Client - Using Jersey2 - Employee

16.1.8.1 Create project using command line

```
1 c:\workspaces\workspace\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=app-hello-jersey2-client-employee
  ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

16.1.8.2 Modify pom.xml

- C:\workspaces\workspace\my-projects\app-hello-jersey2-client-employee\pom.xml

Use properties

Add jersey, json, commons-logging, log4j, slf4j dependencies

Modify junit version

Add maven-compiler-plugin, set JDK version to 1.7

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4     http://maven.apache.org/maven-v4_0_0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <groupId>com.xyz</groupId>
7   <artifactId>app-hello-jersey2-client-employee</artifactId>
8   <packaging>jar</packaging>
9   <version>1.0-SNAPSHOT</version>
10  <name>app-hello-jersey2-client-employee</name>
11  <url>http://maven.apache.org</url>
12  <properties>
13    <commons-logging.version>1.2</commons-logging.version>
14    <java.version>1.7</java.version>
15    <jersey.version>2.22.1</jersey.version>
16    <json.version>20151123</json.version>
17    <junit.version>4.12</junit.version>
18    <log4j.version>1.2.17</log4j.version>
19    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
20    <slf4j.version>1.7.14</slf4j.version>
21  </properties>
22  <dependencies>
23    <dependency>
24      <groupId>commons-logging</groupId>
25      <artifactId>commons-logging</artifactId>
26      <version>${commons-logging.version}</version>
27    </dependency>
28    <dependency>
29      <groupId>org.glassfish.jersey.core</groupId>
30      <artifactId>jersey-client</artifactId>
31      <version>${jersey.version}</version>
32    </dependency>
33    <dependency>
34      <groupId>org.glassfish.jersey.media</groupId>
35      <artifactId>jersey-media-json-jackson</artifactId>
36      <version>${jersey.version}</version>
37    </dependency>
38    <dependency>
39      <groupId>org.json</groupId>
40      <artifactId>json</artifactId>
41      <version>${json.version}</version>
42    </dependency>
43    <dependency>
44      <groupId>junit</groupId>
45      <artifactId>junit</artifactId>
46      <version>${junit.version}</version>
47      <scope>test</scope>
48    </dependency>
49    <dependency>
50      <groupId>log4j</groupId>
51      <artifactId>log4j</artifactId>
52      <version>${log4j.version}</version>
53    </dependency>
54    <dependency>
55      <groupId>org.slf4j</groupId>
56      <artifactId>slf4j-api</artifactId>
57      <version>${slf4j.version}</version>
58    </dependency>
59    <dependency>
60      <groupId>org.slf4j</groupId>
61      <artifactId>slf4j-log4j12</artifactId>
62      <version>${slf4j.version}</version>
63    </dependency>
64  </dependencies>
65  <build>
66    <finalName>app-hello-jersey2-client-employee</finalName>

```

```

65 <plugins>
66   <plugin>
67     <groupId>org.apache.maven.plugins</groupId>
68     <artifactId>maven-compiler-plugin</artifactId>
69     <version>${maven.compiler.plugin.version}</version>
70     <configuration>
71       <source>${java.version}</source>
72       <target>${java.version}</target>
73     </configuration>
74   </plugin>
75 </plugins>
76 </build>
77 </project>
```

16.1.8.3 Make this project an eclipse project

```

1 c:\workspaces\eclipse\my-projects\app-hello-jersey2-client-employee>mvn
  ↳ eclipse:clean eclipse:eclipse
```

16.1.8.4 Import this project to eclipse

- Launch eclipse
File ->Import ->General ->Existing Projects into Workspace
Browse to app-hello-jersey2-client-employee, and import it.

16.1.8.5 Configure log properties

- Create resources directory and add it to Build Path
src/main/resources
src/main/resources/log4j.properties

```

1 # Root logger option
2 log4j.rootLogger=INFO, file, stdout
3
4 # Direct log messages to a log file
5 log4j.appender.file=org.apache.log4j.RollingFileAppender
6 log4j.appender.file.File=system.out.log
7 log4j.appender.file.MaxFileSize=10MB
8 log4j.appender.file.MaxBackupIndex=10
9 log4j.appender.file.layout=org.apache.log4j.PatternLayout
10 log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
    ↳ %c{1}:%L - %m%n
11
12 # Direct log messages to stdout
13 log4j.appender.stdout=org.apache.log4j.ConsoleAppender
14 log4j.appender.stdout.Target=System.out
15 log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
16 log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
    ↳ %c{1}:%L - %m%n
```

16.1.8.6 App class

- Type in the code list below to src/main/java ->com.xyz ->EmployeeApp.java

```

1 package com.xyz;
2
3 import java.net.URI;
4
5 import javax.ws.rs.client.Client;
6 import javax.ws.rs.client.ClientBuilder;
7 import javax.ws.rs.client.WebTarget;
8 import javax.ws.rs.core.MediaType;
```

```

9 import javax.ws.rs.core.Response;
10 import javax.ws.rs.core.UriBuilder;
11
12 import org.apache.commons.logging.Log;
13 import org.apache.commons.logging.LogFactory;
14 import org.glassfish.jersey.client.ClientConfig;
15
16 public class EmployeeApp {
17     private static final Log log = LogFactory.getLog(EmployeeApp.class);
18
19     public EmployeeApp() {
20     }
21
22     public void connect() {
23         ClientConfig config = new ClientConfig();
24         Client client = ClientBuilder.newClient(config);
25
26         URI baseURI =
27             UriBuilder.fromUri("http://localhost:8080/hello-jersey2-service-employee").build();
28         WebTarget target = client.target(baseURI);
29
30         // Fluent interfaces
31         log.info(target.path("rest").path("employee").request().accept(MediaType.TEXT_PLAIN).get(Response.class)
32             .toString());
33
34         // Get Text for application
35         log.info(target.path("rest").path("employee").request().accept(MediaType.TEXT_PLAIN).get(String.class));
36
37         // Get JSON for application
38         log.info(target.path("rest").path("employee").request().accept(MediaType.APPLICATION_JSON).get(String.class));
39
40         // Get XML
41         log.info(target.path("rest").path("employee").request().accept(MediaType.TEXT_XML).get(String.class));
42
43         // Get XML for application
44         log.info(target.path("rest").path("employee").request().accept(MediaType.APPLICATION_XML).get(String.class));
45     }
46
47     public static void main(String[] args) {
48         EmployeeApp client = new EmployeeApp();
49         client.connect();
50     }

```

16.1.8.7 Run App

Make sure REST service is running.

Then right click on project src/main/java ->com.xyz ->EmployeeApp

Run As ->Java Application

You will see in the console and in the log file (system.out.log):

```

2016-02-05 14:02:54 INFO EmployeeApp:30 -
    InboundJaxrsResponse{context=ClientResponse{method=GET,
    uri=http://localhost:8080/hello-jersey2-service-employee/rest/employee,
    status=200, reason=OK}}
2016-02-05 14:02:54 INFO EmployeeApp:34 - Employee [empid: 1 lastName = Smith
    firstName: John]
2016-02-05 14:02:54 INFO EmployeeApp:37 -
    {"empId":1,"lastName":"Smith","firstName":"John"}
2016-02-05 14:02:54 INFO EmployeeApp:40 - <?xml version="1.0" encoding="UTF-8"
    standalone="yes"?><employee><empId>1</empId><firstName>John</firstName><lastName>Smith</lastName></employee>
2016-02-05 14:02:54 INFO EmployeeApp:43 - <?xml version="1.0" encoding="UTF-8"
    standalone="yes"?><employee><empId>1</empId><firstName>John</firstName><lastName>Smith</lastName></employee>

```

16.1.9 REST Service - Using Jersey2 - EmployeeList

We are going to add a new service, still using hello-jersey2-service-employee project.

16.1.9.1 Service Classes

- Create new java service class

src/main/java/com/xyz/service/EmployeeListService.java

```

1 package com.xyz.service;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import javax.ws.rs.GET;
7 import javax.ws.rs.Path;
8 import javax.ws.rs.Produces;
9 import javax.ws.rs.core.MediaType;
10
11 import com.xyz.model.Employee;
12
13 @Path("/employeelist")
14 public class EmployeeListService {
15     public EmployeeListService() {
16     }
17
18     // This method is called if XML is request
19     @GET
20     @Produces({ MediaType.APPLICATION_XML, MediaType.APPLICATION_JSON })
21     public List<Employee> getXML() {
22         List<Employee> employeeList = new ArrayList<Employee>();
23
24         Employee employee1 = new Employee();
25         employee1.setEmpId(1);
26         employee1.setLastName("Smith");
27         employee1.setFirstName("John");
28         employeeList.add(employee1);
29
30         Employee employee2 = new Employee();
31         employee2.setEmpId(2);
32         employee2.setLastName("Doe");
33         employee2.setFirstName("Jane");
34         employeeList.add(employee2);
35
36         return employeeList;
37     }
38
39     // This can be used to test the integration with the browser
40     @GET
41     @Produces({ MediaType.TEXT_XML })
42     public List<Employee> getHTML() {
43         List<Employee> employeeList = new ArrayList<Employee>();
44
45         Employee employee1 = new Employee();
46         employee1.setEmpId(1);
47         employee1.setLastName("Smith");
48         employee1.setFirstName("John");
49         employeeList.add(employee1);
50
51         Employee employee2 = new Employee();
52         employee2.setEmpId(1);
53         employee2.setLastName("Doe");
54         employee2.setFirstName("Jane");
55         employeeList.add(employee2);
56
57         return employeeList;
58     }
59
60     // This method is called if TEXT_PLAIN is request
61     @GET
62     @Produces(MediaType.TEXT_PLAIN)

```

```

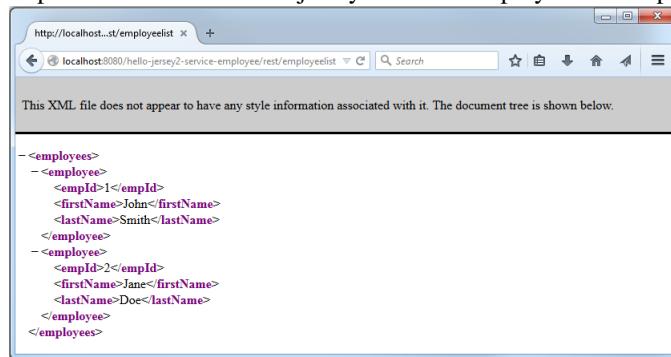
63  public String getPlainText() {
64      List<Employee> employeeList = new ArrayList<Employee>();
65
66      Employee employee1 = new Employee();
67      employee1.setEmpId(1);
68      employee1.setLastName("Smith");
69      employee1.setFirstName("John");
70      employeeList.add(employee1);
71
72      Employee employee2 = new Employee();
73      employee2.setEmpId(1);
74      employee2.setLastName("Doe");
75      employee2.setFirstName("Jane");
76      employeeList.add(employee2);
77
78      return employeeList.toString();
79  }
80 }
```

16.1.9.2 Build the project - command line

```
c:\workspaces\eclipse\my-projects\hello-jersey2-service-employee>mvn install
```

16.1.9.3 Run

- Start Tomcat Server
<http://localhost:8080/hello-jersey2-service-employee/rest/employeelist>



16.1.10 REST Client - Using Jersey2 - EmployeeList

We are going to add a new client, still using app-hello-jersey2-client-employee project.

16.1.10.1 Client class

- Type in the code list below to src/main/java ->com.xyz ->EmployeeListApp.java

```

1 package com.xyz;
2
3 import java.net.URI;
4
5 import javax.ws.rs.client.Client;
6 import javax.ws.rs.client.ClientBuilder;
7 import javax.ws.rs.client.WebTarget;
8 import javax.ws.rs.core.MediaType;
9 import javax.ws.rs.core.Response;
10 import javax.ws.rs.core.UriBuilder;
11
```

```

12 import org.apache.commons.logging.Log;
13 import org.apache.commons.logging.LogFactory;
14 import org.glassfish.jersey.client.ClientConfig;
15
16 public class EmployeeListApp {
17     private static final Log log = LogFactory.getLog(EmployeeListApp.class);
18
19     public EmployeeListApp() {
20     }
21
22     public void connect() {
23         ClientConfig config = new ClientConfig();
24         Client client = ClientBuilder.newClient(config);
25
26         URI baseURI =
27             UriBuilder.fromUri("http://localhost:8080/hello-jersey2-service-employee").build();
28         WebTarget target = client.target(baseURI);
29
30         // Fluent interfaces
31         ↪ log.info(target.path("rest").path("employeelist").request().accept(MediaType.TEXT_PLAIN).get(Response.class));
32
33         // Get Text for application
34         ↪ log.info(target.path("rest").path("employeelist").request().accept(MediaType.TEXT_PLAIN).get(String.class));
35
36         // Get JSON for application
37         ↪ log.info(target.path("rest").path("employeelist").request().accept(MediaType.APPLICATION_JSON).get(String.class));
38
39         // Get XML
40         ↪ log.info(target.path("rest").path("employeelist").request().accept(MediaType.TEXT_XML).get(String.class));
41
42         // Get XML for application
43         log.info(
44             ↪ target.path("rest").path("employeelist").request().accept(MediaType.APPLICATION_XML).get(String.class));
45
46     }
47
48     public static void main(String[] args) {
49         EmployeeListApp client = new EmployeeListApp();
50         client.connect();
51     }
52 }
```

16.1.10.2 Run App

Make sure REST service is running.

Then right click on project src/main/java ->com.xyz ->EmployeeListApp

Run As ->Java Application

You will see in the console and in the log file (system.out.log):

```

2016-02-05 17:19:02 INFO EmployeeListApp:30 -
    ↪ InboundJaxrsResponse{context=ClientResponse{method=GET,
    ↪ uri=http://localhost:8080/hello-jersey2-service-employee/rest/employeelist,
    ↪ status=200, reason=OK}}
2016-02-05 17:19:02 INFO EmployeeListApp:34 - [Employee [empid: 1 lastName =
    ↪ Smith firstName: John], Employee [empid: 1 lastName = Doe firstName: Jane]]
2016-02-05 17:19:02 INFO EmployeeListApp:37 -
    ↪ [{"empId":1,"lastName":"Smith","firstName":"John"}, {"empId":2,"lastName":"Doe","firstName":"Jane"}]
2016-02-05 17:19:02 INFO EmployeeListApp:41 - <?xml version="1.0"
    ↪ encoding="UTF-8"
    ↪ standalone="yes"?><employees><employee><empId>1</empId><firstName>John</firstName><lastName>Smith</lastName></employee>
2016-02-05 17:19:02 INFO EmployeeListApp:44 - <?xml version="1.0"
    ↪ encoding="UTF-8"
    ↪ standalone="yes"?><employees><employee><empId>1</empId><firstName>John</firstName><lastName>Smith</lastName></employee>
```

16.1.11 REST Service - Using Jersey2 - Employee Oper

16.1.11.1 Create project using command line

```
1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=hello-jersey2-service-employee-oper
  ↳ -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false
```

16.1.11.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\hello-jersey2-service-employee\pom.xml

Use properties

Add jersey2, json, commons-logging, log4j, slf4j dependency

Modify junit version

Add maven-compiler-plugin, set JDK version to 1.7

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  ↳ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>hello-jersey2-service-employee-oper</artifactId>
6   <packaging>war</packaging>
7   <version>1.0-SNAPSHOT</version>
8   <name>hello-jersey2-service-employee-oper Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10  <properties>
11    <commons-logging.version>1.2</commons-logging.version>
12    <java.version>1.7</java.version>
13    <jersey.version>2.22.1</jersey.version>
14    <json.version>20151123</json.version>
15    <junit.version>4.12</junit.version>
16    <log4j.version>1.2.17</log4j.version>
17    <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
18    <slf4j.version>1.7.14</slf4j.version>
19  </properties>
20  <dependencies>
21    <dependency>
22      <groupId>commons-logging</groupId>
23      <artifactId>commons-logging</artifactId>
24      <version>${commons-logging.version}</version>
25    </dependency>
26    <dependency>
27      <groupId>org.glassfish.jersey.containers</groupId>
28      <artifactId>jersey-container-servlet</artifactId>
29      <version>${jersey.version}</version>
30    </dependency>
31    <dependency>
32      <groupId>org.glassfish.jersey.media</groupId>
33      <artifactId>jersey-media-json-jackson</artifactId>
34      <version>${jersey.version}</version>
35    </dependency>
36    <dependency>
37      <groupId>org.json</groupId>
38      <artifactId>json</artifactId>
39      <version>${json.version}</version>
40    </dependency>
41    <dependency>
42      <groupId>junit</groupId>
43      <artifactId>junit</artifactId>
44      <version>${junit.version}</version>
45      <scope>test</scope>
46    </dependency>
47    <dependency>
48      <groupId>log4j</groupId>
```

```

49      <artifactId>log4j</artifactId>
50      <version>${log4j.version}</version>
51  </dependency>
52  <dependency>
53    <groupId>org.slf4j</groupId>
54    <artifactId>slf4j-api</artifactId>
55    <version>${slf4j.version}</version>
56  </dependency>
57  <dependency>
58    <groupId>org.slf4j</groupId>
59    <artifactId>slf4j-log4j12</artifactId>
60    <version>${slf4j.version}</version>
61  </dependency>
62 </dependencies>
63 <build>
64   <finalName>hello-jersey2-service-employee-oper</finalName>
65   <plugins>
66     <plugin>
67       <groupId>org.apache.maven.plugins</groupId>
68       <artifactId>maven-compiler-plugin</artifactId>
69       <version>${maven.compiler.plugin.version}</version>
70       <configuration>
71         <source>${java.version}</source>
72         <target>${java.version}</target>
73       </configuration>
74     </plugin>
75   </plugins>
76 </build>
77 </project>
```

16.1.11.3 Make this project an eclipse project

```

1 c:\workspaces\workspace\my-projects\hello-jersey2-service-employee-oper>mvn
  ↳ eclipse:clean eclipse:eclipse -Dwtpversion=2.0
```

16.1.11.4 Import this project to eclipse

- Launch eclipse
File → Import → General → Existing Projects into Workspace
Browse to hello-jersey2-service-employee-oper, and import it.

16.1.11.5 Model Class

- Create java directory and add it to Build Path
src/main/java
- Create new java model class
src/main/java/com/xyz/model/Employee.java

```

1 package com.xyz.model;
2
3 import java.sql.Timestamp;
4
5 import javax.xml.bind.annotation.XmlRootElement;
6
7 @XmlRootElement
8 public class Employee {
9     int empId;
10    String lastName;
11    String firstName;
12    String createdBy;
13    Timestamp createdDate;
14
15    public int getEmpId() {
```

```

16         return empId;
17     }
18     public void setEmpId(int empId) {
19         this.empId = empId;
20     }
21     public String getLastName() {
22         return lastName;
23     }
24     public void setLastName(String lastName) {
25         this.lastName = lastName;
26     }
27     public String getFirstName() {
28         return firstName;
29     }
30     public void setFirstName(String firstName) {
31         this.firstName = firstName;
32     }
33     public String getCreatedBy() {
34         return createdBy;
35     }
36     public void setCreatedBy(String createdBy) {
37         this.createdBy = createdBy;
38     }
39     public Timestamp getCreatedDate() {
40         return createdDate;
41     }
42     public void setCreatedDate(Timestamp createdDate) {
43         this.createdDate = createdDate;
44     }
45
46     @Override
47     public String toString() {
48         return "Employee [empid: " + empId + " lastName = " + lastName + "
49             ↑   firstName: " + firstName + " createdBy: " + createdBy +
50             ↑   createdDate = " + createdDate + "]";
51     }
52 }
```

16.1.11.6 DAO Classes

- Create new java DAO classes

src/main/java/com/xyz/dao/EmployeeDao.java

```

1 package com.xyz.dao;
2
3 import java.util.List;
4
5 import com.xyz.model.Employee;
6
7 public interface EmployeeDao {
8     Employee get(final Integer empId);
9     List<Employee> list();
10    Employee save(final Employee employee);
11    void delete(final Integer empId);
12    int count();
13 }
```

src/main/java/com/xyz/dao/EmployeeDaoImpl.java

```

1 package com.xyz.dao;
2
3 import java.sql.Timestamp;
4 import java.util.ArrayList;
5 import java.util.Date;
6 import java.util.HashMap;
7 import java.util.List;
8 import java.util.Map;
9
10 import com.xyz.model.Employee;
```

```

11  public enum EmployeeDaoImpl implements EmployeeDao {
12      INSTANCE;
13
14      private final Map<Integer, Employee> employeeProvider = new HashMap<>();
15
16      private EmployeeDaoImpl() {
17          init();
18      }
19
20
21      @Override
22      public Employee get(final Integer empId) {
23          return employeeProvider.get(empId);
24      }
25
26      @Override
27      public List<Employee> list() {
28          return new ArrayList<Employee>(employeeProvider.values());
29      }
30
31      @Override
32      public Employee save(final Employee employee) {
33          employee.setCreatedBy("system");
34          employee.setCreatedDate(new Timestamp(new Date().getTime()));
35          employeeProvider.put(employee.getEmpId(), employee);
36          return employee;
37      }
38
39      @Override
40      public void delete(final Integer empId) {
41          if (employeeProvider.containsKey(empId)) {
42              employeeProvider.remove(empId);
43          }
44      }
45
46      @Override
47      public int count() {
48          return employeeProvider.size();
49      }
50
51      private void init() {
52          Employee employee1 = new Employee();
53          employee1.setEmpId(1);
54          employee1.setLastName("Smith");
55          employee1.setFirstName("John");
56          employee1.setCreatedBy("system");
57          employee1.setCreatedDate(new Timestamp(new Date().getTime()));
58          employeeProvider.put(1, employee1);
59
60          Employee employee2 = new Employee();
61          employee2.setEmpId(2);
62          employee2.setLastName("Doe");
63          employee2.setFirstName("Jane");
64          employee2.setCreatedBy("system");
65          employee2.setCreatedDate(new Timestamp(new Date().getTime()));
66          employeeProvider.put(2, employee2);
67      }
68  }

```

16.1.11.7 Service Class

- Create new java service class
src/main/java/com/xyz/service/EmployeeOperService.java

```

1  package com.xyz.service;
2
3  import java.util.List;
4
5  import javax.ws.rs.Consumes;
6  import javax.ws.rs.DELETE;

```

```

7 import javax.ws.rs.GET;
8 import javax.ws.rs.POST;
9 import javax.ws.rs.Path;
10 import javax.ws.rs.PathParam;
11 import javax.ws.rs.Produces;
12 import javax.ws.rs.core.GenericEntity;
13 import javax.ws.rs.core.MediaType;
14 import javax.ws.rs.core.Response;
15
16 import org.apache.commons.logging.Log;
17 import org.apache.commons.logging.LogFactory;
18
19 import com.xyz.dao.EmployeeDao;
20 import com.xyz.dao.EmployeeDaoImpl;
21 import com.xyz.model.Employee;
22
23 @Path("/employee-oper")
24 public class EmployeeOperService {
25     private static final Log log = LogFactory.getLog(EmployeeOperService.class);
26     private final EmployeeDao employeeRepository = EmployeeDaoImpl.INSTANCE;
27
28     public EmployeeOperService() {
29     }
30
31     @GET
32     @Path("/{empId}")
33     @Produces(MediaType.APPLICATION_JSON)
34     public Response get(@PathParam("empId") Integer empId) {
35         log.info("@GET @Path: MediaType.APPLICATION_JSON");
36         Employee employee = employeeRepository.get(empId);
37         return Response.ok(employee).build();
38     }
39
40     @GET
41     @Produces(MediaType.APPLICATION_JSON)
42     public Response list() {
43         log.info("@GET: MediaType.APPLICATION_JSON");
44         List<Employee> employeeList = employeeRepository.list();
45         GenericEntity<List<Employee>> employeeWrapper = new
46             GenericEntity<List<Employee>>(employeeList) {
47         };
48         return Response.ok(employeeWrapper).build();
49     }
50
51     @GET
52     @Produces(MediaType.APPLICATION_XML)
53     public List<Employee> listXML() {
54         log.info("@GET: MediaType.APPLICATION_XML, MediaType.TEXT_XML");
55         List<Employee> employeeList = employeeRepository.list();
56         for (Employee employee : employeeList) {
57             log.info("empId = " + employee.getEmpId());
58             log.info("lastName = " + employee.getLastName());
59             log.info("firstName = " + employee.getFirstName());
60             log.info("createdBy = " + employee.getCreatedBy());
61             log.info("createdDate = " + employee.getCreatedDate());
62         }
63         return employeeList;
64     }
65
66     @GET
67     @Produces(MediaType.TEXT_XML)
68     public List<Employee> listTextXML() {
69         log.info("@GET: MediaType.APPLICATION_XML, MediaType.TEXT_XML");
70         List<Employee> employeeList = employeeRepository.list();
71         for (Employee employee : employeeList) {
72             log.info("empId = " + employee.getEmpId());
73             log.info("lastName = " + employee.getLastName());
74             log.info("firstName = " + employee.getFirstName());
75             log.info("createdBy = " + employee.getCreatedBy());
76             log.info("createdDate = " + employee.getCreatedDate());
77         }
78         return employeeList;
    }

```

```

79
80     @GET
81     @Produces(MediaType.TEXT_PLAIN)
82     public String listText() {
83         log.info("@GET: MediaType.TEXT_PLAIN");
84         List<Employee> employeeList = employeeRepository.list();
85         for (Employee employee : employeeList) {
86             log.info("empId = " + employee.getEmpId());
87             log.info("lastName = " + employee.getLastName());
88             log.info("firstName = " + employee.getFirstName());
89             log.info("createdBy = " + employee.getCreatedBy());
90             log.info("createdAt = " + employee.getCreatedDate());
91         }
92         return employeeList.toString();
93     }
94
95     @POST
96     @Consumes(MediaType.APPLICATION_JSON)
97     @Produces(MediaType.APPLICATION_JSON)
98     public Response save(final Employee employee) {
99         log.info("@POST: save");
100        Employee employeePersisted = employeeRepository.save(employee);
101        return Response.ok(employeePersisted).build();
102    }
103
104    @DELETE
105    @Path("/{empId}")
106    public Response delete(@PathParam("empId") Integer empId) {
107        log.info("@DELETE: delete");
108        employeeRepository.delete(empId);
109        return Response.ok().build();
110    }
111
112    @GET
113    @Path("count")
114    @Produces(MediaType.TEXT_PLAIN)
115    public String count() {
116        log.info("@Path(\"count\")");
117        int count = employeeRepository.count();
118        return String.valueOf(count);
119    }
120}

```

16.1.11.8 Integrate web application - web.xml

- src/main/webapp/WEB-INF/web.xml

```

1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
3           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4           xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
5                           http://java.sun.com/xml/ns/j2ee/web-app_3_0.xsd" id="WebApp_ID"
6           version="3.0">
7
8      <display-name>Hello Jersey2 Service</display-name>
9
10     <servlet>
11         <servlet-name>Jersey REST Service</servlet-name>
12         <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
13         <!-- Register resources and providers under com.xyz package. -->
14         <init-param>
15             <param-name>jersey.config.server.provider.packages</param-name>
16             <param-value>com.xyz;org.codehaus.jackson.jaxrs</param-value>
17         </init-param>
18         <load-on-startup>1</load-on-startup>
19     </servlet>
20
21     <servlet-mapping>
22         <servlet-name>Jersey REST Service</servlet-name>
23         <url-pattern>/rest/*</url-pattern>
24     </servlet-mapping>

```

```

22
23 </web-app>

```

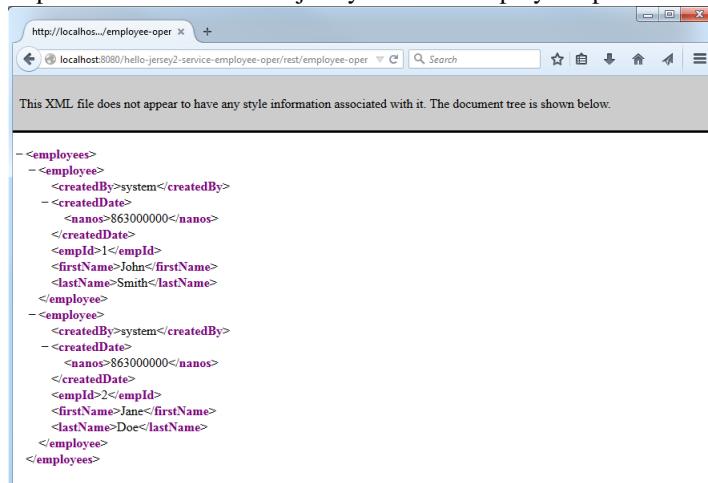
16.1.11.9 Build the project - command line

```
c:\workspaces\eclipse\my-projects\hello-jersey2-service-employee-oper>mvn
  ↵ install
```

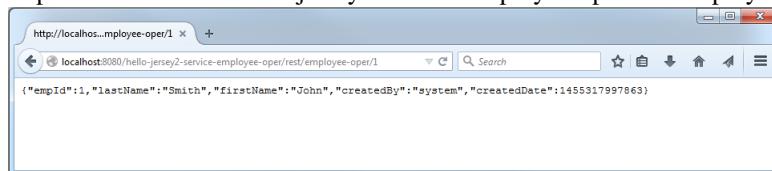
16.1.11.10 Run

- Add project to Tomcat Server
Go to Servers Tab
Right click Tomcat v7.0 Server at localhost [Stopped] → Add and Remove...
Move hello-jersey2-service-employee from Available panel (left side) to Configured panel (right side)

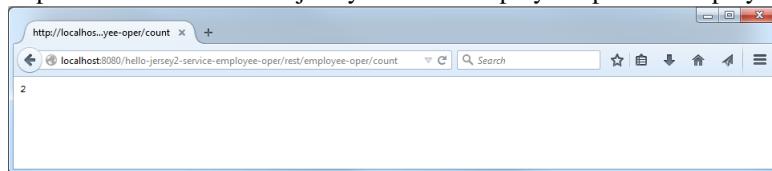
- Start Tomcat Server
<http://localhost:8080/hello-jersey2-service-employee-oper/rest/employee-oper>



<http://localhost:8080/hello-jersey2-service-employee-oper/rest/employee-oper/1>



<http://localhost:8080/hello-jersey2-service-employee-oper/rest/employee-oper/count>



16.1.12 REST Client - Using Jersey2 - Employee Oper

16.1.12.1 Create project using command line

```

1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=app-hello-jersey2-client-employee-oper
  ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

```

16.1.12.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\app-hello-jersey2-client-employee-oper\pom.xml

Use properties

Add jersey, json, commons-logging, log4j, slf4j dependencies

Modify junit version

Add maven-compiler-plugin, set JDK version to 1.7

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
  ↳ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    ↳ http://maven.apache.org/maven-v4_0_0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>com.xyz</groupId>
<artifactId>app-hello-jersey2-client-employee-oper</artifactId>
<packaging>jar</packaging>
<version>1.0-SNAPSHOT</version>
<name>app-hello-jersey2-client-employee-oper</name>
<url>http://maven.apache.org</url>
<properties>
  <commons-logging.version>1.2</commons-logging.version>
  <java.version>1.7</java.version>
  <jersey.version>2.22.1</jersey.version>
  <json.version>20151123</json.version>
  <junit.version>4.12</junit.version>
  <log4j.version>1.2.17</log4j.version>
  <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
  <slf4j.version>1.7.14</slf4j.version>
</properties>
<dependencies>
  <dependency>
    <groupId>commons-logging</groupId>
    <artifactId>commons-logging</artifactId>
    <version>${commons-logging.version}</version>
  </dependency>
  <dependency>
    <groupId>org.glassfish.jersey.core</groupId>
    <artifactId>jersey-client</artifactId>
    <version>${jersey.version}</version>
  </dependency>
  <dependency>
    <groupId>org.glassfish.jersey.media</groupId>
    <artifactId>jersey-media-json-jackson</artifactId>
    <version>${jersey.version}</version>
  </dependency>
  <dependency>
    <groupId>org.json</groupId>
    <artifactId>json</artifactId>
    <version>${json.version}</version>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>${log4j.version}</version>
  </dependency>
</dependencies>

```

```

53      <groupId>org.slf4j</groupId>
54      <artifactId>slf4j-api</artifactId>
55      <version>${slf4j.version}</version>
56    </dependency>
57    <dependency>
58      <groupId>org.slf4j</groupId>
59      <artifactId>slf4j-log4j12</artifactId>
60      <version>${slf4j.version}</version>
61    </dependency>
62  </dependencies>
63  <build>
64    <finalName>app-hello-jersey2-client-employee-oper</finalName>
65    <plugins>
66      <plugin>
67        <groupId>org.apache.maven.plugins</groupId>
68        <artifactId>maven-compiler-plugin</artifactId>
69        <version>${maven.compiler.plugin.version}</version>
70        <configuration>
71          <source>${java.version}</source>
72          <target>${java.version}</target>
73        </configuration>
74      </plugin>
75    </plugins>
76  </build>
77 </project>

```

16.1.12.3 Make this project an eclipse project

```

1 c:\workspaces\eclipse\my-projects\app-hello-jersey2-client-employee-oper>mvn
  ↳ eclipse:clean eclipse:eclipse

```

16.1.12.4 Import this project to eclipse

- Launch eclipse
File ->Import ->General ->Existing Projects into Workspace
Browse to app-hello-jersey2-client-employee-oper, and import it.

16.1.12.5 Configure log properties

- Create resources directory and add it to Build Path
src/main/resources
src/main/resources/log4j.properties

```

1 # Root logger option
2 log4j.rootLogger=INFO, file, stdout
3
4 # Direct log messages to a log file
5 log4j.appender.file=org.apache.log4j.RollingFileAppender
6 log4j.appender.file.File=system.out.log
7 log4j.appender.file.MaxFileSize=10MB
8 log4j.appender.file.MaxBackupIndex=10
9 log4j.appender.file.layout=org.apache.log4j.PatternLayout
10 log4j.appender.file.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
   ↳ %c{1}:%L - %m%n
11
12 # Direct log messages to stdout
13 log4j.appender.stdout=org.apache.log4j.ConsoleAppender
14 log4j.appender.stdout.Target=System.out
15 log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
16 log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p
   ↳ %c{1}:%L - %m%n

```

16.1.12.6 Model Class

- Create new java model class

src/main/java/com/xyz/model/Employee.java

```

1 package com.xyz.model;
2
3 import java.sql.Timestamp;
4
5 import javax.xml.bind.annotation.XmlRootElement;
6
7 @XmlRootElement
8 public class Employee {
9     int empId;
10    String lastName;
11    String firstName;
12    String createdBy;
13    Timestamp createdDate;
14
15    public int getEmpId() {
16        return empId;
17    }
18    public void setEmpId(int empId) {
19        this.empId = empId;
20    }
21    public String getLastname() {
22        return lastName;
23    }
24    public void setLastname(String lastName) {
25        this.lastName = lastName;
26    }
27    public String getFirstname() {
28        return firstName;
29    }
30    public void setFirstname(String firstName) {
31        this.firstName = firstName;
32    }
33    public String getCreatedBy() {
34        return createdBy;
35    }
36    public void setCreatedBy(String createdBy) {
37        this.createdBy = createdBy;
38    }
39    public Timestamp getCreatedDate() {
40        return createdDate;
41    }
42    public void setCreatedDate(Timestamp createdDate) {
43        this.createdDate = createdDate;
44    }
45
46    @Override
47    public String toString() {
48        return "Employee [empid: " + empId + " lastName = " + lastName + "
49           ↪ firstName: " + firstName + " createdBy: " + createdBy + "
50           ↪ createdDate = " + createdDate + "]";
51    }
52 }
```

16.1.12.7 App class

- Type in the code list below to src/main/java ->com.xyz ->EmployeeOperApp.java

```

1 package com.xyz;
2
3 import java.net.URI;
4 import java.util.List;
5
6 import javax.ws.rs.client.Client;
7 import javax.ws.rs.client.ClientBuilder;
8 import javax.ws.rs.client.Entity;
9 import javax.ws.rs.client.WebTarget;
10 import javax.ws.rs.core.GenericType;
```

```

11 import javax.ws.rs.core.MediaType;
12 import javax.ws.rs.core.Response;
13 import javax.ws.rs.core.UriBuilder;
14
15 import org.apache.commons.logging.Log;
16 import org.apache.commons.logging.LogFactory;
17 import org.glassfish.jersey.client.ClientConfig;
18
19 import com.xyz.model.Employee;
20
21 public class EmployeeOperApp {
22     private static final Log log = LogFactory.getLog(EmployeeOperApp.class);
23
24     public EmployeeOperApp() {
25     }
26
27     public void connect() {
28         ClientConfig config = new ClientConfig();
29         Client client = ClientBuilder.newClient(config);
30
31         URI baseURI =
32             UriBuilder.fromUri("http://localhost:8080/hello-jersey2-service-employee-oper").build();
33         WebTarget target = client.target(baseURI);
34
35         // Regular checking ...
36
37         // Fluent interfaces
38
39         // Get Text for application
40
41         // Get JSON for application
42
43         // Get XML
44
45         // Get XML for application
46
47         // Operation ...
48
49         // 1. Save
50         // Save John Doe
51         Employee employee_save = new Employee();
52         employee_save.setEmpId(3);
53         employee_save.setLastName("Doe");
54         employee_save.setFirstName("John");
55
56         Employee employeePersisted =
57             target.path("rest").path("employee-oper").request()
58             .post(Entity.entity(employee_save, MediaType.APPLICATION_JSON),
59                   Employee.class);
60
61         // Save Charlie Brown
62         employee_save = new Employee();
63         employee_save.setEmpId(1001);
64         employee_save.setLastName("Borwn");
65         employee_save.setFirstName("Charlie");
66
67         employeePersisted = target.path("rest").path("employee-oper").request()
68             .post(Entity.entity(employee_save, MediaType.APPLICATION_JSON),
69                   Employee.class);
70
71
72
73
74

```

```

75     int empId = employeePersisted.getEmpId();
76     log.info("empId = " + empId);
77     // Get JSON for application
78
79     ↵   log.info(target.path("rest").path("employee-oper").request().accept(MediaType.APPLICATION_JSON)
80             .get(String.class));
81
82     // 2. Fetch employee by empId
83     Employee employee_fetch =
84         ↵   target.path("rest").path("employee-oper").path("/{empId}")
85             .resolveTemplate("empId", empId).request().get(Employee.class);
86     log.info("Emp ID = " + employee_fetch.getEmpId());
87     log.info("Last Name = " + employee_fetch.getLastname());
88     log.info("First Name = " + employee_fetch.getFirstName());
89     log.info("Created By = " + employee_fetch.getCreatedBy());
90     log.info("Created Date = " + employee_fetch.getCreatedDate());
91
92     // 3. Fetch all employees
93     GenericType<List<Employee>> employeeType = new
94         ↵   GenericType<List<Employee>>() {
95     }; // generic type to wrap a generic list of books
96     List<Employee> employeeList =
97         ↵   target.path("rest").path("employee-oper").request().get(employeeType);
98     log.info("employee count = " + employeeList.size());
99     for (Employee employee : employeeList) {
100         ↵   log.info(employee);
101
102     // 4. Delete a employee
103
104         ↵   target.path("rest").path("employee-oper").path("/{empId}").resolveTemplate("empId",
105             ↵   empId).request().delete();
106     List<Employee> employeeList_final =
107         ↵   target.path("rest").path("employee-oper").request().get(employeeType);
108     log.info("employee count = " + employeeList_final.size());
109
110         ↵   log.info(target.path("rest").path("employee-oper").request().accept(MediaType.APPLICATION_JSON)
111             .get(String.class));
112     }
113
114     public static void main(String[] args) {
115         EmployeeOperApp client = new EmployeeOperApp();
116         client.connect();
117     }
118 }
```

16.1.12.8 Run App

Make sure REST service is running.

Then right click on project src/main/java ->com.xyz ->EmployeeOperApp

Run As ->Java Application

You will see in the console and in the log file (system.out.log):

```

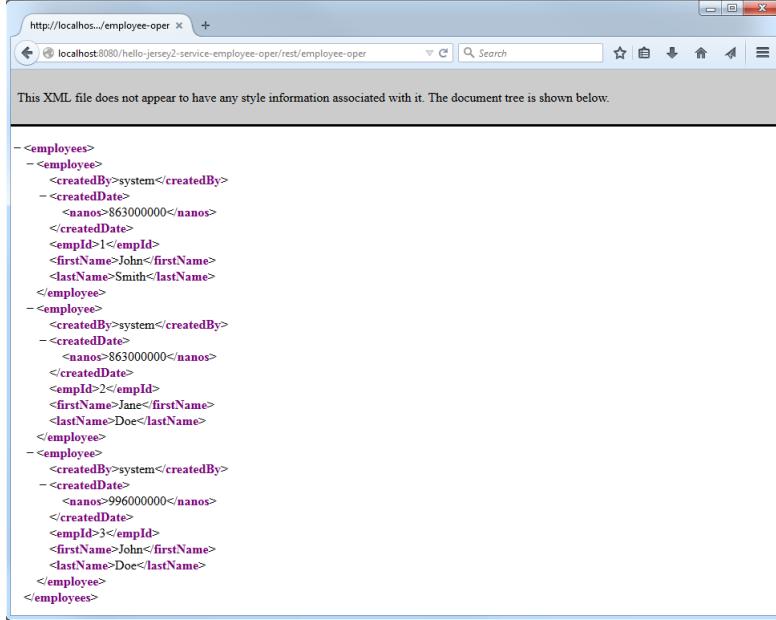
2016-02-12 17:02:17 INFO EmployeeOperApp:37 -
    ↵   InboundJaxrsResponse{context=ClientResponse{method=GET,
    ↵   uri=http://localhost:8080/hello-jersey2-service-employee-oper/rest/employee-oper,
    ↵   status=200, reason=OK}}
2016-02-12 17:02:17 INFO EmployeeOperApp:41 - [Employee [empid: 1 lastName =
    ↵   Smith firstName: John createdBy: system createdDate = 2016-02-12
    ↵   16:59:57.863], Employee [empid: 2 lastName = Doe firstName: Jane createdBy:
    ↵   system createdDate = 2016-02-12 16:59:57.863]]
2016-02-12 17:02:17 INFO EmployeeOperApp:44 -
    ↵   [{"empId":1,"lastName":"Smith","firstName":"John","createdBy":"system","createdDate":1455317997863},{"empId":2,"la
2016-02-12 17:02:17 INFO EmployeeOperApp:48 - <?xml version="1.0"
    ↵   encoding="UTF-8"
    ↵   standalone="yes"?><employees><employee><createdBy>system</createdBy><createdDate><nanos>863000000</nanos></crea
2016-02-12 17:02:17 INFO EmployeeOperApp:51 - <?xml version="1.0"
    ↵   encoding="UTF-8"
    ↵   standalone="yes"?><employees><employee><createdBy>system</createdBy><createdDate><nanos>863000000</nanos></crea

```

```

2016-02-12 17:02:18 INFO EmployeeOperApp:76 - empId = 1001
2016-02-12 17:02:18 INFO EmployeeOperApp:78 -
  ↳ [{"empId":1001,"lastName":"Borwn","firstName":"Charlie","createdBy":"system","createdDate":1455318138065}, {"empId":1001,"lastName":"Doe","firstName":"Jane","createdBy":"system","createdDate":1455318138065}, {"empId":1001,"lastName":"Smith","firstName":"John","createdBy":"system","createdDate":1455317997863}, {"empId":1002,"lastName":"Doe","firstName":"Jane","createdBy":"system","createdDate":1455317997863}, {"empId":1003,"lastName":"Doe","firstName":"John","createdBy":"system","createdDate":1455317997863}]
2016-02-12 17:02:18 INFO EmployeeOperApp:84 - Emp ID = 1001
2016-02-12 17:02:18 INFO EmployeeOperApp:85 - Last Name = Borwn
2016-02-12 17:02:18 INFO EmployeeOperApp:86 - First Name = Charlie
2016-02-12 17:02:18 INFO EmployeeOperApp:87 - Created By = system
2016-02-12 17:02:18 INFO EmployeeOperApp:88 - Created Date = 2016-02-12
  ↳ "17:02:18.065"
2016-02-12 17:02:18 INFO EmployeeOperApp:94 - employee count = 4
2016-02-12 17:02:18 INFO EmployeeOperApp:96 - Employee [empid: 1001 lastName =
  ↳ Borwn firstName: Charlie createdBy: system createdDate = 2016-02-12
  ↳ "17:02:18.065"]
2016-02-12 17:02:18 INFO EmployeeOperApp:96 - Employee [empid: 1 lastName =
  ↳ Smith firstName: John createdBy: system createdDate = 2016-02-12
  ↳ "16:59:57.863"]
2016-02-12 17:02:18 INFO EmployeeOperApp:96 - Employee [empid: 2 lastName = Doe
  ↳ firstName: Jane createdBy: system createdDate = 2016-02-12 16:59:57.863]
2016-02-12 17:02:18 INFO EmployeeOperApp:96 - Employee [empid: 3 lastName = Doe
  ↳ firstName: John createdBy: system createdDate = 2016-02-12 17:02:17.996]
2016-02-12 17:02:18 INFO EmployeeOperApp:102 - employee count = 3
2016-02-12 17:02:18 INFO EmployeeOperApp:103 -
  ↳ [{"empId":1,"lastName":"Smith","firstName":"John","createdBy":"system","createdDate":1455317997863}, {"empId":2,"lastName":"Doe","firstName":"Jane","createdBy":"system","createdDate":1455317997863}, {"empId":3,"lastName":"Doe","firstName":"John","createdBy":"system","createdDate":1455317997863}]]
```

<http://localhost:8080/hello-jersey2-service-employee-oper/rest/employee-oper>



16.2 SOAP

16.2.1 SOAP Service - Using Endpoint Publisher

16.2.1.1 Create project using command line

```

1 c:\workspaces\workspace\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=app-hello-soap-service
  ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

16.2.1.2 Make this project an eclipse project

```
1 c:\workspaces\ eclipse\my-projects\app-hello-soap-service>mvn eclipse:clean
→ eclipse:eclipse
```

16.2.1.3 Import this project to eclipse

- Launch eclipse
File → Import → General → Existing Projects into Workspace
Browse to app-hello-soap-service, and import it.

16.2.1.4 Use both service endpoint interface and service endpoint implementation

- A service endpoint interface or service endpoint implementation (SEI) is a Java interface or class, respectively, that declares the methods that a client can invoke on the service.
- You may specify an explicit interface by adding the endpointInterface element to the @WebService annotation in the implementation class. You must then provide an interface that defines the public methods made available in the endpoint implementation class.
- Create a Web Service Endpoint Interface
src/main/java/com/xyz/normal/HelloWorld.java

```
1 package com.xyz.normal;
2
3 import javax.jws.WebMethod;
4 import javax.jws.WebService;
5
6 @WebService
7 public interface HelloWorld {
8     @WebMethod
9     public String hello(String name);
10 }
```

- Create a Web Service Endpoint Implementation
src/main/java/com/xyz/normal/HelloWorldImpl.java

```
1 package com.xyz.normal;
2
3 import javax.jws.WebService;
4
5 @WebService(endpointInterface = "com.xyz.normal.HelloWorld")
6 public class HelloWorldImpl implements HelloWorld {
7     @Override
8     public String hello(String name) {
9         return "Hello " + name;
10    }
11 }
```

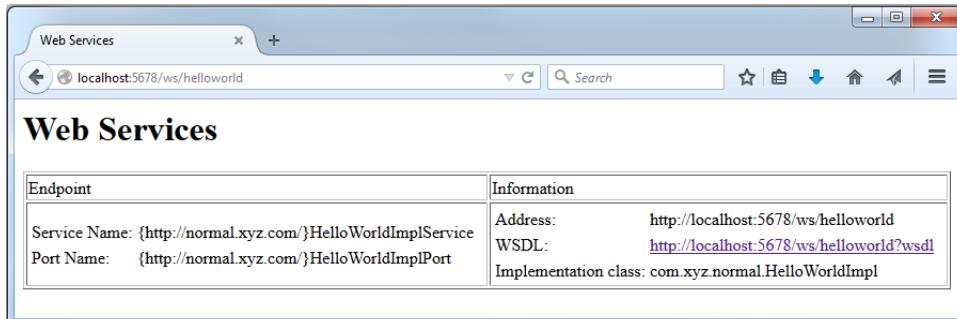
- Create a Endpoint Publisher
src/main/java/com/xyz/normal/HelloWorldPublisher.java

```
1 package com.xyz.normal;
2
3 import javax.xml.ws.Endpoint;
4
5 public class HelloWorldPublisher {
6     public static void main(String[] args) {
7         Endpoint.publish("http://localhost:5678/ws/helloworld", new
8             HelloWorldImpl());
```

```
8     }
9 }
```

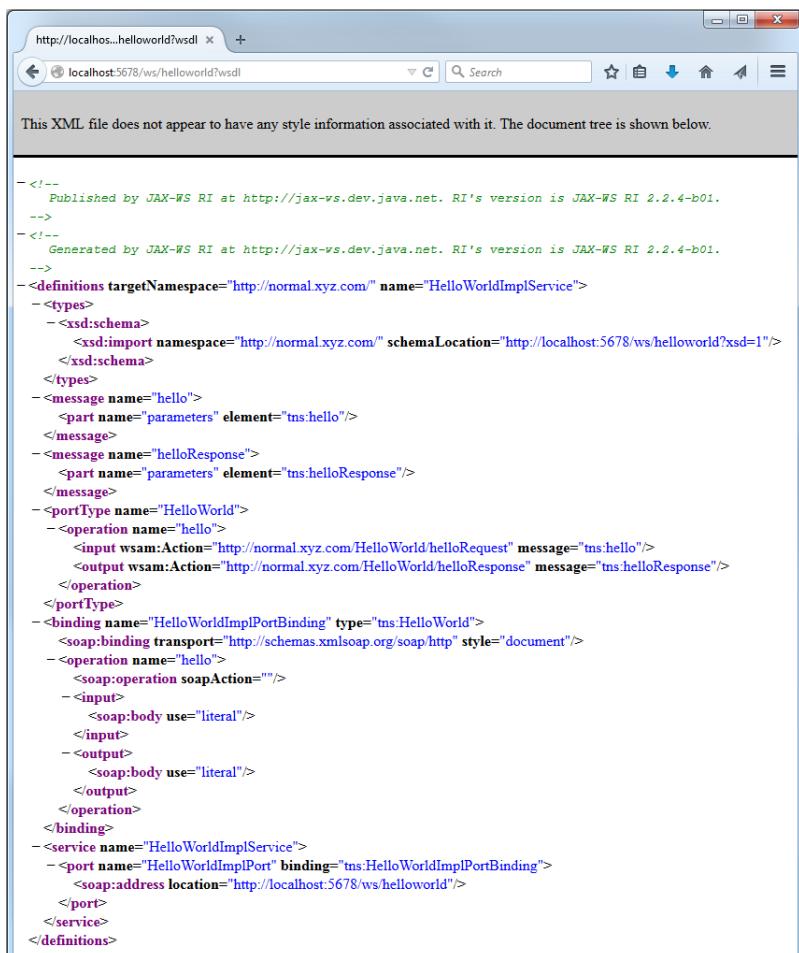
16.2.1.5 Run

- Right click on project src/main/java ->com.xyz.normal ->HelloWorldPublisher
Run As ->Java Application



The screenshot shows a "Web Services" browser window. The address bar contains "localhost:5678/ws/helloworld". The main content area is titled "Web Services" and displays the following table:

Endpoint	Information
Service Name: {http://normal.xyz.com/}HelloWorldImplService Port Name: {http://normal.xyz.com/}HelloWorldImplPort	Address: http://localhost:5678/ws/helloworld WSDL: http://localhost:5678/ws/helloworld?wsdl Implementation class: com.xyz.normal.HelloWorldImpl



The screenshot shows a browser window with the URL "http://localhost:5678/ws/helloworld?wsdl". The page content starts with the message: "This XML file does not appear to have any style information associated with it. The document tree is shown below." Below this, the WSDL XML code is displayed:

```

<!--
  Published by JAX-WS RI at http://jax-vs.dev.java.net. RI's version is JAX-WS RI 2.2.4-b01.
-->
<!--
  Generated by JAX-WS RI at http://jax-vs.dev.java.net. RI's version is JAX-WS RI 2.2.4-b01.
-->
<definitions targetNamespace="http://normal.xyz.com/" name="HelloWorldImplService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://normal.xyz.com/" schemaLocation="http://localhost:5678/ws/helloworld?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="hello">
    <part name="parameters" element="tns:hello"/>
  </message>
  <message name="helloResponse">
    <part name="parameters" element="tns:helloResponse"/>
  </message>
  <portType name="HelloWorld">
    <operation name="hello">
      <input wsam:Action="http://normal.xyz.com/HelloWorld/helloRequest" message="tns:hello"/>
      <output wsam:Action="http://normal.xyz.com/HelloWorld/helloResponse" message="tns:helloResponse"/>
    </operation>
  </portType>
  <binding name="HelloWorldImplPortBinding" type="tns:HelloWorld">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="hello">
      <soap:operation soapAction="" />
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="HelloWorldImplService">
    <port name="HelloWorldImplPort" binding="tns:HelloWorldImplPortBinding">
      <soap:address location="http://localhost:5678/ws/helloworld"/>
    </port>
  </service>
</definitions>
```

- wsdl

```

<?xml version="1.0" encoding="UTF-8"?>
  <!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
      -->
  <!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
      -->
<definitions>
  <!-- xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
      xmlns: wsp="http://www.w3.org/ns/ws-policy"
      xmlns: wspi_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
      xmlns: wsam="http://www.w3.org/2007/05/addressing/metadata"
      xmlns: xsdp="http://schemas.xmlsoap.org/wsdl/soap/"
      xmlns: tns="http://normal.xyz.com/"
      xmlns: xsd="http://www.w3.org/2001/XMLSchema"
      xmlns="http://schemas.xmlsoap.org/wsdl/"
      targetNamespace="http://normal.xyz.com/"
      name="HelloWorldImplService">
    <types>
      <xsd:schema>
        <xsd:import namespace="http://normal.xyz.com/" schemaLocation="http://localhost:5678/ws/helloworld?xsd=1"></xsd:import>
      </xsd:schema>
    </types>
    <message name="hello">
      <part name="parameters" element="tns:hello"></part>
    </message>
    <message name="helloResponse">
      <part name="parameters" element="tns:helloResponse"></part>
    </message>
    <portType name="HelloWorld">
      <operation name="hello">
        <input wsam:Action="http://normal.xyz.com/HelloWorld/helloRequest"
               message="tns:hello"></input>
        <output wsam:Action="http://normal.xyz.com/HelloWorld/helloResponse"
               message="tns:helloResponse"></output>
      </operation>
    </portType>
    <binding name="HelloWorldImplPortBinding" type="tns:HelloWorld">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
                    style="document"></soap:binding>
      <operation name="hello">
        <soap:operation soapAction=""></soap:operation>
        <input>
          <soap:body use="literal"></soap:body>
        </input>
        <output>
          <soap:body use="literal"></soap:body>
        </output>
      </operation>
    </binding>
    <service name="HelloWorldImplService">
      <port name="HelloWorldImplPort" binding="tns:HelloWorldImplPortBinding">
        <soap:address
          location="http://localhost:5678/ws/helloworld"></soap:address>
      </port>
    </service>
  </definitions>

```

16.2.1.6 Web Service Client Without Tool

- Without tool, you can create a Java web service client like this.
- Still use app-hello-soap-service, because we need HelloWorld Interface and HelloWordImpl class.
- Create a Web Service Client
src/main/java/com/xyz/normal/HelloWorldClient.java

```

1 package com.xyz.normal;
2
3 import java.net.URL;
4
5 import javax.xml.namespace.QName;
6 import javax.xml.ws.Service;
7
8 public class HelloWorldClient {
9     public static void main(String[] args) throws Exception {
10         URL url = new URL("http://localhost:5678/ws/helloworld?wsdl");
11
12         // 1st argument service URI, refer to wsdl document above
13         // 2nd argument is service name, refer to wsdl document above
14         QName qname = new QName("http://normal.xyz.com/",
15             "HelloWorldImplService");
16
17         Service service = Service.create(url, qname);
18
19         HelloWorld helloWorld = service.getPort(HelloWorld.class);
20
21         System.out.println(helloWorld.hello("xyz"));
22     }
23 }
```

Right click on project src/main/java ->com.xyz ->HelloWorldClient
Run As ->Java Application

Hello xyz

16.2.1.7 Only use service endpoint implementation

- An interface is not required when building a JAX-WS endpoint. The web service implementation class implicitly defines an SEI.
- Create the Service Endpoint Implementation Class

The implementation class, HelloWorld, is annotated as a web service endpoint using the @WebService annotation. HelloWorld declares a single method named hello, annotated with the @WebMethod annotation. @WebMethod exposes the annotated method to web service clients. hello returns a greeting to the client, using the name passed to hello to compose the greeting. The implementation class also must define a default, public, no-argument constructor.

src/main/java/com/xyz/service/HelloWorld.java

```

1 package com.xyz.service;
2
3 import javax.jws.WebMethod;
4 import javax.jws.WebService;
5
6 @WebService
7 public class HelloWorld {
8     public HelloWorld() {
9     }
10
11     @WebMethod
12     public String hello(String name) {
13         return "Hello " + name;
14     }
15 }
```

- Create a Endpoint Publisher

src/main/java/com/xyz/service/HelloWorldPublisher.java

```

1 package com.xyz.service;
2
3 import javax.xml.ws.Endpoint;
4
5 public class HelloWorldPublisher {
6     public static void main(String[] args) {
7         Endpoint.publish("http://localhost:1234/ws/helloworld", new
8             HelloWorld());
9     }
}

```

16.2.1.8 Run

- Right click on project src/main/java ->com.xyz.service ->HelloWorldPublisher
Run As ->Java Application

The image contains two screenshots of a Java IDE's Web Services interface.

Top Screenshot: A window titled "Web Services" displays the following endpoint information:

Endpoint	Information
Service Name: {http://service.xyz.com/}HelloWorldService	Address: http://localhost:1234/ws/helloworld
Port Name: {http://service.xyz.com/}HelloWorldPort	WSDL: http://localhost:1234/ws/helloworld?wsdl
	Implementation class: com.xyz.service.HelloWorld

Bottom Screenshot: A window titled "http://localhost:1234/ws/helloworld?wsdl" shows the WSDL XML document. The XML content is as follows:

```

<!--
Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.4-b01.
-->
<!--
Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.4-b01.
-->
<definitions targetNamespace="http://service.xyz.com/" name="HelloWorldService">
    <types>
        <xsd:schema>
            <xsd:import namespace="http://service.xyz.com/" schemaLocation="http://localhost:1234/ws/helloworld?xsd=1"/>
        </xsd:schema>
    </types>
    <message name="hello">
        <part name="parameters" element="tns:hello"/>
    </message>
    <message name="helloResponse">
        <part name="parameters" element="tns:helloResponse"/>
    </message>
    <portType name="HelloWorld">
        <operation name="hello">
            <input wsam:Action="http://service.xyz.com/HelloWorld/helloRequest" message="tns:hello"/>
            <output wsam:Action="http://service.xyz.com/HelloWorld/helloResponse" message="tns:helloResponse"/>
        </operation>
    </portType>
    <binding name="HelloWorldPortBinding" type="tns:HelloWorld">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
        <operation name="hello">
            <soap:operation soapAction="" />
            <input>
                <soap:body use="literal"/>
            </input>
            <output>
                <soap:body use="literal"/>
            </output>
        </operation>
    </binding>
</service name="HelloWorldService">
<port name="HelloWorldPort" binding="tns:HelloWorldPortBinding">
    <soap:address location="http://localhost:1234/ws/helloworld"/>
</port>
</service>
</definitions>

```

■ wsdl

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
    -->
    JAX-WS RI 2.2.4-b01. -->
<!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
    -->
    JAX-WS RI 2.2.4-b01. -->
<definitions
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    xmlns: wsp="http://www.w3.org/ns/ws-policy"
    xmlns: wspi_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns: wsam="http://www.w3.org/2007/05/addressing/metadata"
    xmlns: soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns: tns="http://service.xyz.com/"
    xmlns: xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    targetNamespace="http://service.xyz.com/"
    name="HelloWorldService">
<types>
    <xsd:schema>
        <xsd:import namespace="http://service.xyz.com/"
            schemaLocation="http://localhost:1234/ws/helloworld?xsd=1"></xsd:import>
    </xsd:schema>
</types>
<message name="hello">
    <part name="parameters" element="tns:hello"></part>
</message>
<message name="helloResponse">
    <part name="parameters" element="tns:helloResponse"></part>
</message>
<portType name="HelloWorld">
    <operation name="hello">
        <input wsam:Action="http://service.xyz.com/HelloWorld/helloRequest"
            message="tns:hello"></input>
        <output wsam:Action="http://service.xyz.com/HelloWorld/helloResponse"
            message="tns:helloResponse"></output>
    </operation>
</portType>
<binding name="HelloWorldPortBinding" type="tns:HelloWorld">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
        style="document"></soap:binding>
    <operation name="hello">
        <ssoap:operation soapAction=""></ssoap:operation>
        <input>
            <soap:body use="literal"></soap:body>
        </input>
        <output>
            <soap:body use="literal"></soap:body>
        </output>
    </operation>
</binding>
<service name="HelloWorldService">
    <port name="HelloWorldPort" binding="tns:HelloWorldPortBinding">
        <ssoap:address
            location="http://localhost:1234/ws/helloworld"></ssoap:address>
    </port>
</service>
</definitions>

```

16.2.2 Java Web Service Client via wsimport tool

16.2.2.1 Create project using command line

```

1 c:\workspaces\workspace\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↳ -DartifactId=app-hello-soap-client
  ↳ -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

```

16.2.2.2 Make this project an eclipse project

```
1 c:\workspaces\workspace\my-projects\app-hello-soap-client>mvn eclipse:clean
  ↳ eclipse:eclipse
```

16.2.2.3 Import this project to eclipse

- Launch eclipse

File → Import → General → Existing Projects into Workspace

Browse to app-hello-soap-client, and import it.

16.2.2.4 wsimport

- you can use "wsimport" tool to parse the published wsdl file, and generate necessary client files (stub) to access the published web service.
 - This wsimport tool is bundle with the JDK, you can find it at "JDK_PATH/bin" folder.
 - Issue "wsimport" command.
- In command line, go to your workspace, project directory, src/main/java directory

```
1 wsimport -keep http://localhost:1234/ws/helloworld?wsdl
```

```
1 c:\>cd \workspaces\workspace\my-projects\app-hello-soap-client\src\main\java
2 c:\workspaces\workspace\my-projects\app-hello-soap-client\src\main\java>wsimport
  ↳ -keep http://localhost:1234/ws/helloworld?wsdl
3 parsing WSDL...
4
5 Generating code...
6
7 Compiling code...
8
9 c:\workspaces\workspace\my-projects\app-hello-soap-client\src\main\java>
```

- It will generate necessary client files, which depends on the provided wsdl file. In this case, it will generate one interface and one service implementation file.
- src/main/java/com/xyz/service/HelloWorld.java

```
1 package com.xyz.service;
2
3 import javax.jws.WebMethod;
4 import javax.jws.WebParam;
5 import javax.jws.WebResult;
6 import javax.jws.WebService;
7 import javax.xml.bind.annotation.XmlSeeAlso;
8 import javax.xml.ws.Action;
9 import javax.xml.ws.RequestWrapper;
10 import javax.xml.ws.ResponseWrapper;
11
12
13 /**
14  * This class was generated by the JAX-WS RI.
15  * JAX-WS RI 2.2.4-b01
16  * Generated source version: 2.2
17  *
18  */
19 @WebService(name = "HelloWorld", targetNamespace = "http://service.xyz.com/")
20 @XmlSeeAlso({
21     ObjectFactory.class
22 })
23 public interface HelloWorld {
24
25 }
```

```

26     /**
27      *
28      * @param arg0
29      * @return
30      *      returns java.lang.String
31      */
32     @WebMethod
33     @WebResult(targetNamespace = "")
34     @RequestWrapper(localName = "hello", targetNamespace =
35         "http://service.xyz.com/", className = "com.xyz.service.Hello")
36     @ResponseWrapper(localName = "helloResponse", targetNamespace =
37         "http://service.xyz.com/", className = "com.xyz.service.HelloResponse")
38     @Action(input = "http://service.xyz.com/HelloWorld/helloRequest", output =
39         "http://service.xyz.com/HelloWorld/helloResponse")
40     public String hello(
41         @WebParam(name = "arg0", targetNamespace = "")  

42         String arg0);
43
44 }

```

src/main/java/com/xyz/service/HelloWorldService.java

```

1 package com.xyz.service;
2
3 import java.net.MalformedURLException;
4 import java.net.URL;
5 import javax.xml.namespace.QName;
6 import javax.xml.ws.Service;
7 import javax.xml.ws.WebEndpoint;
8 import javax.xml.ws.WebServiceClient;
9 import javax.xml.ws.WebServiceException;
10 import javax.xml.ws.WebServiceFeature;
11
12
13 /**
14  * This class was generated by the JAX-WS RI.
15  * JAX-WS RI 2.2.4-b01
16  * Generated source version: 2.2
17  *
18  */
19 @WebServiceClient(name = "HelloWorldService", targetNamespace =
20     "http://service.xyz.com/", wsdlLocation =
21     "http://localhost:1234/ws/helloworld?wsdl")
22 public class HelloWorldService
23     extends Service
24 {
25
26     private final static URL HELLOWORLDSERVICE_WSDL_LOCATION;
27     private final static WebServiceException HELLOWORLDSERVICE_EXCEPTION;
28     private final static QName HELLOWORLDSERVICE_QNAME = new
29         QName("http://service.xyz.com/", "HelloWorldService");
30
31     static {
32         URL url = null;
33         WebServiceException e = null;
34         try {
35             url = new URL("http://localhost:1234/ws/helloworld?wsdl");
36         } catch (MalformedURLException ex) {
37             e = new WebServiceException(ex);
38         }
39         HELLOWORLDSERVICE_WSDL_LOCATION = url;
40         HELLOWORLDSERVICE_EXCEPTION = e;
41     }
42
43     public HelloWorldService() {
44         super(__getWsdlLocation(), HELLOWORLDSERVICE_QNAME);
45     }
46
47     public HelloWorldService(WebServiceFeature... features) {
48         super(__getWsdlLocation(), HELLOWORLDSERVICE_QNAME, features);
49     }
50
51 }

```

```

48     public HelloWorldService(URL wsdlLocation) {
49         super(wsdlLocation, HELLOWORLDSERVICE_QNAME);
50     }
51
52     public HelloWorldService(URL wsdlLocation, WebServiceFeature... features) {
53         super(wsdlLocation, HELLOWORLDSERVICE_QNAME, features);
54     }
55
56     public HelloWorldService(URL wsdlLocation, QName serviceName) {
57         super(wsdlLocation, serviceName);
58     }
59
60     public HelloWorldService(URL wsdlLocation, QName serviceName,
61         WebServiceFeature... features) {
62         super(wsdlLocation, serviceName, features);
63     }
64
65     /**
66      *
67      * @return
68      *      returns HelloWorld
69      */
70     @WebEndpoint(name = "HelloWorldPort")
71     public HelloWorld getHelloWorldPort() {
72         return super.getPort(new QName("http://service.xyz.com/",
73             "HelloWorldPort"), HelloWorld.class);
74     }
75
76     /**
77      *
78      * @param features
79      *      A list of {@link javax.xml.ws.WebServiceFeature} to configure on the
80      * proxy. Supported features not in the <code>features</code> parameter will
81      * have their default values.
82      * @return
83      *      returns HelloWorld
84      */
85     @WebEndpoint(name = "HelloWorldPort")
86     public HelloWorld getHelloWorldPort(WebServiceFeature... features) {
87         return super.getPort(new QName("http://service.xyz.com/",
88             "HelloWorldPort"), HelloWorld.class, features);
89     }
90
91     private static URL __getWsdlLocation() {
92         if (HELLOWORLDSERVICE_EXCEPTION!= null) {
93             throw HELLOWORLDSERVICE_EXCEPTION;
94         }
95         return HELLOWORLDSERVICE_WSDL_LOCATION;
96     }
97 }

```

It also generate Hello.java, HelloResponse.java, ObjectFactory.java, package-info.java

16.2.2.5 Java Web Service Client

- Create a Java web service client which depends on the above generated files.
src/main/java/com/xyz/client/HelloWorldClient.java

```

1 package com.xyz.client;
2
3 import com.xyz.service.HelloWorld;
4 import com.xyz.service.HelloWorldService;
5
6 public class HelloWorldClient {
7     public static void main(String[] args) throws Exception {
8         HelloWorldService helloWorldService = new HelloWorldService();
9         HelloWorld helloWorld = helloWorldService.getHelloWorldPort();
10        System.out.println(helloWorld.hello("xyz"));
11    }
12 }

```

16.2.2.6 Run

- Right click on project src/main/java ->com.xyz ->HelloWorldClient
Run As ->Java Application

```
Hello xyz
```

16.2.3 Tracing SOAP Traffic

16.2.3.1 Trace SOAP message in Eclipse IDE

- In SOAP web service, each HTTP request or response encapsulates a SOAP envelope, these messages are easy to trace by using Eclipse IDE, build-in "TCP/IP monitor" tool. The idea is host another server in between the client and server to perform port forward function to intercept the HTTP traffic.
 - Normal SOAP envelope flows In normal SOAP service, client send a HTTP request to server, and server send back a HTTP response to client directly.

```
1. Client ----> SOAP envelope ----> Server:1234
2. Server:1234 ----> SOAP envelope ---> Client
```

- Intercepted SOAP envelope flows To intercept SOAP envelope, you can host another server ("TcpMonitorServer") in between client and server, see new flows:

```
1. Client ----> SOAP envelope ----> TcpMonitorServer:8888
2. TcpMonitorServer:8888 --> SOAP envelope ---> Server:1234
3. Server:1234 ----> SOAP envelope ---> TcpMonitorServer:8888
4. TcpMonitorServer:8888 ----> SOAP envelope ---> Client
```

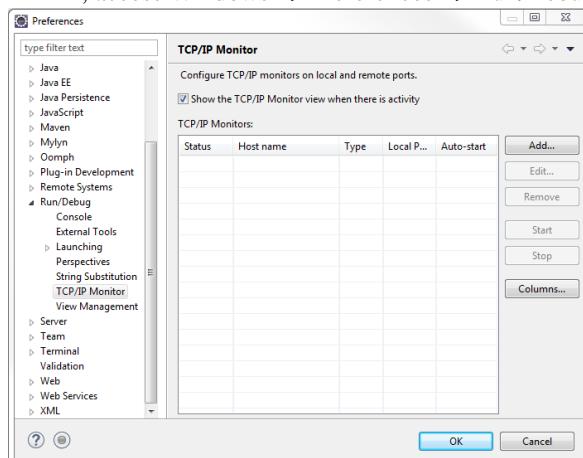
Note: This method required port changed in your web service client.

- Eclipse IDE + TCP/IP Monitor

Eclipse IDE comes with a very easy to use traffic interceptor tool, known as "TCP/IP Monitor". We can enable this "TCP/IP Monitor" in Eclipse IDE, and also intercept the SOAP messages generated by web service.

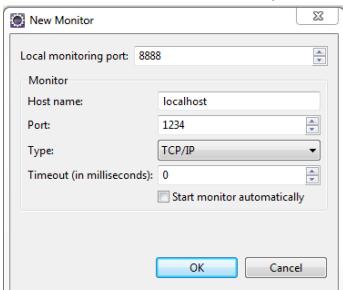
Heres the steps to enable "TCP/IP Monitor" in Eclispe IDE.

- In IDE, access Windows ->Preferences ->Run/Debug ->TCP/IP Monitor

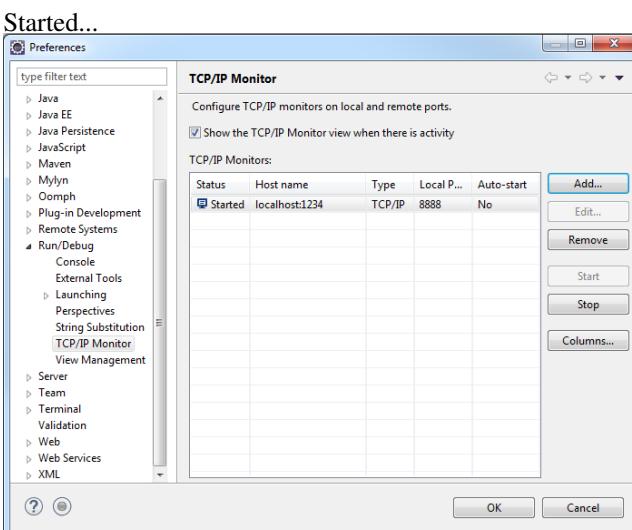
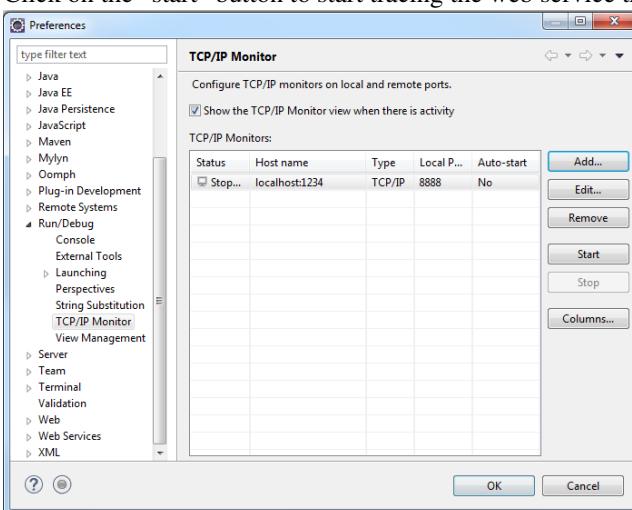


Click Add...

- Fill in server information, and choose type = "TCP/IP"



- Click on the "start" button to start tracing the web service traffic.



- Add "TCP/IP Monitor" view.

In IDE, access Windows ->Show View ->Other... ->Debug ->TCP/IP Monitor

- Create a Web Service Client

Still using app-hello-soap-client project. But port needs to be changed to 8888 for

TCP/IP Monitor

src/main/java/com/xyz/client/HelloWorldMonitorClient.java

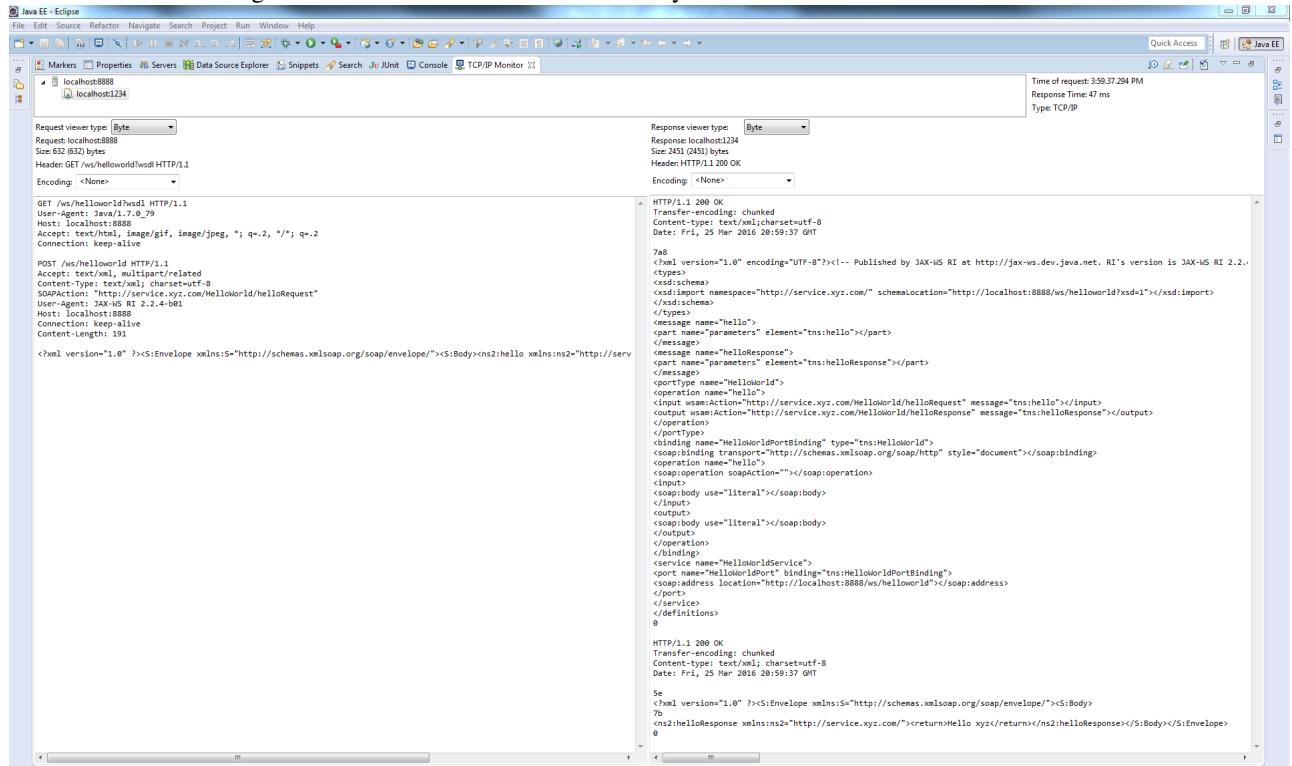
```

1 package com.xyz.client;
2
3 import java.net.URL;
4
5 import javax.xml.namespace.QName;
6 import javax.xml.ws.Service;
7
8 import com.xyz.service.HelloWorld;
9
10 public class HelloWorldMonitorClient {
11     public static void main(String[] args) throws Exception {
12         URL url = new URL("http://localhost:8888/ws/helloworld?wsdl");
13         QName qname = new QName("http://service.xyz.com/", "HelloWorldService");
14         Service service = Service.create(url, qname);
15         HelloWorld helloWorld = service.getPort(HelloWorld.class);
16         System.out.println(helloWorld.hello("xyz"));
17     }
18 }
```

Right click on project src/main/java ->com.xyz.client ->HelloWorldMonitorClient
Run As ->Java Application

Hello xyz

- Show the traced messaged in the "TCP/IP Monitor" view if any.



16.2.3.2 Explain SOAP Traffic

- Request a WSDL file

First, client send a wsdl request to service endpoint, see HTTP traffic below:

Client send request:

```
GET /ws/helloworld?wsdl HTTP/1.1
User-Agent: Java/1.7.0_79
Host: localhost:8888
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
```

Server send response:

```
HTTP/1.1 200 OK
Transfer-encoding: chunked
Content-type: text/xml;charset=utf-8
Date: Fri, 25 Mar 2016 20:59:37 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
    -->
<!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
    -->
<definitions
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    xmlns:wsp="http://www.w3.org/ns/ws-policy"
    xmlns:wspl_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://service.xyz.com/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    targetNamespace="http://service.xyz.com/"
    name="HelloWorldService">
    <types>
        <xsd:schema>
            <xsd:import namespace="http://service.xyz.com/" schemaLocation="http://localhost:8888/ws/helloworld?xsd=1"></xsd:import>
        </xsd:schema>
    </types>
    <message name="hello">
        <part name="parameters" element="tns:hello"></part>
    </message>
    <message name="helloResponse">
        <part name="parameters" element="tns:helloResponse"></part>
    </message>
    <portType name="HelloWorld">
        <operation name="hello">
            <input wsam:Action="http://service.xyz.com/HelloWorld/helloRequest" message="tns:hello"></input>
            <output wsam:Action="http://service.xyz.com/HelloWorld/helloResponse" message="tns:helloResponse"></output>
        </operation>
    </portType>
    <binding name="HelloWorldPortBinding" type="tns:HelloWorld">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"></soap:binding>
        <operation name="hello">
            <soap:operation soapAction=""></soap:operation>
            <input>
                <soap:body use="literal"></soap:body>
            </input>
            <output>
                <soap:body use="literal"></soap:body>
            </output>
        </operation>
    </binding>
    <service name="HelloWorldService">
        <port name="HelloWorldPort" binding="tns:HelloWorldPortBinding">
            <soap:address
                location="http://localhost:8888/ws/helloworld"></soap:address>
```

```
</port>
</service>
</definitions>
```

- **helloWorld.hello()**

A second call, client put method invoke request in SOAP envelope and send it to service endpoint. At the service endpoint, call the requested method and put the result in a SOAP envelope and send it back to client.

Client send request:

```
POST /ws/helloworld HTTP/1.1
Accept: text/xml, multipart/related
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://service.xyz.com/HelloWorld/helloRequest"
User-Agent: JAX-WS RI 2.2.4-b01
Host: localhost:8888
Connection: keep-alive
Content-Length: 191

<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:hello xmlns:ns2="http://service.xyz.com/">
      <arg0>xyz</arg0>
    </ns2:hello>
  </S:Body>
</S:Envelope>
```

Server send response:

```
HTTP/1.1 200 OK
Transfer-encoding: chunked
Content-type: text/xml; charset=utf-8
Date: Fri, 25 Mar 2016 20:59:37 GMT

<?xml version="1.0" ?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:helloResponse xmlns:ns2="http://service.xyz.com/">
      <return>Hello xyz</return>
    </ns2:helloResponse>
  </S:Body>
</S:Envelope>
```

16.2.4 SOAP Service - Deploy JAX-WS web services on Tomcat

16.2.4.1 Create project using command line

```
1 c:\workspaces\ eclipse\my-projects>mvn archetype:generate -DgroupId=com.xyz
  ↵ -DartifactId=hello-soap-service
  ↵ -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false
```

16.2.4.2 Modify pom.xml

- C:\workspaces\ eclipse\my-projects\hello-soap-service\pom.xml

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"
  ↵ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  ↵ http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.xyz</groupId>
5   <artifactId>hello-soap-service</artifactId>
```

```

6  <packaging>war</packaging>
7  <version>1.0-SNAPSHOT</version>
8  <name>hello-soap-service Maven Webapp</name>
9  <url>http://maven.apache.org</url>
10 <properties>
11   <java.version>1.7</java.version>
12   <junit.version>4.12</junit.version>
13   <maven.compiler.plugin.version>3.3</maven.compiler.plugin.version>
14   <jaxws.version>2.2.10</jaxws.version>
15 </properties>
16 <dependencies>
17   <dependency>
18     <groupId>com.sun.xml.ws</groupId>
19     <artifactId>jaxws-rt</artifactId>
20     <version>${jaxws.version}</version>
21   </dependency>
22   <dependency>
23     <groupId>junit</groupId>
24     <artifactId>junit</artifactId>
25     <version>${junit.version}</version>
26     <scope>test</scope>
27   </dependency>
28 </dependencies>
29 <build>
30   <finalName>hello-soap-service</finalName>
31   <plugins>
32     <plugin>
33       <groupId>org.apache.maven.plugins</groupId>
34       <artifactId>maven-compiler-plugin</artifactId>
35       <version>${maven.compiler.plugin.version}</version>
36       <configuration>
37         <source>${java.version}</source>
38         <target>${java.version}</target>
39       </configuration>
40     </plugin>
41   </plugins>
42 </build>
43 </project>

```

16.2.4.3 Make this project an eclipse project

```

1 c:\workspaces\eclipse\my-projects\hello-soap-service>mvn eclipse:clean
  ↵  eclipse:eclipse -Dwtpversion=2.0

```

16.2.4.4 Import this project to eclipse

- Launch eclipse
- File → Import → General → Existing Projects into Workspace
- Browse to hello-soap-service, and import it.

16.2.4.5 Service

- Create java directory and add it to Build Path
src/main/java
- Create the Service Endpoint Implementation Class
src/main/java/com/xyz/service/HelloWorld.java

```

1 package com.xyz.service;
2
3 import javax.jws.WebMethod;
4 import javax.jws.WebService;
5
6 @WebService

```

```

7  public class HelloWorld {
8      public HelloWorld() {
9          }
10     @WebMethod
11     public String hello(String name) {
12         return "Hello " + name;
13     }
14 }

```

16.2.4.6 Web Service Deployment Descriptor

- Create a web service deployment descriptor, which is also known as JAX-WS RI deployment descriptor - sun-jaxws.xml
src/main/webapp/WEB-INF/sun-jaxws.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <endpoints xmlns="http://java.sun.com/xml/ns/jax-ws/ri/runtime" version="2.0">
3     <endpoint
4         name="hello-soap-service"
5         implementation="com.xyz.service.HelloWorld"
6         url-pattern="/helloworld" />
7 </endpoints>

```

When user access /hello/ URL path, it will fire the declared web service, which is HelloWorld.java.

16.2.4.7 web.xml Deployment Descriptor

- Modifyte web.xml deployment descriptor for the deployment. Defines WSServletContextListener as listener class, WSServlet as your hello servlet.
src/main/webapp/WEB-INF/web.xml

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
5         http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
6     version="2.4">
7
8     <display-name>Hello Soap Service</display-name>
9
10    <listener>
11        <listener-class>com.sun.xml.ws.transport.http.servlet.WSServletContextListener</listener-class>
12    </listener>
13
14    <servlet>
15        <servlet-name>helloworld</servlet-name>
16
17        <servlet-class>com.sun.xml.ws.transport.http.servlet.WSServlet</servlet-class>
18        <load-on-startup>1</load-on-startup>
19    </servlet>
20
21    <servlet-mapping>
22        <servlet-name>helloworld</servlet-name>
23        <url-pattern>/helloworld</url-pattern>
24    </servlet-mapping>
25
26 </web-app>

```

16.2.4.8 Build the project

```
c:\workspaces\workspace\my-projects\hello-soap-service>mvn install
```

16.2.4.9 Run

- Add project to Tomcat Server

Go to Servers Tab

Right click Tomcat v7.0 Server at localhost [Stopped] → Add and Remove...

Move hello-soap-service from Available panel (left side) to Configured panel (right side)

- Start Tomcat Server

<http://localhost:8080/hello-soap-service/helloworld>

Endpoint	Information
Service Name: {http://service.xyz.com/}HelloWorldService Port Name: {http://service.xyz.com/}HelloWorldPort	Address: http://localhost:8080/hello-soap-service/helloworld WSDL: http://localhost:8080/hello-soap-service/helloworld?wsdl Implementation class: com.xyz.service.HelloWorld


```

<!--
Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.10 svn-revision#919b322c92f13ad085a933e8dd6dd35d4947364b.
-->
<!--
Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.10 svn-revision#919b322c92f13ad085a933e8dd6dd35d4947364b.
-->
<definitions targetNamespace="http://service.xyz.com" name="HelloWorldService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://service.xyz.com" schemaLocation="http://localhost:8080/hello-soap-service/helloworld?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="hello">
    <part name="parameters" element="tns:hello"/>
  </message>
  <message name="helloResponse">
    <part name="parameters" element="tns:helloResponse"/>
  </message>
  <portType name="HelloWorld">
    <operation name="hello">
      <input wsam:Action="http://service.xyz.com/HelloWorld/HelloRequest" message="tns:hello"/>
      <output wsam:Action="http://service.xyz.com/HelloWorld/HelloResponse" message="tns:helloResponse"/>
    </operation>
  </portType>
  <binding name="HelloWorldPortBinding" type="tns>HelloWorld">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="hello">
      <soap:operation soapAction="" />
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="HelloWorldService">
    <port name="HelloWorldPort" binding="tns>HelloWorldPortBinding">
      <soap:address location="http://localhost:8080/hello-soap-service/helloworld"/>
    </port>
  </service>
</definitions>

```

■ wsdl

```

<?xml version='1.0' encoding='UTF-8'?>
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS
     RI 2.2.10 svn-revision#919b322c92f13ad085a933e8dd6dd35d4947364b. -->
<!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS
     RI 2.2.10 svn-revision#919b322c92f13ad085a933e8dd6dd35d4947364b. -->
<definitions
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    xmlns: wsp="http://www.w3.org/ns/ws-policy"
    xmlns: wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns: wsam="http://www.w3.org/2007/05/addressing/metadata"
    xmlns: soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns: tns="http://service.xyz.com/"
    xmlns: xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://schemas.xmlsoap.org/wsdl/"
    targetNamespace="http://service.xyz.com/"
    name="HelloWorldService">
    <types>
        <xsd:schema>
            <xsd:import namespace="http://service.xyz.com/">
                <!-- schemaLocation="http://localhost:8080/hello-soap-service/helloworld?xsd=1" -->
                </xsd:import>
        </xsd:schema>
    </types>
    <message name="hello">
        <part name="parameters" element="tns:hello" />
    </message>
    <message name="helloResponse">
        <part name="parameters" element="tns:helloResponse" />
    </message>
    <portType name="HelloWorld">
        <operation name="hello">
            <input wsam:Action="http://service.xyz.com/HelloWorld/helloRequest"
                   message="tns:hello" />
            <output wsam:Action="http://service.xyz.com/HelloWorld/helloResponse"
                   message="tns:helloResponse" />
        </operation>
    </portType>
    <binding name="HelloWorldPortBinding" type="tns:HelloWorld">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
                      style="document" />
        <operation name="hello">
            <ssoap:operation soapAction="" />
            <input>
                <ssoap:body use="literal" />
            </input>
            <output>
                <ssoap:body use="literal" />
            </output>
        </operation>
    </binding>
    <service name="HelloWorldService">
        <port name="HelloWorldPort" binding="tns:HelloWorldPortBinding">
            <soap:address
                location="http://localhost:8080/hello-soap-service/helloworld" />
        </port>
    </service>
</definitions>
```

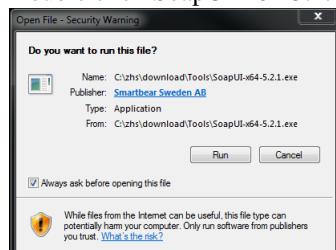
16.2.5 SoapUI

16.2.5.1 Download SoapUI

- Download SoapUI OpenSource (SoapUI-x64-5.2.1.exe) from:
<https://www.soapui.org/downloads/soapui.html>

16.2.5.2 Install SoapUI

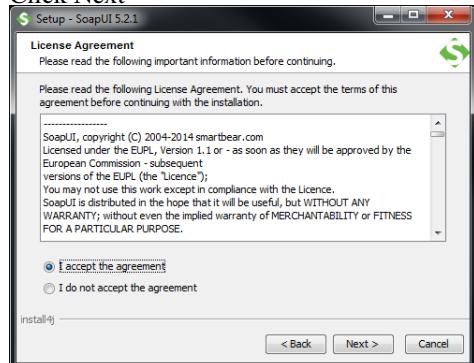
Double click SoapUI-x64-5.2.1.exe



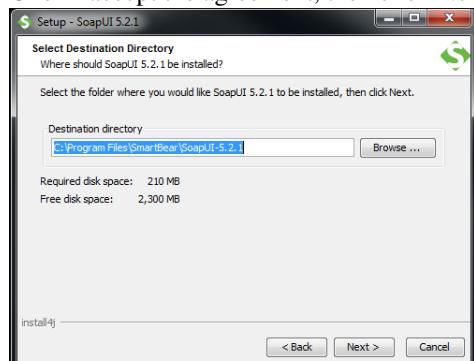
Click Run



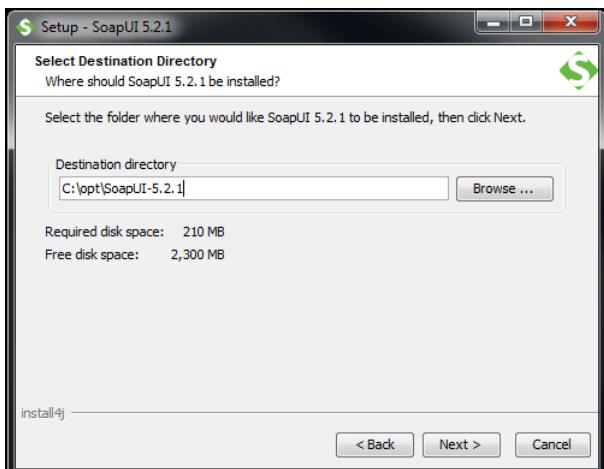
Click Next



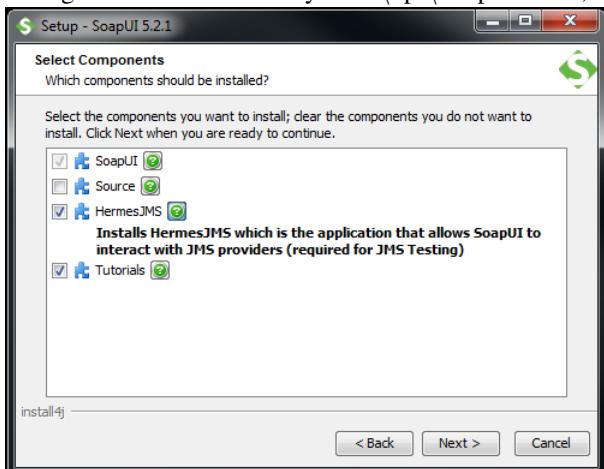
Click I accept the agreement, then click Next



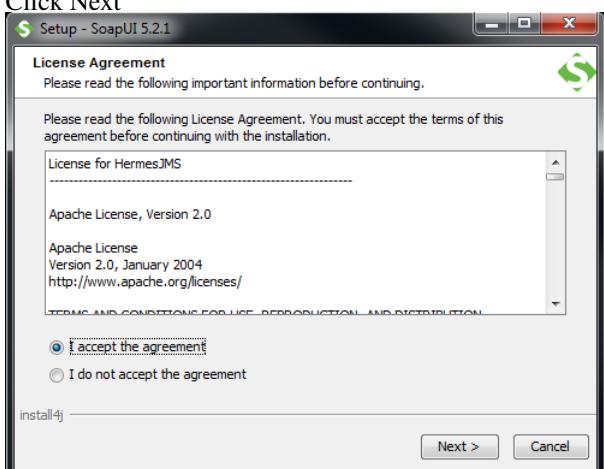
This is the default Destination directory



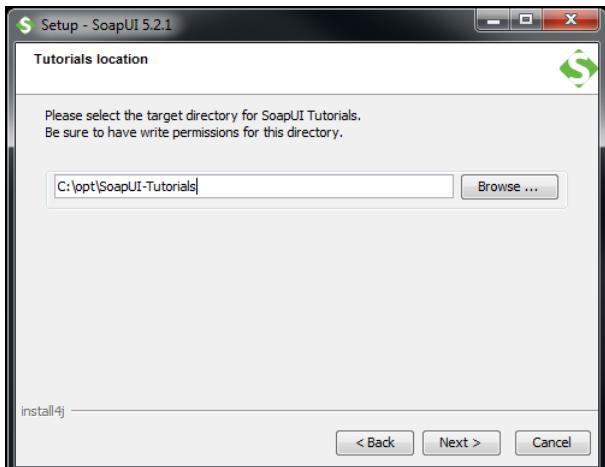
Change Destination directory to C:\opt\SoapUI-5.2.1, then click Next



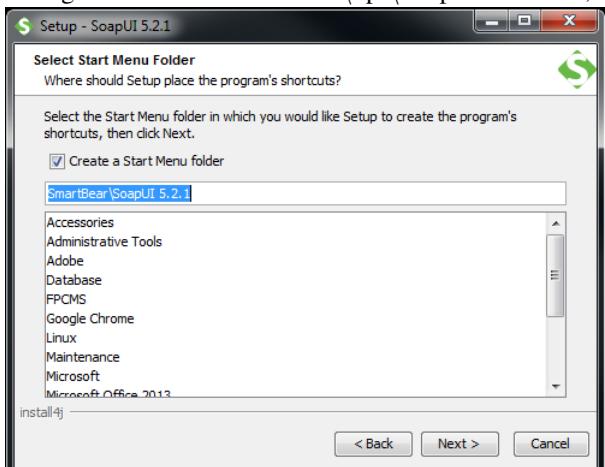
Click Next



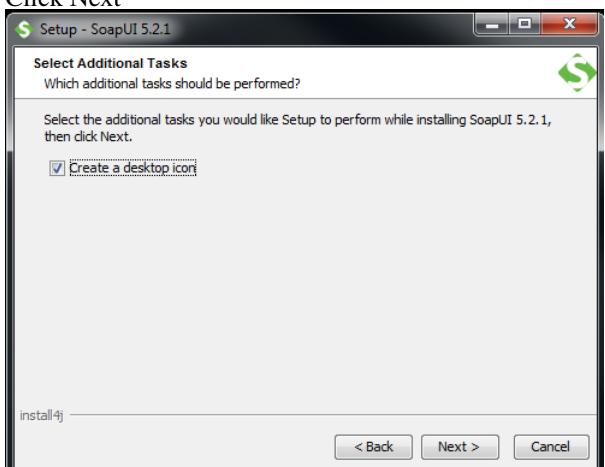
Click I accept the agreement, then click Next



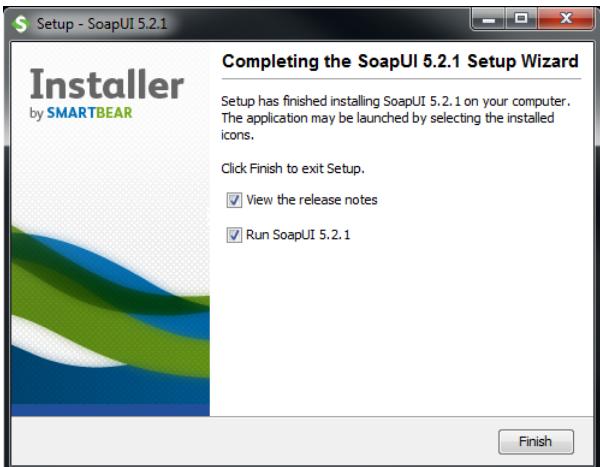
Change Tutorials location to C:\opt\SoapUI-Tutorials, then click Next



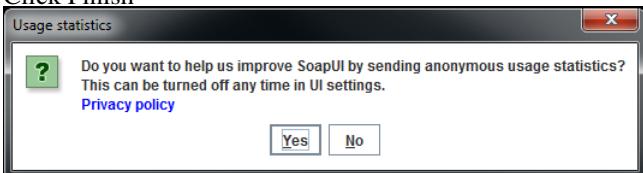
Click Next



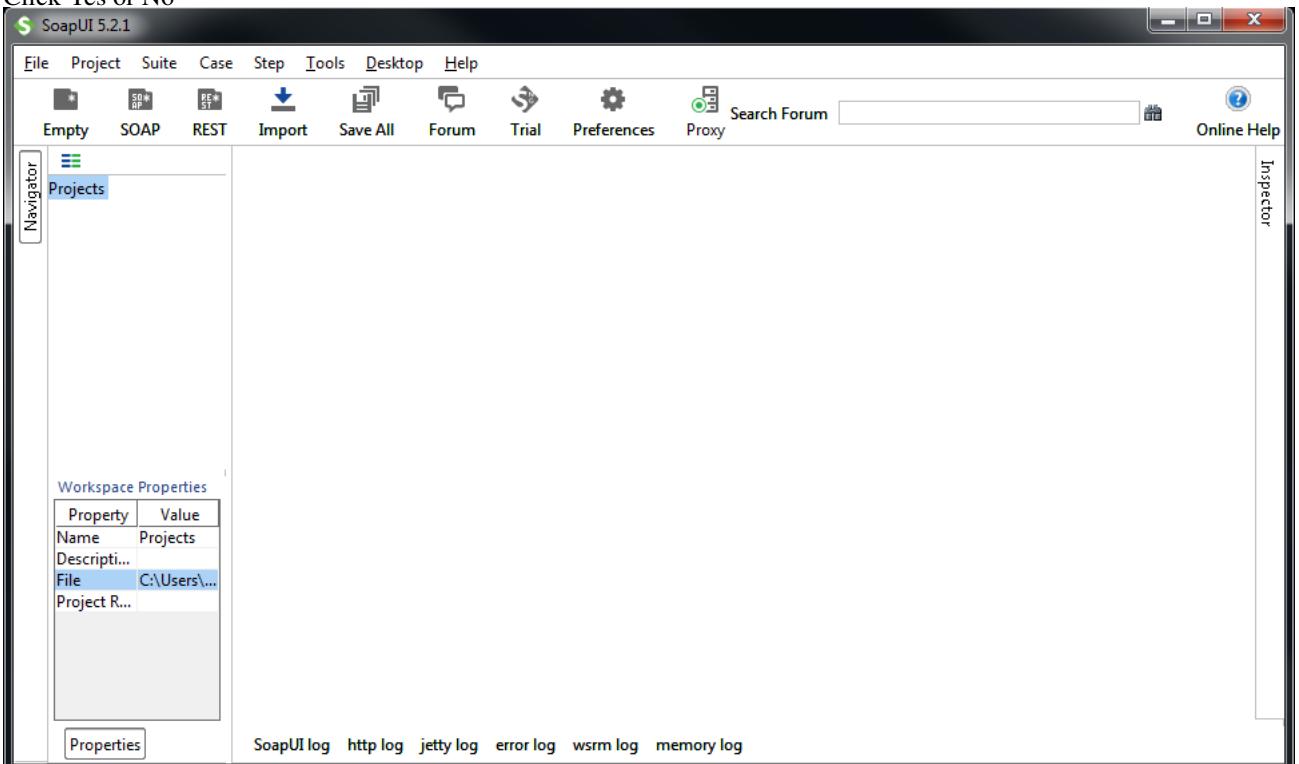
Click Next



Click Finish

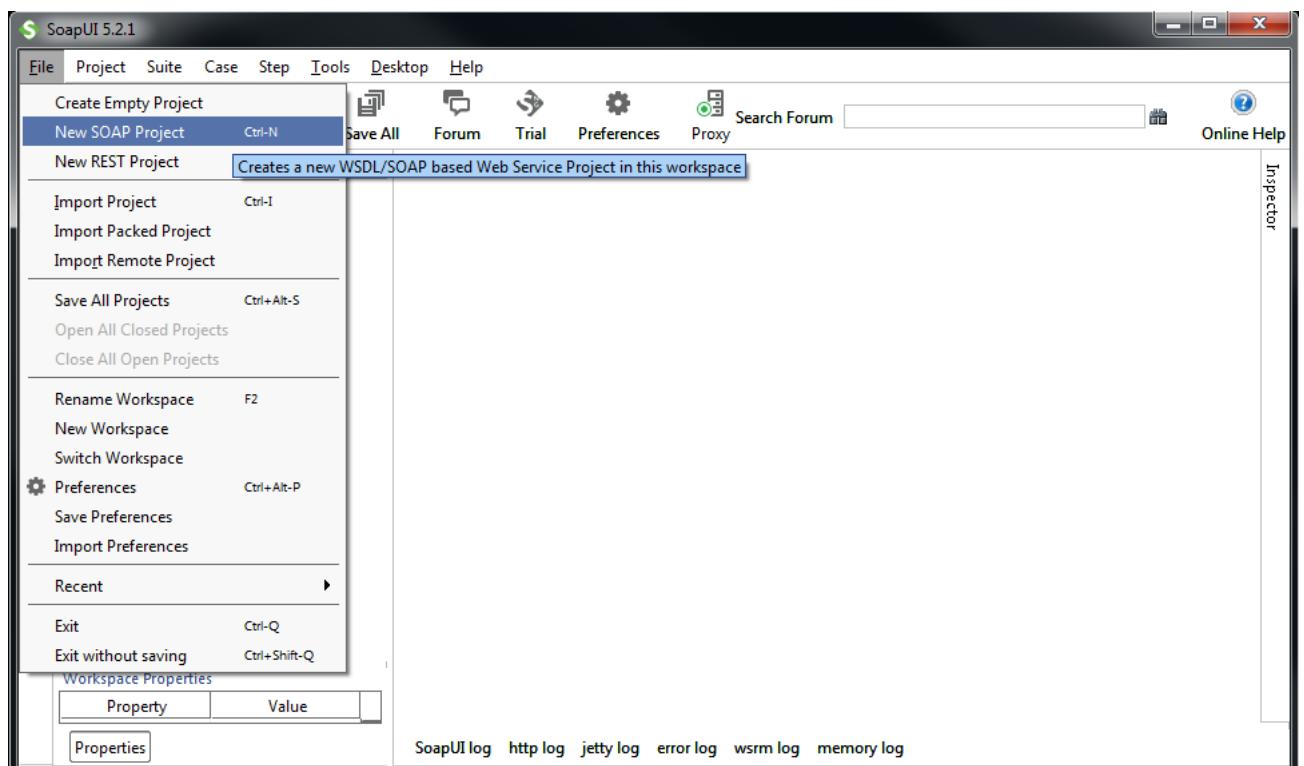


Click Yes or No

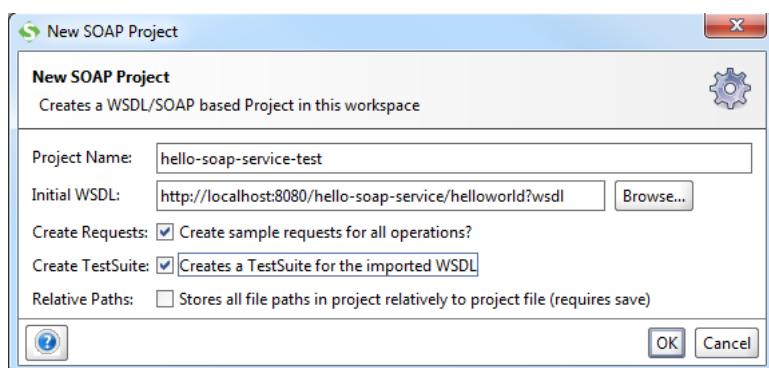


16.2.5.3 Install SoapUI

- Let's start hello-soap-service.
<http://localhost:8080/hello-soap-service/helloworld?wsdl>



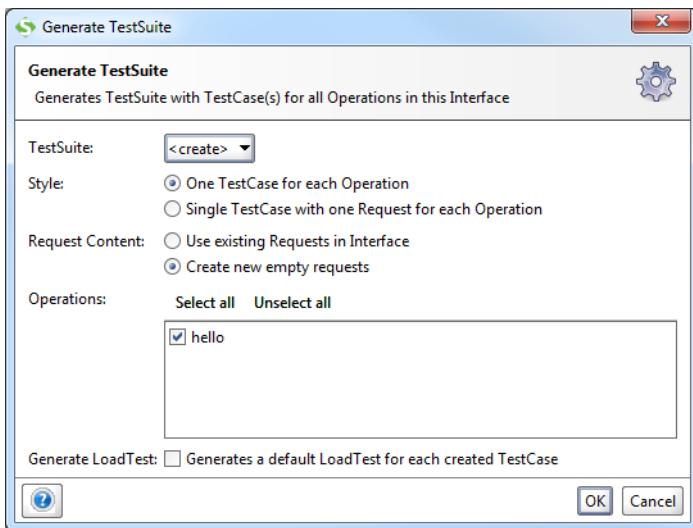
File → New SOAP Project



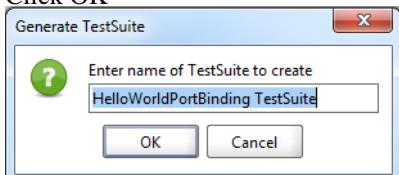
Project Name: hello-soap-service-test

Initial WSDL: http://localhost:8080/hello-soap-service/helloworld?wsdl

Check Create Requests and Create TestSuite, then click OK

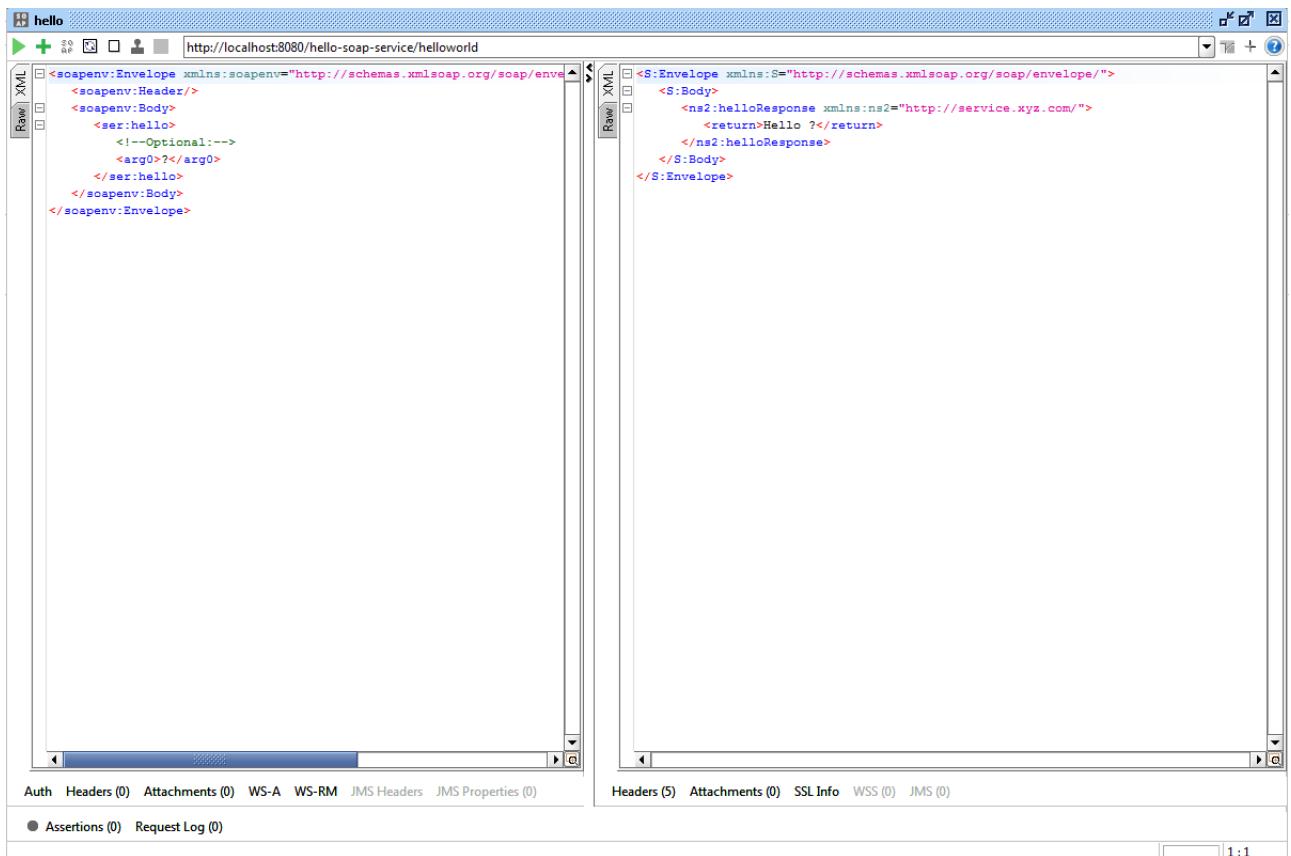


Click OK



Click OK

Click HelloWorldPortBinding, you can see Overview, Service Endpoints, WSDL Content, WS-I Compliance



Click hello TestCase, you can see request and response template
Request:

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
2   <!--Optional:-->
3   <soapenv:Header>
4     <soapenv:Body>
5       <ser:hello>
6         <!--Optional:-->
7         <arg0?></arg0>
8       </ser:hello>
9     </soapenv:Body>
</soapenv:Envelope>

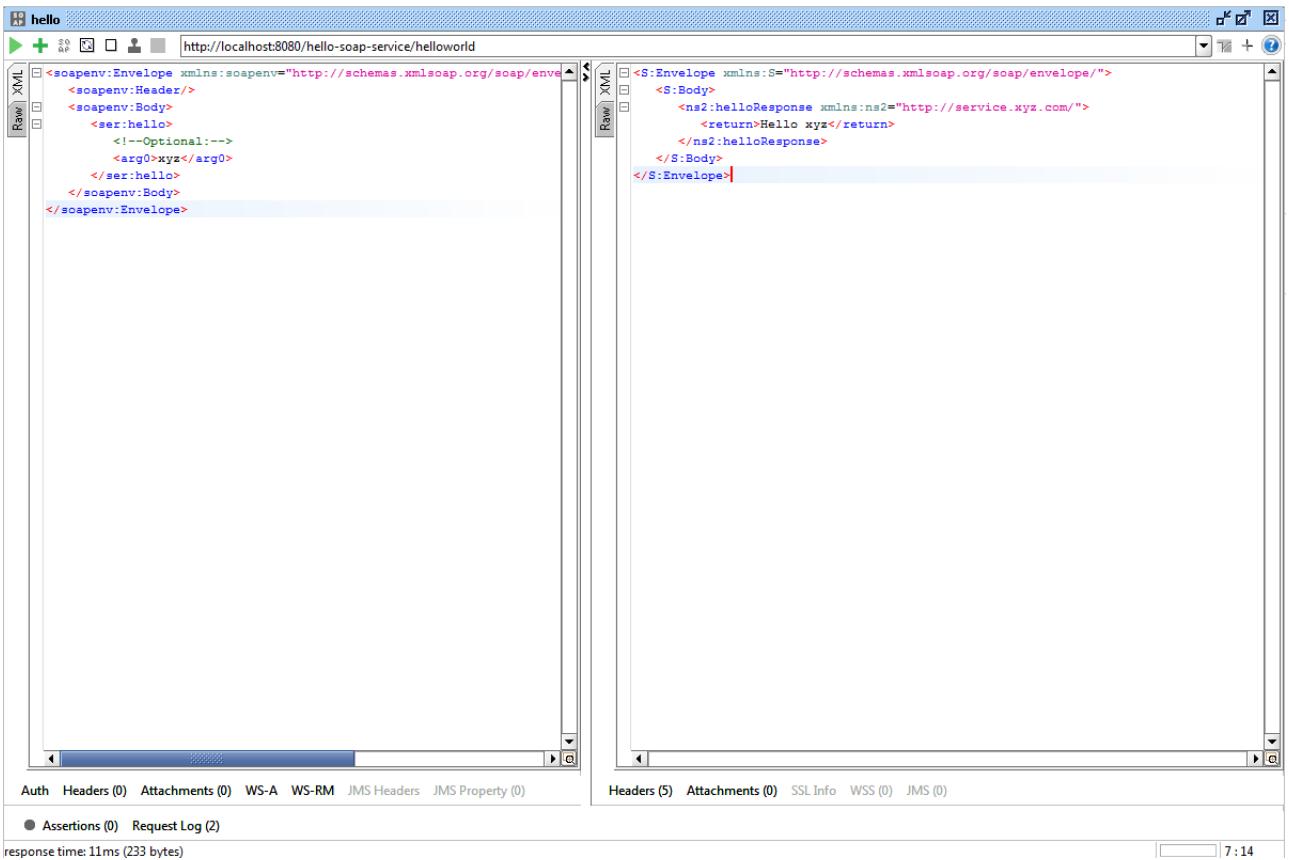
```

Response:

```

1 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
2   <S:Body>
3     <ns2:helloResponse xmlns:ns2="http://service.xyz.com/">
4       <return>Hello ?</return>
5     </ns2:helloResponse>
6   </S:Body>
7 </S:Envelope>

```



Change request arg0 to xyz, click green run button
Request:

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
2   <!--Optional:-->
3   <soapenv:Header/>
4   <soapenv:Body>
5     <ser:hello>
6       <!--Optional:-->
7       <arg0>xyz</arg0>
8     </ser:hello>
9   </soapenv:Body>
</soapenv:Envelope>

```

Response:

```

1 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope">
2   <S:Body>
3     <ns2:helloResponse xmlns:ns2="http://service.xyz.com/">
4       <return>Hello xyz</return>
5     </ns2:helloResponse>
6   </S:Body>
7 </S:Envelope>

```


PART VI

APPENDIX

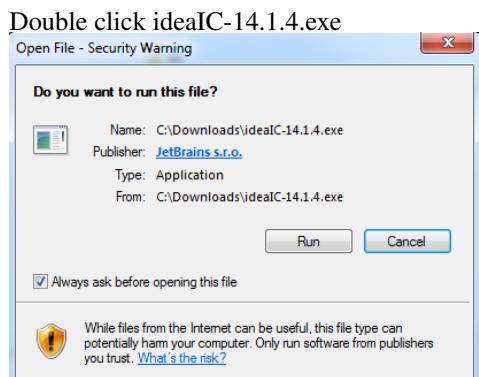
APPENDIX A

INTELLIJ

A.1 Download IntelliJ

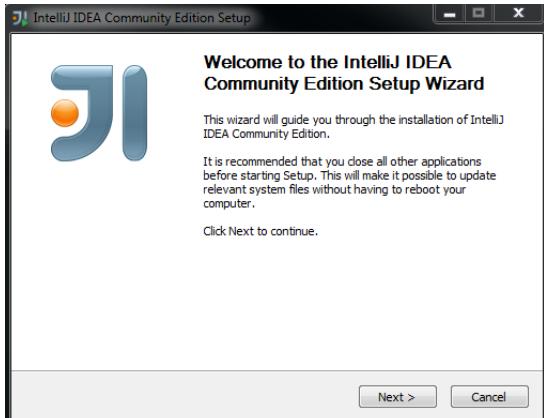
- Download IntelliJ Community Edition (ideaIC-14.1.4.exe) from:
<https://www.jetbrains.com/idea/download/>

A.2 Install IntelliJ

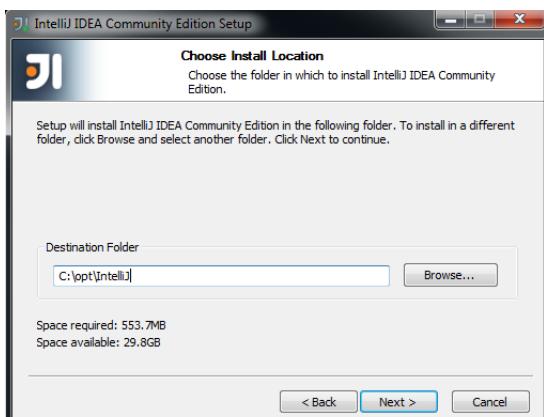


Click Run

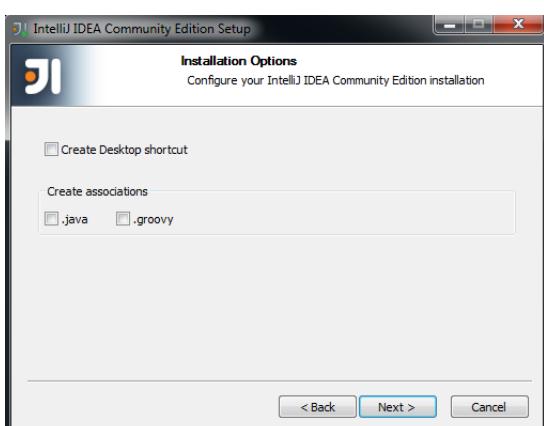
When Popup asking for administrator userid/password, click No



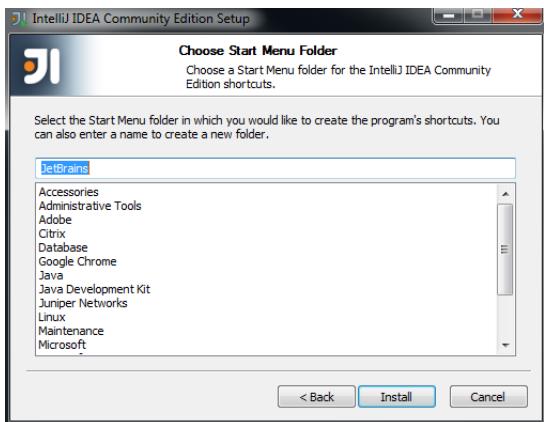
Click Next



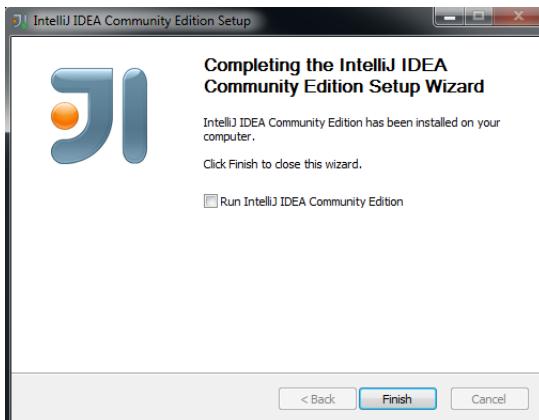
Change Destination Folder to C:\opt\IntelliJ, then click Next



Click Next



Click Install

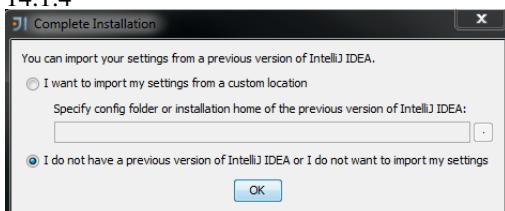


Click Finish

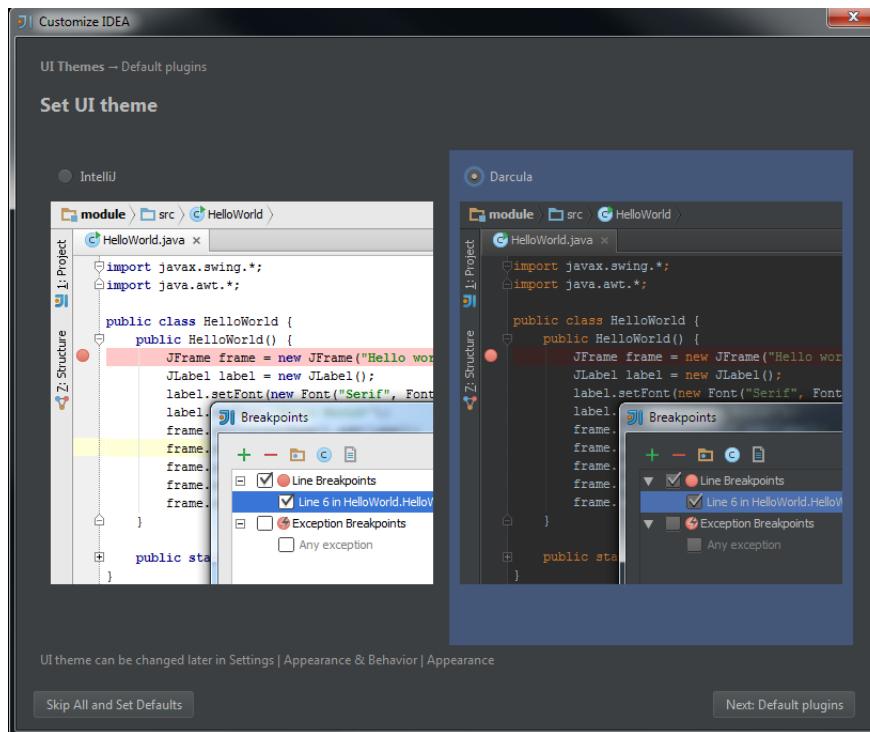
A.3 IntelliJ - hello-world web application

A.3.1 Lanuch IntelliJ

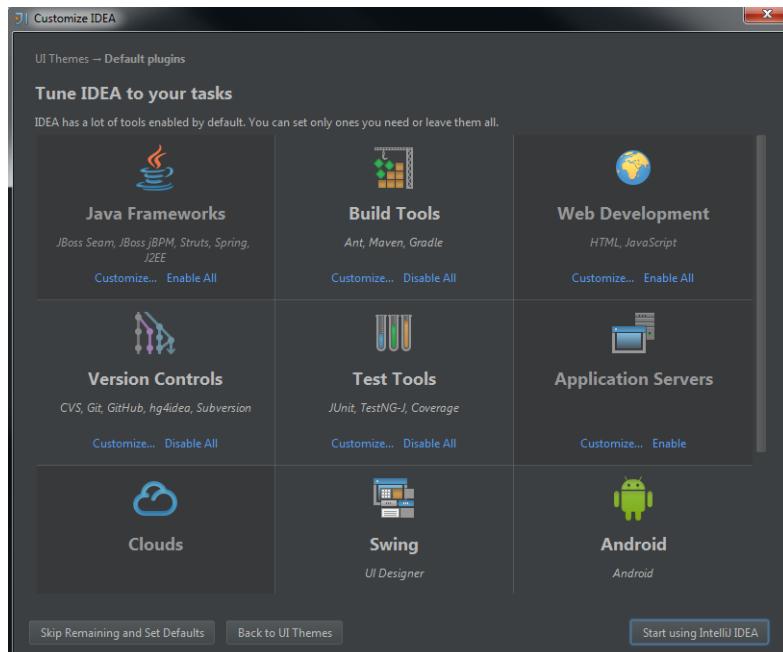
Go to Windows → All Programs → JetBrains → IntelliJ IDEA Community Edition 14.1.4



Click OK

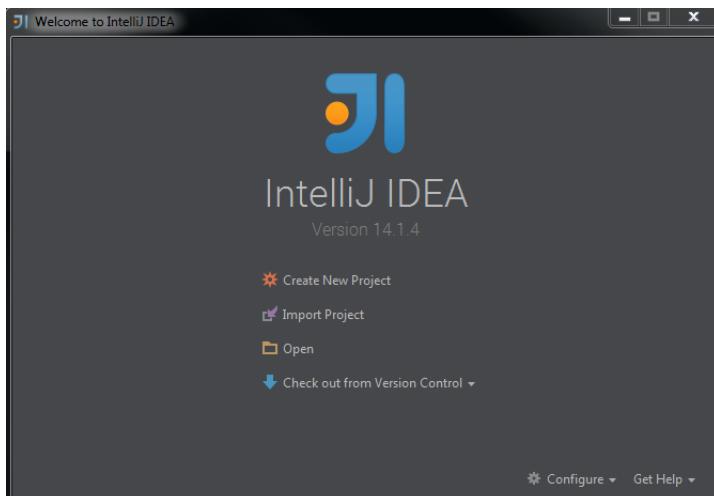


Select UI theme: Darcula
Click Next: Default plugins

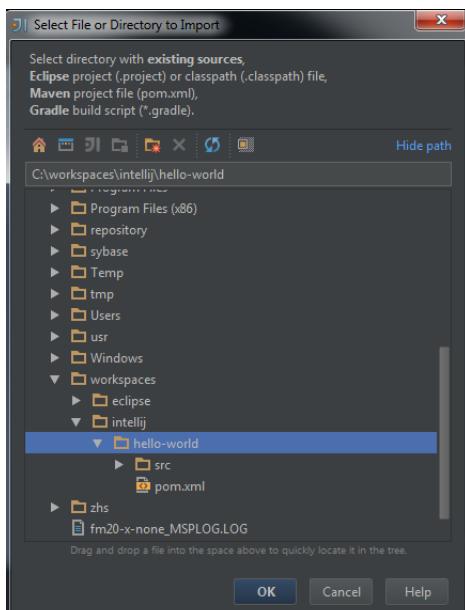


Select Start using IntelliJ IDEA

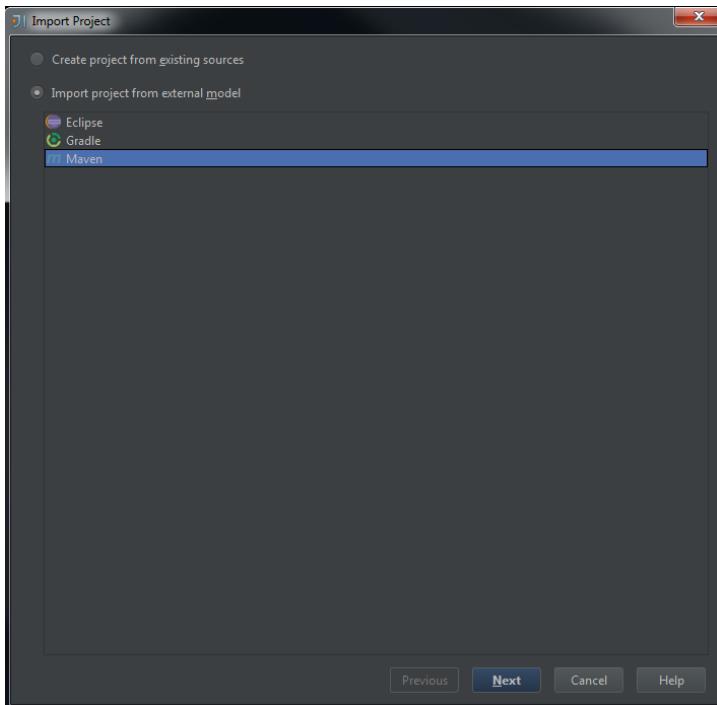
A.3.2 Import hello-world Maven Project



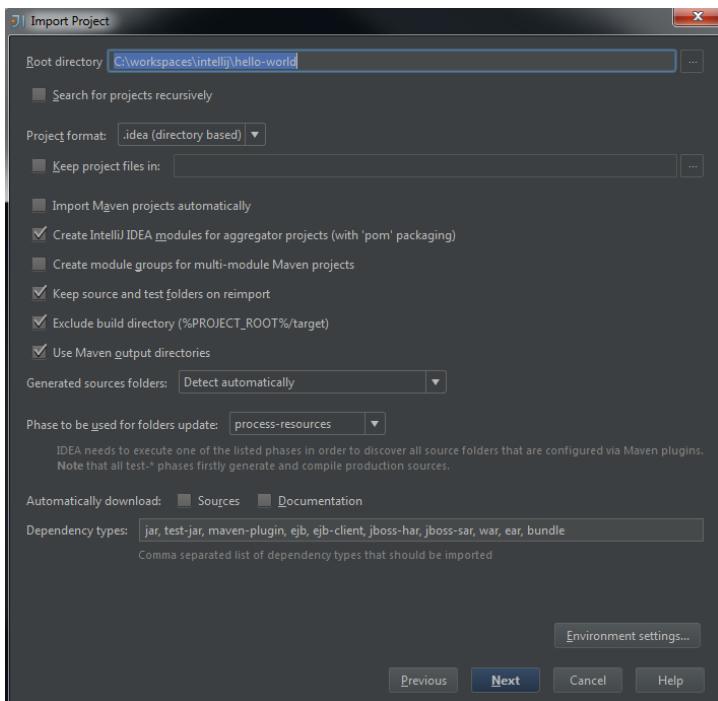
Click Import Project



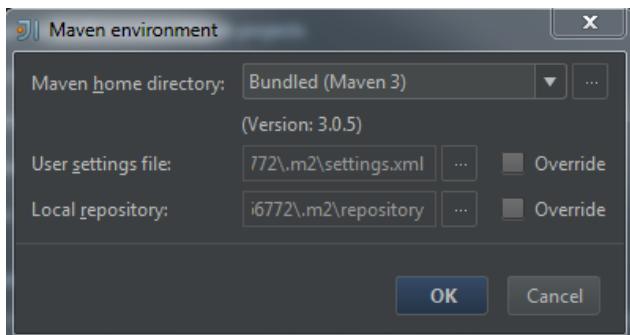
Select your hello-world maven project (C:\workspaces\intellij\hello-world), then click OK



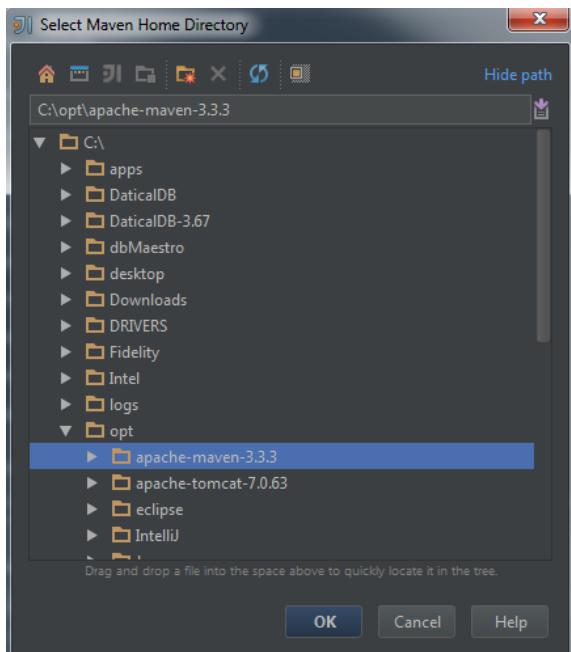
Select Import project from external model, Maven, then click Next



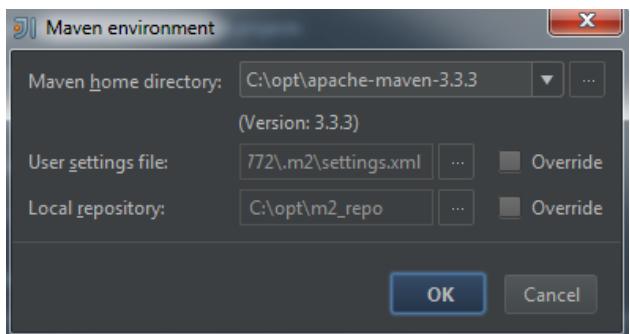
Click Environment settings



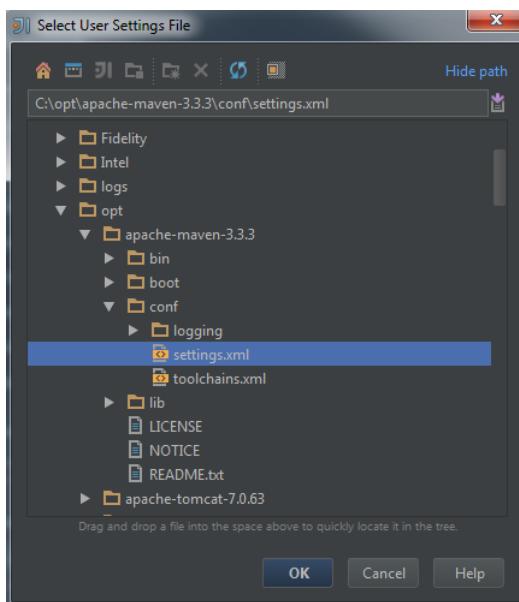
Change Maven home directory: click ...



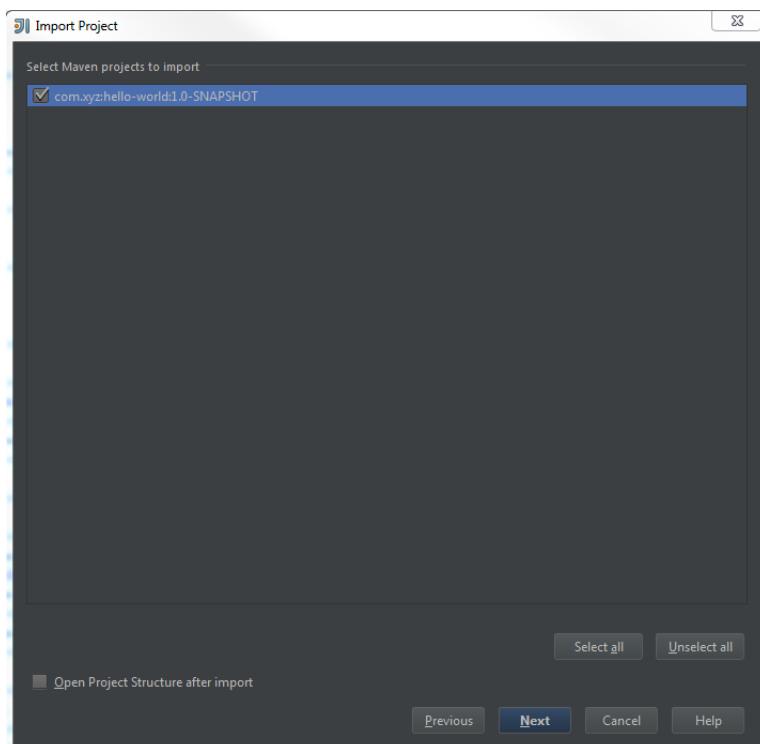
Select C:\opt\apache-maven-3.3.3, then click OK



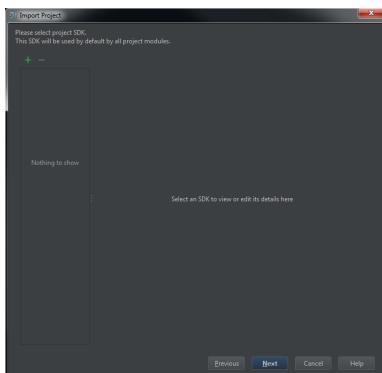
Change User settings file: check Override, then click ...



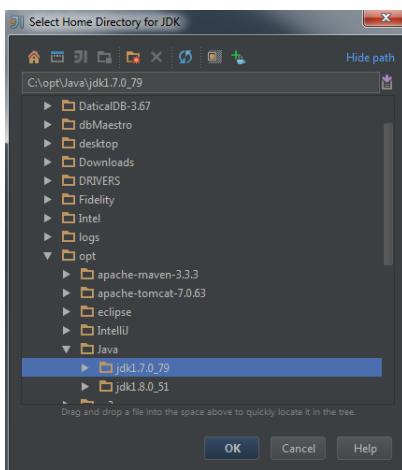
Select C:\opt\apache-maven-3.3.3\conf\settings.xml, then click OK



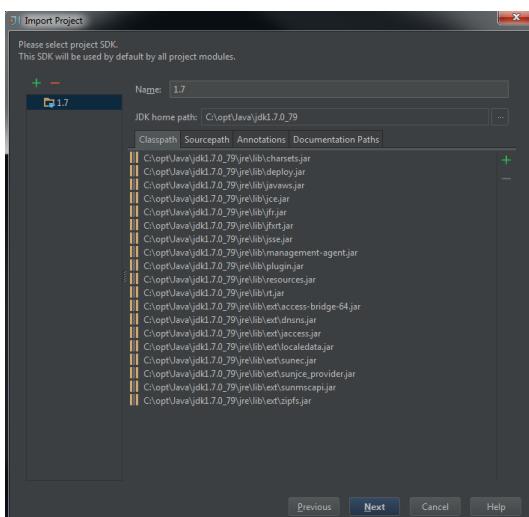
Click Next



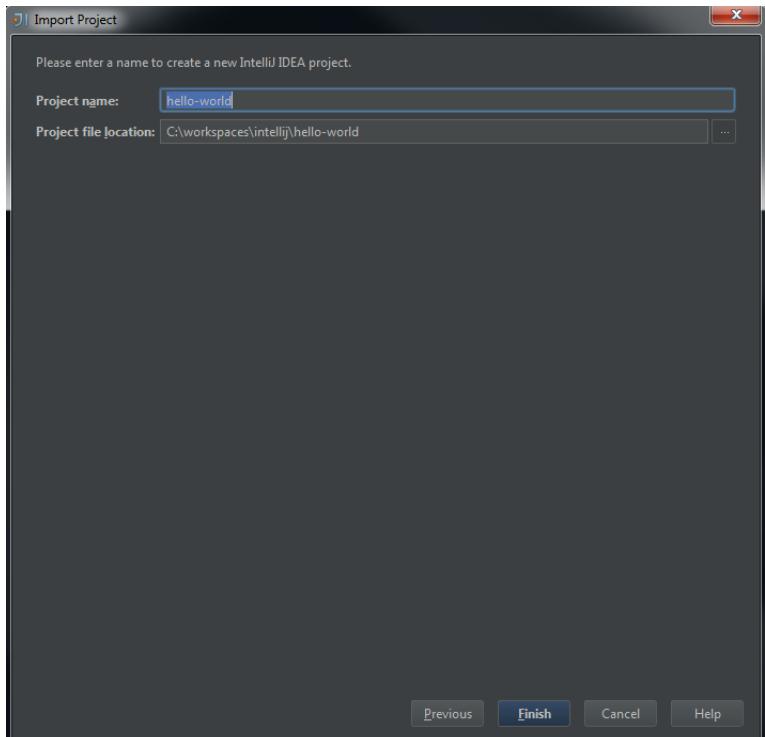
Click + sign, Add New JDK ->JDK



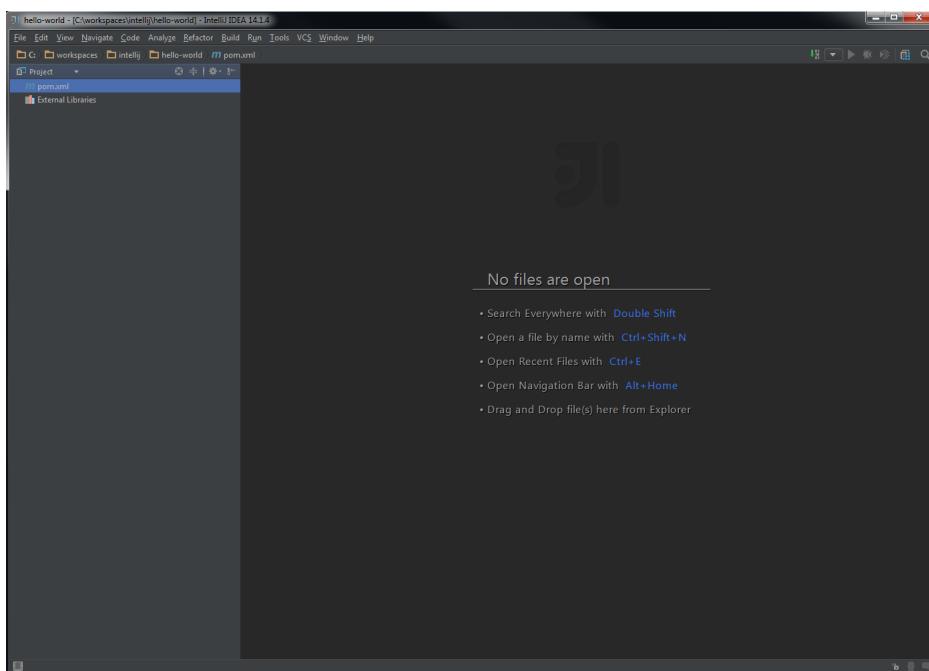
Select C:\opt\Java\jdk1.7.0_79, click OK



Click Next



Click Finish



A.3.3 Configure Tomcat in IntelliJ

- Configure Tomcat in your Maven project pom.xml

Since IntelliJ Community Edition does not have build in Application Server, we need to add maven tomcat plugin

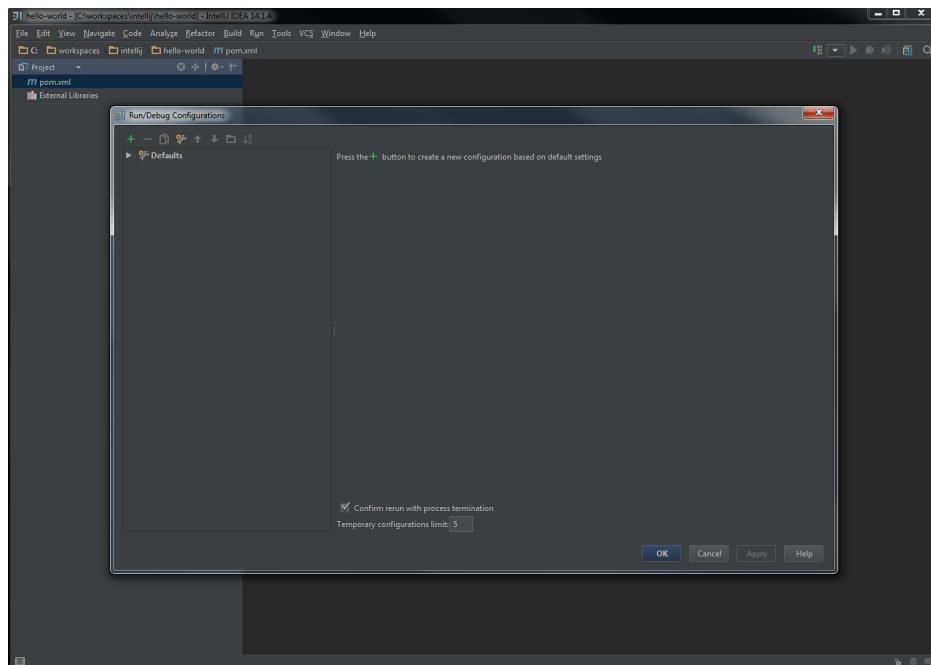
```

1 <project xmlns="http://maven.apache.org/POM/4.0.0"
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
4     http://maven.apache.org/maven-v4_0_0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>com.xyz</groupId>
8   <artifactId>hello-world</artifactId>
9   <packaging>war</packaging>
10  <version>1.0-SNAPSHOT</version>
11  <name>hello-world Maven Webapp</name>
12  <url>http://maven.apache.org</url>
13
14  <properties>
15    <junit.version>4.12</junit.version>
16    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
17      <tomcat7.maven.plugin>2.2</tomcat7.maven.plugin>
18  </properties>
19
20  <dependencies>
21    <dependency>
22      <groupId>junit</groupId>
23      <artifactId>junit</artifactId>
24      <version>${junit.version}</version>
25      <scope>test</scope>
26    </dependency>
27  </dependencies>
28
29  <build>
30    <finalName>hello-world</finalName>
31    <plugins>
32      <plugin>
33        <groupId>org.apache.tomcat.maven</groupId>
34        <artifactId>tomcat7-maven-plugin</artifactId>
35        <version>${tomcat7.maven.plugin}</version>
36        <configuration>
37          <url>http://localhost:8080/manager/text</url>
38        </configuration>
39      </plugin>
40    </plugins>
41  </build>
42</project>
```

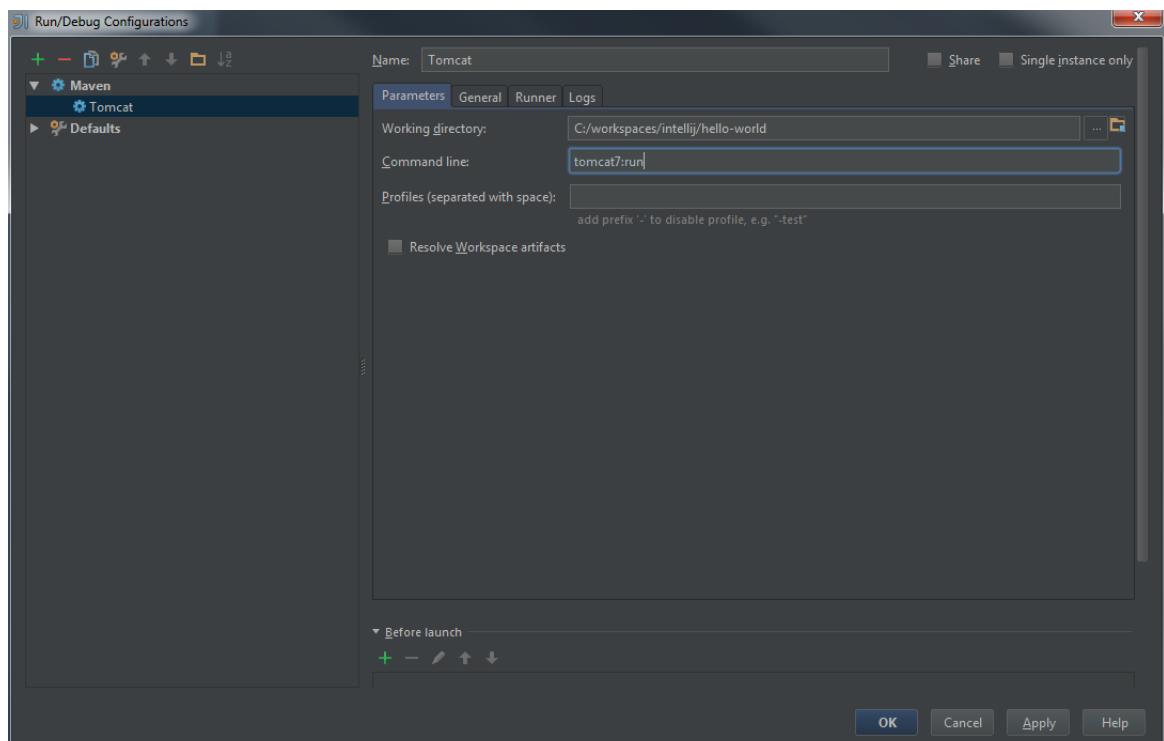
- Configure Tomcat in IntelliJ IDE

Make sure that we do not have jsp-api, el-api or jboss-el dependencies in our pom file as tomcat has these libraries already in its /lib directory.

Go to Run menu and click Edit Configurations, click + icon at the top left and select Maven as depicted in the following diagram.

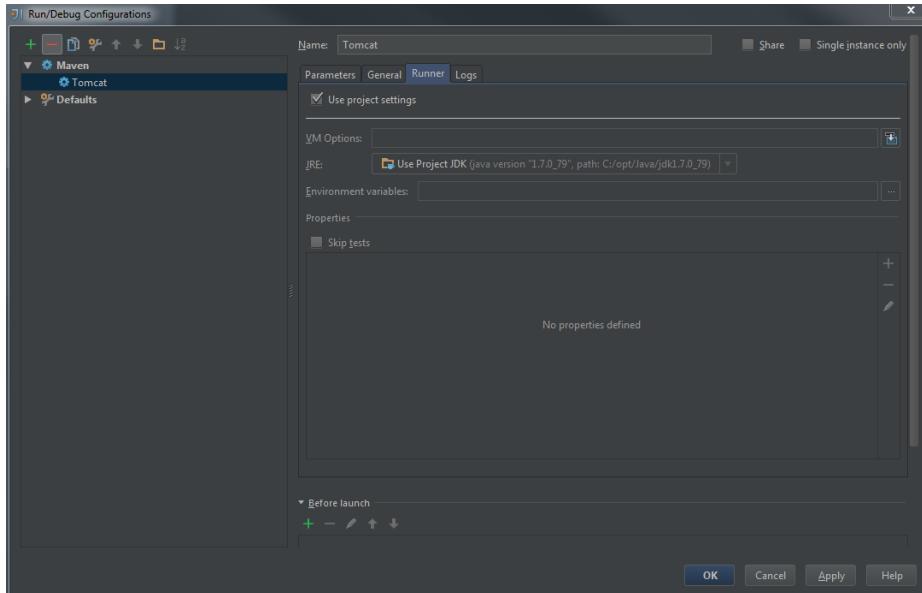


Now in the first Parameter tab enter Tomcat in the name field to recognize these settings. In the Command Line enter tomcat7:run as depicted in the following diagram.



Click Runner tab and provide appropriate VM options as per your need and select ap-

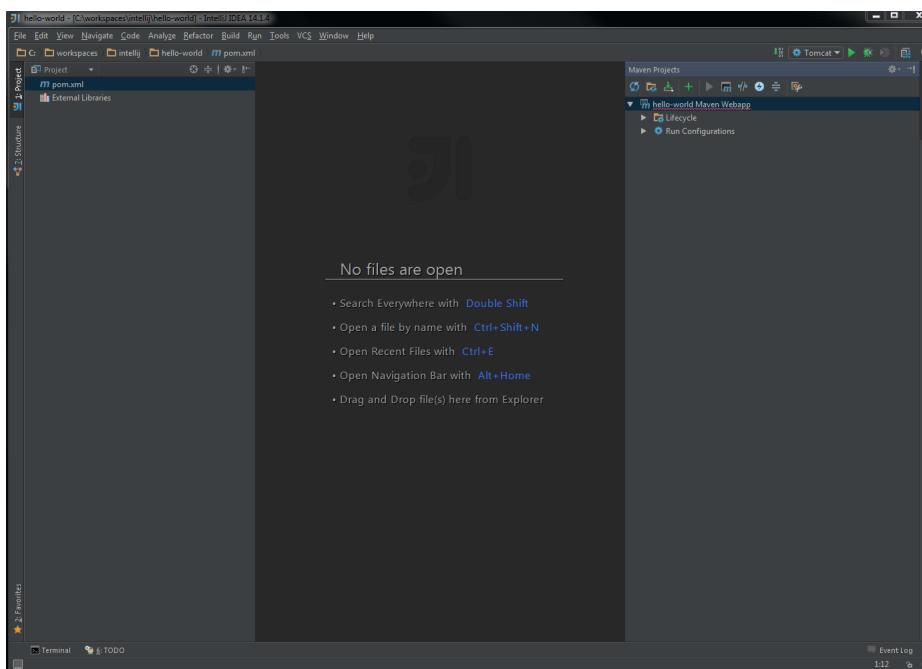
appropriate JRE.



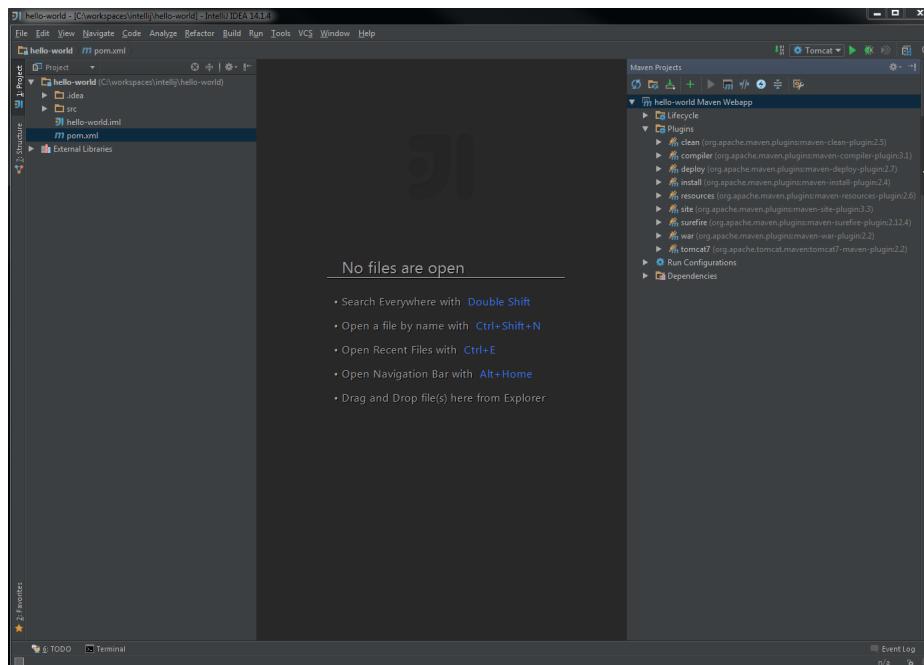
Click OK.

Now click Maven Projects from right pane as shown in the following diagram and select tomcat7 from plugins and click reimport icon to update everything.

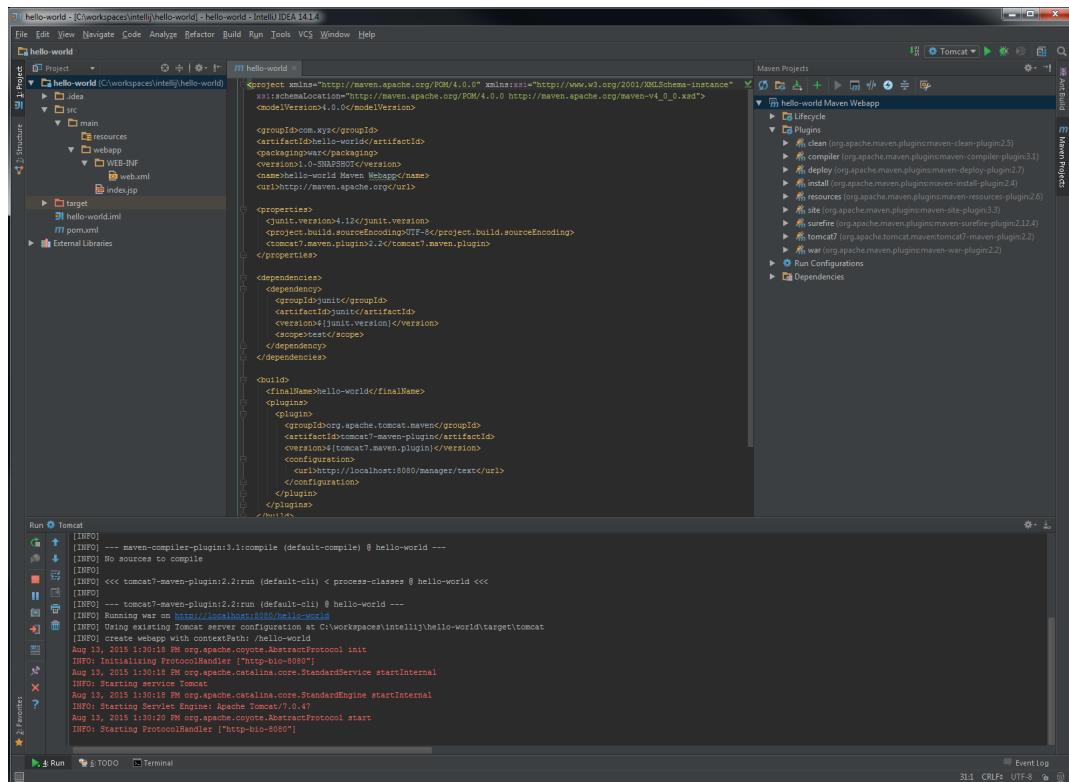
If Maven Projects not showing on the right side, click the icon on the left bottom side.



If Plugins does not show up, click the refresh icon: Reimport All Maven Projects

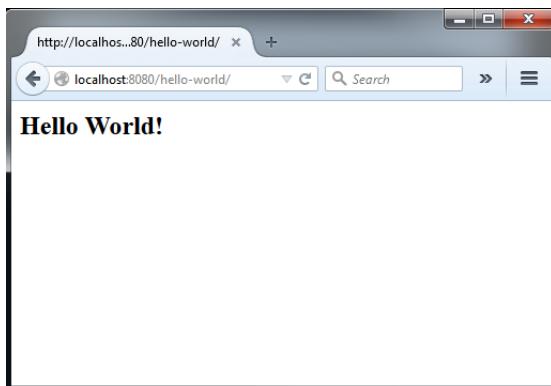


If Plugins does not show up, click the refresh icon: Reimport All Maven Projects
Finally click green Run button to start tomcat



A.3.4 hello-world

- Test helloworld in your browser
<http://localhost:8080/helloworld/>



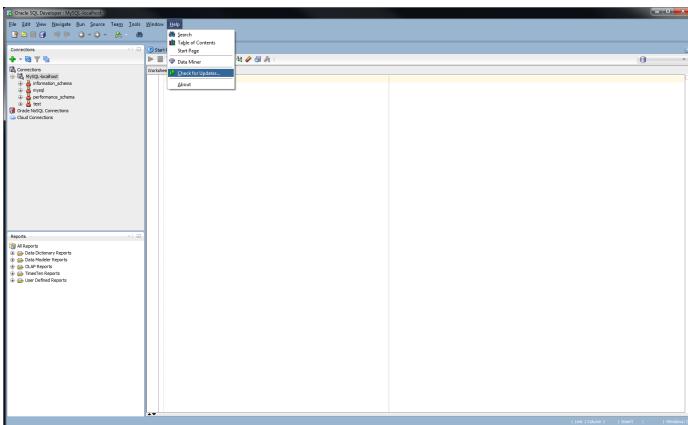
APPENDIX B

SQL DEVELOPER

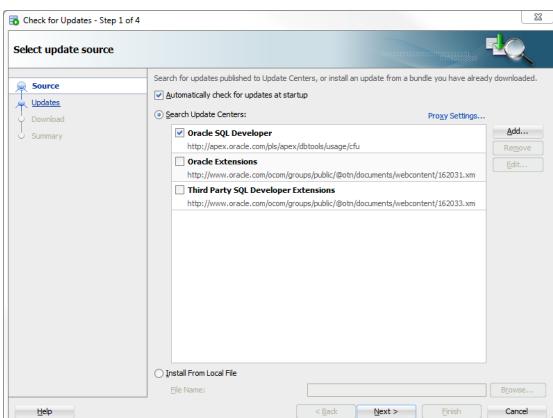
B.1 Download Show Me Password

- The "Show me password" is a simple extension for Oracle SQL Developer or Oracle JDeveloper i.e. simple extension (tool) that decrypts all saved (encrypted) password for database connections, application server connection, servers etc. in SQL Developer or JDeveloper.
- Download the latest version of the extension for Oracle SQL Developer 4.x (ShowMePasswordSQLDeveloper4_v1.0.0.zip), to your machine (the extension will be packaged as a zip file - pls. don't unzip the extension).
<http://show-me-password.tomecode.com/>
http://show-me-password-sql-developer.googlecode.com/files>ShowMePasswordSQLDeveloper4_v1.0.0.zip

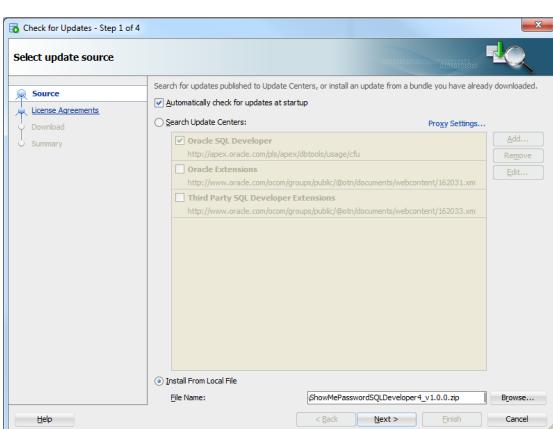
B.2 Install Show Me Password



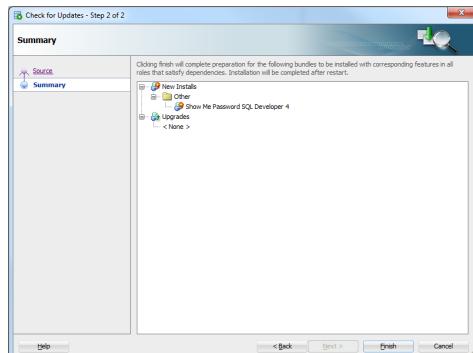
Select Help –>Check for Updates...



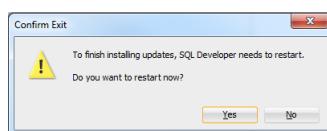
Select Install From Local File



Click Browse..., and point to the zip file you have downloaded.

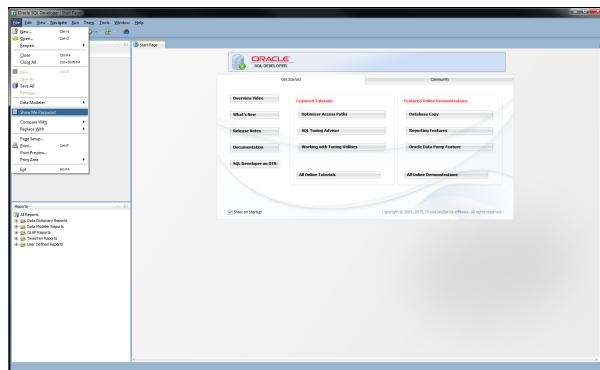


Click Finish

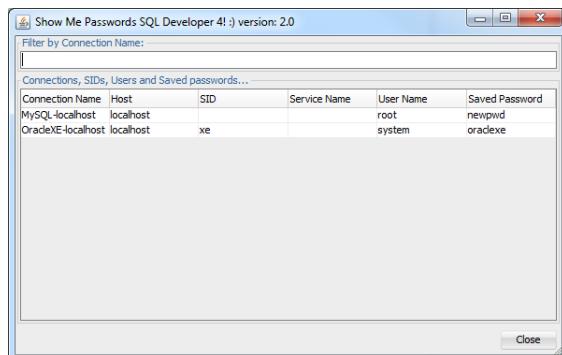


Click Yes to restart SQL Developer to complete the extension installation

B.3 Use Show Me Password



Select File ->Show Me Password



APPENDIX C

VIRTUAL MACHINE - ORACLE RESPONSE FILES

C.1 db_install.rsp

```
1 [root@localhost ~]# vi /tmp/database/response/db_install.rsp
2 ##### Copyright (c) Oracle Corporation 1998,2014. All rights reserved.##
3 ## Specify values for the variables listed below to customize
4 ## your installation.
5 ## Each variable is associated with a comment. The comment
6 ## can help to populate the variables with the appropriate
7 ## values.
8 ## IMPORTANT NOTE: This file contains plain text passwords and
9 ## should be secured to have read permission only by oracle user
10 ## or db administrator who owns this installation.
11 ##
12 ##
13 ##
14 ##
15 ##
16 #####
17
18 -----
19 # Do not change the following system generated value.
20 -----
21 #
22 oracle.install.responseFileVersion=/oracle/install/rspfmt_dbinstall_response_schema_v12.1.0
23 -----
24 #
25 # Specify the installation option.
```

```

26  # It can be one of the following:
27  #   - INSTALL_DB_SWONLY
28  #   - INSTALL_DB_AND_CONFIG
29  #   - UPGRADE_DB
30  #-----
31 #oracle.install.option=
32 oracle.install.option=INSTALL_DB_SWONLY
33
34 #-----
35 # Specify the hostname of the system as set during the install. It can be used
36 # to force the installation to use an alternative hostname rather than using the
37 # first hostname found on the system. (e.g., for systems with multiple hostnames
38 # and network interfaces)
39 #-----
40 ORACLE_HOSTNAME=
41
42 #-----
43 # Specify the Unix group to be set for the inventory directory.
44 #-----
45 #UNIX_GROUP_NAME=
46 UNIX_GROUP_NAME=oinstall
47
48 #-----
49 # Specify the location which holds the inventory files.
50 # This is an optional parameter if installing on
51 # Windows based Operating System.
52 #-----
53 #INVENTORY_LOCATION=
54 INVENTORY_LOCATION=orcl
55
56 #-----
57 # Specify the languages in which the components will be installed.
58 #
59 # en   : English          ja   : Japanese
60 # fr   : French           ko   : Korean
61 # ar   : Arabic            es   : Latin American Spanish
62 # bn   : Bengali          lv   : Latvian
63 # pt_BR: Brazilian Portuguese lt   : Lithuanian
64 # bg   : Bulgarian         ms   : Malay
65 # fr_CA: Canadian French es_MX: Mexican Spanish
66 # ca   : Catalan           no   : Norwegian
67 # hr   : Croatian          pl   : Polish
68 # cs   : Czech              pt   : Portuguese
69 # da   : Danish             ro   : Romanian
70 # nl   : Dutch              ru   : Russian
71 # ar_EG: Egyptian         zh_CN: Simplified Chinese
72 # en_GB: English (Great Britain) sk   : Slovak
73 # et   : Estonian          sl   : Slovenian
74 # fi   : Finnish            es_ES: Spanish
75 # de   : German             sv   : Swedish
76 # el   : Greek              th   : Thai
77 # iw   : Hebrew             zh_TW: Traditional Chinese
78 # hu   : Hungarian          tr   : Turkish
79 # is   : Icelandic          uk   : Ukrainian
80 # in   : Indonesian         vi   : Vietnamese
81 # it   : Italian
82 #
83 # all_langs   : All languages
84 #
85 # Specify value as the following to select any of the languages.
86 # Example : SELECTED_LANGUAGES=en,fr,ja
87 #
88 # Specify value as the following to select all the languages.
89 # Example : SELECTED_LANGUAGES=all_langs
90 #-----
91 SELECTED_LANGUAGES=en
92
93 #-----
94 # Specify the complete path of the Oracle Home.
95 #-----
96 #ORACLE_HOME=
97 ORACLE_HOME=/u01/app/oracle/product/12.1.0.2/db_1
98

```

```

99  -----
100 # Specify the complete path of the Oracle Base.
101 #
102 #ORACLE_BASE=
103 ORACLE_BASE=/u01/app/oracle
104
105 -----
106 # Specify the installation edition of the component.
107 #
108 # The value should contain only one of these choices.
109 # - EE      : Enterprise Edition
110
111 -----
112 #oracle.install.db.InstallEdition=
113 oracle.install.db.InstallEdition=EE
114
115 ######
116 #
117 # PRIVILEGED OPERATING SYSTEM GROUPS
118 #
119 # Provide values for the OS groups to which OSDBA and OSOPER privileges
120 # needs to be granted. If the install is being performed as a member of the
121 # group "dba", then that will be used unless specified otherwise below.
122 #
123 # The value to be specified for OSDBA and OSOPER group is only for UNIX based
124 # Operating System.
125 #
126 #####
127
128 -----
129 # The DBA_GROUP is the OS group which is to be granted OSDBA privileges.
130 #
131 #oracle.install.db.DBA_GROUP=
132 oracle.install.db.DBA_GROUP=dba
133
134 -----
135 # The OPER_GROUP is the OS group which is to be granted OSOPER privileges.
136 # The value to be specified for OSOPER group is optional.
137 #
138 #oracle.install.db.OPER_GROUP=
139 oracle.install.db.OPER_GROUP=dba
140
141 -----
142 # The BACKUPDBA_GROUP is the OS group which is to be granted OSBACKUPDBA
143 #     privileges.
144 #
145 #oracle.install.db.BACKUPDBA_GROUP=
146 oracle.install.db.BACKUPDBA_GROUP=dba
147
148 # The DGDBA_GROUP is the OS group which is to be granted OSDGDBA privileges.
149 #
150 #oracle.install.db.DGDBA_GROUP=
151 oracle.install.db.DGDBA_GROUP=dba
152
153 -----
154 # The KMDBA_GROUP is the OS group which is to be granted OSKMDBA privileges.
155 #
156 #oracle.install.db.KMDBA_GROUP=
157 oracle.install.db.KMDBA_GROUP=dba
158
159 #####
160 #
161 #             Grid Options
162 #
163 #####
164
165 # Specify the type of Real Application Cluster Database
166 #
167 # - ADMIN_MANAGED: Admin-Managed
168 # - POLICY_MANAGED: Policy-Managed
169 #
170 # If left unspecified, default will be ADMIN_MANAGED

```

452 VIRTUAL MACHINE - ORACLE RESPONSE FILES

```
171 #-----  
172 oracle.install.db.rac.configurationType=  
173 #-----  
174 # Value is required only if RAC database type is ADMIN_MANAGED  
175 #  
176 # Specify the cluster node names selected during the installation.  
177 # Leaving it blank will result in install on local server only (Single Instance)  
178 #  
179 # Example : oracle.install.db.CLUSTER_NODES=node1,node2  
180 #-----  
181 oracle.install.db.CLUSTER_NODES=  
182 #-----  
183 #-----  
184 # This variable is used to enable or disable RAC One Node install.  
185 #  
186 #     - true   : Value of RAC One Node service name is used.  
187 #     - false  : Value of RAC One Node service name is not used.  
188 #  
189 # If left blank, it will be assumed to be false.  
190 #-----  
191 oracle.install.db.isRACOneInstall=  
192 #-----  
193 #-----  
194 # Value is required only if oracle.install.db.isRACOneInstall is true.  
195 #  
196 # Specify the name for RAC One Node Service  
197 #-----  
198 oracle.install.db.racOneServiceName=  
199 #-----  
200 #-----  
201 # Value is required only if RAC database type is POLICY_MANAGED  
202 #  
203 # Specify a name for the new Server pool that will be configured  
204 # Example : oracle.install.db.rac.serverpoolName=pool1  
205 #-----  
206 oracle.install.db.rac.serverpoolName=  
207 #-----  
208 #-----  
209 # Value is required only if RAC database type is POLICY_MANAGED  
210 #  
211 # Specify a number as cardinality for the new Server pool that will be  
212 #     → configured  
213 # Example : oracle.install.db.rac.serverpoolCardinality=2  
214 #-----  
215 oracle.install.db.rac.serverpoolCardinality=  
216 #####  
217 #-----  
218 #  
219 #             Database Configuration Options  
220 #-----  
221 #####  
222 #-----  
223 #-----  
224 # Specify the type of database to create.  
225 # It can be one of the following:  
226 #     - GENERAL_PURPOSE  
227 #     - DATA_WAREHOUSE  
228 # GENERAL_PURPOSE: A starter database designed for general purpose use or  
229 #     → transaction-heavy applications.  
230 # DATA_WAREHOUSE : A starter database optimized for data warehousing  
231 #     → applications.  
232 #-----  
233 oracle.install.db.config.starterdb.type=GENERAL_PURPOSE  
234 #-----  
235 # Specify the Starter Database Global Database Name.  
236 #-----  
237 oracle.install.db.config.starterdb.globalDBName=  
238 #-----  
239 # Specify the Starter Database SID.  
240 #-----
```

```

241 oracle.install.db.config.starterdb.SID=
242 -----
243 # Specify whether the database should be configured as a Container database.
244 # The value can be either "true" or "false". If left blank it will be assumed
245 # to be "false".
246 -----
247 oracle.install.db.ConfigureAsContainerDB=
248 -----
249 # Specify the Pluggable Database name for the pluggable database in Container
250 # Database.
251 -----
252 oracle.install.db.config.PDBName=
253 -----
254 # Specify the Starter Database character set.
255 #
256 # One of the following
257 # AL32UTF8, WE8ISO8859P15, WE8MSWIN1252, EE8ISO8859P2,
258 # EE8MSWIN1250, NE8ISO8859P10, NEE8ISO8859P4, BLT8MSWIN1257,
259 # BLT8ISO8859P13, CL8ISO8859P5, CL8MSWIN1251, AR8ISO8859P6,
260 # AR8MSWIN1256, EL8ISO8859P7, EL8MSWIN1253, IW8ISO8859P8,
261 # IW8MSWIN1255, JA16EUC, JA16EUCTILDE, JA16SJIS, JA16SJISTILDE,
262 # KO16MSWIN949, ZHS16GBK, TH8TISASCII, ZHT32EUC, ZHT16MSWIN950,
263 # ZHT16HKSCS, WE8ISO8859P9, TR8MSWIN1254, VN8MSWIN1258
264 -----
265 #oracle.install.db.config.starterdb.characterSet=
266 oracle.install.db.config.starterdb.characterSet=AL32UTF8
267 -----
268 # This variable should be set to true if Automatic Memory Management
269 # in Database is desired.
270 # If Automatic Memory Management is not desired, and memory allocation
271 # is to be done manually, then set it to false.
272 -----
273 #oracle.install.db.config.starterdb.memoryOption=
274 oracle.install.db.config.starterdb.memoryOption=true
275 -----
276 # Specify the total memory allocation for the database. Value(in MB) should be
277 # at least 256 MB, and should not exceed the total physical memory available
278 # on the system.
279 # Example: oracle.install.db.config.starterdb.memoryLimit=512
280 -----
281 oracle.install.db.config.starterdb.memoryLimit=
282 -----
283 # This variable controls whether to load Example Schemas onto
284 # the starter database or not.
285 # The value can be either "true" or "false". If left blank it will be assumed
286 # to be "false".
287 -----
288 #oracle.install.db.config.starterdb.installExampleSchemas=
289 oracle.install.db.config.starterdb.installExampleSchemas=false
290 -----
291 ######
292 #
293 # Passwords can be supplied for the following four schemas in the
294 # starter database:
295 #   SYS
296 #   SYSTEM
297 #   DBSNMP (used by Enterprise Manager)
298 #   Same password can be used for all accounts (not recommended)
299 #   or different passwords for each account can be provided (recommended)
300 #
301 #
302 #
303 #
304 #
305 #
306 #
307 #####
308 -----
309 # This variable holds the password that is to be used for all schemas in the
310 # starter database.

```

454 VIRTUAL MACHINE - ORACLE RESPONSE FILES

```
312 #-----
313 oracle.install.db.config.starterdb.password.ALL=
314 #
315 #-----
316 # Specify the SYS password for the starter database.
317 #-----
318 oracle.install.db.config.starterdb.password.SYS=
319 #
320 #-----
321 # Specify the SYSTEM password for the starter database.
322 #-----
323 oracle.install.db.config.starterdb.password.SYSTEM=
324 #
325 #-----
326 # Specify the DBSNMP password for the starter database.
327 #-----
328 oracle.install.db.config.starterdb.password.DBSNMP=
329 #
330 #-----
331 # Specify the PDBADMIN password required for creation of Pluggable Database in
332 #   the Container Database.
333 #-----
334 oracle.install.db.config.starterdb.password.PDBADMIN=
335 #
336 #-----
337 # Specify the management option to use for managing the database.
338 # Options are:
339 # 1. CLOUD_CONTROL - If you want to manage your database with Enterprise Manager
340 #   Cloud Control along with Database Express.
341 # 2. DEFAULT -If you want to manage your database using the default Database
342 #   Express option.
343 #-----
344 #oracle.install.db.config.starterdb.managementOption=
345 oracle.install.db.config.starterdb.managementOption=DEFAULT
346 #
347 #-----
348 # Specify the OMS host to connect to Cloud Control.
349 # Applicable only when
350 #   oracle.install.db.config.starterdb.managementOption=CLOUD_CONTROL
351 #-----
352 oracle.install.db.config.starterdb.omsHost=
353 #
354 #-----
355 # Specify the OMS port to connect to Cloud Control.
356 # Applicable only when
357 #   oracle.install.db.config.starterdb.managementOption=CLOUD_CONTROL
358 #-----
359 oracle.install.db.config.starterdb.omsPort=
360 #
361 #-----
362 # Specify the EM Admin user name to use to connect to Cloud Control.
363 # Applicable only when
364 #   oracle.install.db.config.starterdb.managementOption=CLOUD_CONTROL
365 #-----
366 oracle.install.db.config.starterdb.emAdminUser=
367 #
368 ######
369 #
370 # SPECIFY RECOVERY OPTIONS
371 #   #
372 # -----
373 # Recovery options for the database can be mentioned using the entries below #
374 #   #
375 ######
```

```

376 #-----
377 # This variable is to be set to false if database recovery is not required. Else
378 # this can be set to true.
379 #-----
380 oracle.install.db.config.starterdb.enableRecovery=
381
382 #-----
383 # Specify the type of storage to use for the database.
384 # It can be one of the following:
385 #   - FILE_SYSTEM_STORAGE
386 #   - ASM_STORAGE
387 #-----
388 #oracle.install.db.config.starterdb.storageType=
389 oracle.install.db.config.starterdb.storageType=FILE_SYSTEM_STORAGE
390
391 #-----
392 # Specify the database file location which is a directory for datafiles, control
393 # files, redo logs.
394 #
395 # Applicable only when
396 #   → oracle.install.db.config.starterdb.storage=FILE_SYSTEM_STORAGE
397 #-----
398 oracle.install.db.config.starterdb.fileSystemStorage.dataLocation=
399
400 #-----
401 # Specify the recovery location.
402 #
403 # Applicable only when
404 #   → oracle.install.db.config.starterdb.storage=FILE_SYSTEM_STORAGE
405 #-----
406 oracle.install.db.config.starterdb.fileSystemStorage.recoveryLocation=
407
408 #-----
409 # Specify the existing ASM disk groups to be used for storage.
410 #
411 # Applicable only when
412 #   → oracle.install.db.config.starterdb.storageType=ASM_STORAGE
413 #-----
414 oracle.install.db.config.asm.diskGroup=
415
416 #-----
417 # Specify the password for ASMSNMP user of the ASM instance.
418 #
419 # Applicable only when oracle.install.db.config.starterdb.storage=ASM_STORAGE
420 #-----
421 oracle.install.db.config.asm.ASMSNMPPassword=
422
423 #-----
424 # Specify the My Oracle Support Account Username.
425 #
426 # Example : MYORACLESUPPORT_USERNAME=abc@oracle.com
427 #-----
428 MYORACLESUPPORT_USERNAME=
429
430 #-----
431 # Specify the My Oracle Support Account Username password.
432 #
433 # Example : MYORACLESUPPORT_PASSWORD=password
434 #-----
435 MYORACLESUPPORT_PASSWORD=
436
437 #-----
438 # Specify whether to enable the user to set the password for
439 # My Oracle Support credentials. The value can be either true or false.
440 # If left blank it will be assumed to be false.
441 #
442 # Example : SECURITY_UPDATES_VIA_MYORACLESUPPORT=true
443 #-----
444 SECURITY_UPDATES_VIA_MYORACLESUPPORT=
445
446 #-----
447 # Specify whether user doesn't want to configure Security Updates.
448 # The value for this variable should be true if you don't want to configure

```

456 VIRTUAL MACHINE - ORACLE RESPONSE FILES

```
446 # Security Updates, false otherwise.  
447 #  
448 # The value can be either true or false. If left blank it will be assumed  
449 # to be false.  
450 #  
451 # Example : DECLINE_SECURITY_UPDATES=false  
452 #-----  
453 #DECLINE_SECURITY_UPDATES=  
454 DECLINE_SECURITY_UPDATES=true  
455  
456 #-----  
457 # Specify the Proxy server name. Length should be greater than zero.  
458 #  
459 # Example : PROXY_HOST=proxy.domain.com  
460 #-----  
461 PROXY_HOST=  
462  
463 #-----  
464 # Specify the proxy port number. Should be Numeric and at least 2 chars.  
465 #  
466 # Example : PROXY_PORT=25  
467 #-----  
468 PROXY_PORT=  
469  
470 #-----  
471 # Specify the proxy user name. Leave PROXY_USER and PROXY_PWD  
472 # blank if your proxy server requires no authentication.  
473 #  
474 # Example : PROXY_USER=username  
475 #-----  
476 PROXY_USER=  
477  
478 #-----  
479 # Specify the proxy password. Leave PROXY_USER and PROXY_PWD  
480 # blank if your proxy server requires no authentication.  
481 #  
482 # Example : PROXY_PWD=password  
483 #-----  
484 PROXY_PWD=  
485  
486 #-----  
487 # Specify the Oracle Support Hub URL.  
488 #  
489 # Example : COLLECTOR_SUPPORTHUB_URL=https://orasupporthub.company.com:8080/  
490 #-----  
491 COLLECTOR_SUPPORTHUB_URL=  
492  
493 [root@localhost ~] #
```

C.2 dbca.rsp

```
1 [root@localhost ~]# vi /tmp/database/response/dbca.rsp  
2 #####  
3 ##  
4 ## DBCA response file  
5 ## -----  
6 ## Copyright 1998, 2014, Oracle Corporation. All Rights Reserved.  
7 ##  
8 ## Specify values for the variables listed below to customize Oracle  
9 ## Database Configuration installation.  
10##  
11## Each variable is associated with a comment. The comment identifies the  
12## variable type.  
13##  
14## Please specify the values in the following format :  
15## Type : Example  
16## String : "<value>"  
17## Boolean : True or False  
18## Number : <numeric value>
```

```

19      ##      StringList : {"<value1>","<value2>"}          ##
20      ##
21  ## Examples :
22  ##   1. dbca -progress_only -responseFile <response file>      ##
23  ##       Display a progress bar depicting progress of database creation  ##
24  ##       process.                                                 ##
25  ##
26  ##   2. dbca -silent -responseFile <response file>           ##
27  ##       Creates database silently. No user interface is displayed.    ##
28  ##
29  ##   3. dbca -silent -createDatabase -cloneTemplate             ##
30  ##                   -responseFile <response file>                  ##
31  ##       Creates database silently with clone template. The template in  ##
32  ##       responsefile is a clone template.                           ##
33  ##   ↳  ##
34  ##   4. dbca -silent -deleteDatabase -responseFile <response file>  ##
35  ##       Deletes database silently.                                ##
36 ######
37
38 #-----#
39 # GENERAL section is required for all types of database creations.
40 #-----#
41 [GENERAL]
42
43 #-----
44 # Name          : RESPONSEFILE_VERSION
45 # Datatype      : String
46 # Description   : Version of the database to create
47 # Valid values  : "12.1.0"
48 # Default value : None
49 # Mandatory    : Yes
50 #
51 RESPONSEFILE_VERSION = "12.1.0"
52
53 #-----
54 # Name          : OPERATION_TYPE
55 # Datatype      : String
56 # Description   : Type of operation
57 # Valid values  : "createDatabase" \ "createTemplateFromDB" \
58 #                   ↳ "createCloneTemplate" \ "deleteDatabase" \ "configureDatabase" \
59 #                   ↳ "addInstance" (RAC-only) \ "deleteInstance" (RAC-only) \
60 #                   ↳ "createPluggableDatabase" \ "unplugDatabase" \ "deletePluggableDatabase" \
61 #                   ↳ "configurePluggableDatabase"
62 # Default value : None
63 # Mandatory    : Yes
64 #-----#
65 #-----#
66 # CREATEDATABASE section is used when OPERATION_TYPE is defined as
67 #   ↳ "createDatabase".
68 #-----#
69 [CREATEDATABASE]
70
71 #-----
72 # Name          : GDBNAME
73 # Datatype      : String
74 # Description   : Global database name of the database
75 # Valid values  : <db_name>.<db_domain> - when database domain isn't NULL
76 #                   <db_name>                - when database domain is NULL
77 # Default value : None
78 # Mandatory    : Yes
79 #
80 #GDBNAME = "orcl12c.us.oracle.com"
81 GDBNAME = "orcl"
82
83 #-----
84 # Name          : DATABASECONFTYPE
85 # Datatype      : String

```

458 VIRTUAL MACHINE - ORACLE RESPONSE FILES

```
85 # Description : database conf type as Single Instance, Real Application
86 #           ↳ Cluster or Real Application Cluster One Nodes database
86 # Valid values : SI\RAC\RACONENODE
87 # Default value : SI
88 # Mandatory : No
89 #-----
90 #DATABASECONFTYPE = "SI"
91
92 #-----
93 # Name : RACONENODESERVICENAME
94 # Datatype : String
95 # Description : Service is required by application to connect to RAC One
96 #           Node Database
97 # Valid values : Service Name
98 # Default value : None
99 # Mandatory : No [required in case DATABASECONFTYPE is set to RACONENODE ]
100 #-----
101 #RACONENODESERVICENAME =
102
103 #-----
104 # Name : POLICYMANAGED
105 # Datatype : Boolean
106 # Description : Set to true if Database is policy managed and
107 #           set to false if Database is admin managed
108 # Valid values : TRUE\FALSE
109 # Default value : FALSE
110 # Mandatory : No
111 #-----
112 #POLICYMANAGED = "false"
113
114 #-----
115 # Name : CREATESERVERPOOL
116 # Datatype : Boolean
117 # Description : Set to true if new server pool need to be created for database
118 #           if this option is specified then the newly created database
119 #           will use this newly created serverpool.
120 #           Multiple serverpoolname can not be specified for database
121 # Valid values : TRUE\FALSE
122 # Default value : FALSE
123 # Mandatory : No
124 #-----
125 #CREATESERVERPOOL = "false"
126
127 #-----
128 # Name : SERVERPOOLNAME
129 # Datatype : String
130 # Description : Only one serverpool name need to be specified
131 #           if Create Server Pool option is specified.
132 #           Comma-separated list of Serverpool names if db need to use
133 #           multiple Server pool
134 # Valid values : ServerPool name
135
136 # Default value : None
137 # Mandatory : No [required in case of RAC service centric database]
138 #-----
139 #SERVERPOOLNAME =
140
141 #-----
142 # Name : CARDINALITY
143 # Datatype : Number
144 # Description : Specify Cardinality for create server pool operation
145
146 # Valid values : any positive Integer value
147 # Default value : Number of qualified nodes on cluster
148 # Mandatory : No [Required when a new serverpool need to be created]
149 #-----
150 #CARDINALITY =
151
152
153
154
155 #-----
156 # Name : FORCE
```

```

157 # Datatype      : Boolean
158 # Description   : Set to true if new server pool need to be created by force
159 #                   if this option is specified then the newly created serverpool
160 #                   will be assigned server even if no free servers are
161 #                   available.
162 #                   This may affect already running database.
163 #                   This flag can be specified for Admin managed as well as
164 #                   policy managed db.
165 # Valid values   : TRUE\FALSE
166 # Default value  : FALSE
167 # Mandatory     : No
168 #-----
169 #FORCE = "false"
170
171 #-----
172 # Name          : PQPOOLNAME
173 # Datatype      : String
174 # Description   : Only one serverpool name needs to be specified
175 #                   if create server pool option is specified.
176 #                   Comma-separated list of serverpool names if use
177 #                   server pool. This is required to
178 #                   create Parallel Query (PQ) database. Applicable to Big
179 #                   Cluster
180 # Valid values   : Parallel Query (PQ) pool name
181 # Default value  : None
182 # Mandatory     : No [required in case of RAC service centric database]
183 #PQPOOLNAME =
184
185 #-----
186 # Name          : PQCARDINALITY
187 # Datatype      : Number
188 # Description   : Specify Cardinality for create server pool operation.
189 #                   Applicable to Big Cluster
190 # Valid values   : any positive Integer value
191 # Default value  : Number of qualified nodes on cluster
192 # Mandatory     : No [Required when a new serverpool need to be created]
193 #PQCARDINALITY =
194
195
196
197
198
199 #-----
200 # Name          : SID
201 # Datatype      : String
202 # Description   : System identifier (SID) of the database
203 # Valid values   : Check Oracle12c Administrator's Guide
204 # Default value  : <db_name> specified in GDBNAME
205 # Mandatory     : No
206 #-----
207 #SID = "orcl112c"
208 SID = "orcl"
209
210 #-----
211 # Name          : CREATEASCONTAINERDATABASE
212 # Datatype      : boolean
213 # Description   : flag to create database as container database
214 # Valid values   : Check Oracle12c Administrator's Guide
215 # Default value  : false
216 # Mandatory     : No
217 #-----
218 #CREATEASCONTAINERDATABASE =
219
220 #-----
221 # Name          : NUMBEROFPDBS
222 # Datatype      : Number
223 # Description   : Specify the number of pdb to be created
224 # Valid values   : 0 to 252
225 # Default value  : 0
226 # Mandatory     : No

```

```

227 #-----
228 #NUMBEROFPDBS =
229 #
230 #-----
231 # Name : PDBNAME
232 # Datatype : String
233 # Description : Specify the dbname/pdbname prefix if one or more pdb need to
234 #                be created
235 # Valid values : Check Oracle12c Administrator's Guide
236 # Default value : None
237 # Mandatory : No
238 #-----
239 #PDBNAME =
240 #
241 #-----
242 # Name : PDBADMINPASSWORD
243 # Datatype : String
244 # Description : PDB Administrator user password
245 # Valid values : Check Oracle12c Administrator's Guide
246 # Default value : None
247 # Mandatory : No
248 #-----
249 # PDBADMINPASSWORD = ""
250 #
251 #-----
252 # Name : NODELIST
253 # Datatype : String
254 # Description : Comma-separated list of cluster nodes
255 # Valid values : Cluster node names
256 # Default value : None
257 # Mandatory : No (Yes for RAC database-centric database )
258 #-----
259 #NODELIST=
260 #
261 #-----
262 # Name : TEMPLATENAME
263 # Datatype : String
264 # Description : Name of the template
265 # Valid values : Template file name
266 # Default value : None
267 # Mandatory : Yes
268 #-----
269 # TEMPLATENAME = "General_Purpose.dbc"
270 #
271 #-----
272 # Name : OBFUSCATEDPASSWORDS
273 # Datatype : Boolean
274 # Description : Set to true if passwords are encrypted
275 # Valid values : TRUE\FALSE
276 # Default value : FALSE
277 # Mandatory : No
278 #-----
279 #OBFUSCATEDPASSWORDS = FALSE
280 #
281 #-----
282 # Name : SYSPASSWORD
283 # Datatype : String
284 # Description : Password for SYS user
285 # Valid values : Check Oracle12c Administrator's Guide
286 # Default value : None
287 # Mandatory : Yes
288 #-----
289 #SYSPASSWORD = "password"
290 SYSPASSWORD = "password"
291 #
292 #-----
293 # Name : SYSTEMPASSWORD
294 # Datatype : String
295 # Description : Password for SYSTEM user
296 # Valid values : Check Oracle12c Administrator's Guide
297 # Default value : None
298 # Mandatory : Yes

```

```

299 #-----
300 #SYSTEMPASSWORD = "password"
301 SYSTEMPASSWORD = "password"
302
303 #-----
304 # Name : SERVICEUSERPASSWORD
305 # Datatype : String
306 # Description : Password for Windows Service user
307 # Default value : None
308 # Mandatory : If Oracle home is installed with windows service user
309 #
310 #SERVICEUSERPASSWORD = "password"
311
312 #-----
313 # Name : EMCONFIGURATION
314 # Datatype : String
315 # Description : Enterprise Manager Configuration Type
316 # Valid values : CENTRAL|DBEXPRESS|ALL|NONE
317 # Default value : NONE
318 # Mandatory : No
319 #
320 #EMCONFIGURATION = "NONE"
321
322
323 #-----
324 # Name : EMEXPRESSPORT
325 # Datatype : Number
326 # Description : Enterprise Manager Configuration Type
327 # Valid values : Check Oracle12c Administrator's Guide
328 # Default value : NONE
329 # Mandatory : No, will be picked up from DBEXPRESS_HTTPS_PORT env variable
330 # or auto generates a free port between 5500 and 5599
331 #
332 #EMEXPRESSPORT = ""
333
334
335 #-----
336 # Name : RUNCVUCHECKS
337 # Datatype : Boolean
338 # Description : Specify whether to run Cluster Verification Utility checks
339 # periodically in Cluster environment
340 # Valid values : TRUE\FALSE
341 # Default value : FALSE
342 # Mandatory : No
343 #
344 #RUNCVUCHECKS = FALSE
345
346 #-----
347 # Name : DBSNMPPASSWORD
348 # Datatype : String
349 # Description : Password for DBSNMP user
350 # Valid values : Check Oracle12c Administrator's Guide
351 # Default value : None
352 # Mandatory : Yes, if EMCONFIGURATION is specified or
353 # the value of RUNCVUCHECKS is TRUE
354 #
355 #DBSNMPPASSWORD = "password"
356
357 #-----
358 # Name : OMSHOST
359 # Datatype : String
360 # Description : EM management server host name
361 # Default value : None
362 # Mandatory : Yes, if CENTRAL is specified for EMCONFIGURATION
363 #
364 #OMSHOST =
365
366 #-----
367 # Name : OMSPORT
368 # Datatype : Number
369 # Description : EM management server port number
370 # Default value : None
371 # Mandatory : Yes, if CENTRAL is specified for EMCONFIGURATION

```

```

372 #-----
373 #OMSPORT =
374 #
375 #-----
376 # Name : EMUSER
377 # Datatype : String
378 # Description : EM Admin username to add or modify targets
379 # Default value : None
380 # Mandatory : Yes, if CENTRAL is specified for EMCONFIGURATION
381 #
382 #EMUSER =
383 #
384 #-----
385 # Name : EMPASSWORD
386 # Datatype : String
387 # Description : EM Admin user password
388 # Default value : None
389 # Mandatory : Yes, if CENTRAL is specified for EMCONFIGURATION
390 #
391 #EMPASSWORD=
392 #
393 #
394 #-----
395 # Name : DVCONFIGURATION
396 # Datatype : Boolean
397 # Description : Specify "True" to configure and enable Oracle Database vault
398 # Valid values : True/False
399 # Default value : False
400 # Mandatory : No
401 #
402 #DVCONFIGURATION = "false"
403 #
404 #-----
405 # Name : DVOWNERNAME
406 # Datatype : String
407 # Description : DataVault Owner
408 # Valid values : Check Oracle12c Administrator's Guide
409 # Default value : None
410 # Mandatory : Yes, if DataVault option is chosen
411 #
412 #DVOWNERNAME = ""
413 #
414 #-----
415 # Name : DVOWNERPASSWORD
416 # Datatype : String
417 # Description : Password for DataVault Owner
418 # Valid values : Check Oracle12c Administrator's Guide
419 # Default value : None
420 # Mandatory : Yes, if DataVault option is chosen
421 #
422 #DVOWNERPASSWORD = ""
423 #
424 #-----
425 # Name : DVACCOUNTMANAGERNAME
426 # Datatype : String
427 # Description : DataVault Account Manager
428 # Valid values : Check Oracle12c Administrator's Guide
429 # Default value : None
430 # Mandatory : No
431 #
432 #DVACCOUNTMANAGERNAME = ""
433 #
434 #-----
435 # Name : DVACCOUNTMANAGERPASSWORD
436 # Datatype : String
437 # Description : Password for DataVault Account Manager
438 # Valid values : Check Oracle12c Administrator's Guide
439 # Default value : None
440 # Mandatory : No
441 #
442 #DVACCOUNTMANAGERPASSWORD = ""
443 #
444 #

```

```

445 # Name : OLSCONFIGURATION
446 # Datatype : Boolean
447 # Description : Specify "True" to configure and enable Oracle Label Security
448 # Valid values : True/False
449 # Default value : False
450 # Mandatory : No
451 #-----
452 #OLSCONFIGURATION = "false"
453
454 #-----
455 # Name : DATAFILEJARLOCATION
456 # Datatype : String
457 # Description : Location of the data file jar
458 # Valid values : Directory containing compressed datafile jar
459 # Default value : None
460 # Mandatory : No
461 #-----
462 #DATAFILEJARLOCATION =
463
464 #-----
465 # Name : DATAFILEDESTINATION
466 # Datatype : String
467 # Description : Location of the data file's
468 # Valid values : Directory for all the database files
469 # Default value : $ORACLE_BASE/oradata
470 # Mandatory : No
471 #-----
472 #DATAFILEDESTINATION =
473
474 #-----
475 # Name : RECOVERYAREADESTINATION
476 # Datatype : String
477 # Description : Location of the data file's
478 # Valid values : Recovery Area location
479 # Default value : $ORACLE_BASE/flash_recovery_area
480 # Mandatory : No
481 #-----
482 #RECOVERYAREADESTINATION=
483
484 #-----
485 # Name : STORAGETYPE
486 # Datatype : String
487 # Description : Specifies the storage on which the database is to be created
488 # Valid values : FS (CFS for RAC), ASM
489 # Default value : FS
490 # Mandatory : No
491 #-----
492 #STORAGETYPE=FS
493
494 #-----
495 # Name : DISKGROUPNAME
496 # Datatype : String
497 # Description : Specifies the disk group name for the storage
498 # Default value : DATA
499 # Mandatory : No
500 #-----
501 #DISKGROUPNAME=DATA
502
503 #-----
504 # Name : ASMSNMP_PASSWORD
505 # Datatype : String
506 # Description : Password for ASM Monitoring
507 # Default value : None
508 # Mandatory : No
509 #-----
510 #ASMSNMP_PASSWORD=""
511
512 #-----
513 # Name : RECOVERYGROUPNAME
514 # Datatype : String
515 # Description : Specifies the disk group name for the recovery area
516 # Default value : RECOVERY
517 # Mandatory : No

```

```

518 #-----
519 #RECOVERYGROUPNAME=RECOVERY
520
521
522 #-----
523 # Name : CHARACTERSET
524 # Datatype : String
525 # Description : Character set of the database
526 # Valid values : Check Oracle12c National Language Support Guide
527 # Default value : "US7ASCII"
528 # Mandatory : NO
529 #
530 #CHARACTERSET = "US7ASCII"
531
532 #-----
533 # Name : NATIONALCHARACTERSET
534 # Datatype : String
535 # Description : National Character set of the database
536 # Valid values : "UTF8" or "AL16UTF16". For details, check Oracle12c National
      → Language Support Guide
537 # Default value : "AL16UTF16"
538 # Mandatory : No
539 #
540 #NATIONALCHARACTERSET= "UTF8"
541
542 #-----
543 # Name : REGISTERWITHDIRSERVICE
544 # Datatype : Boolean
545 # Description : Specifies whether to register with Directory Service.
546 # Valid values : TRUE \ FALSE
547 # Default value : FALSE
548 # Mandatory : No
549 #
550 #REGISTERWITHDIRSERVICE= TRUE
551
552 #
553 # Name : DIRSERVICEUSERNAME
554 # Datatype : String
555 # Description : Specifies the name of the directory service user
556 # Mandatory : YES, if the value of registerWithDirService is TRUE
557 #
558 #DIRSERVICEUSERNAME= "name"
559
560 #
561 # Name : DIRSERVICEPASSWORD
562 # Datatype : String
563 # Description : The password of the directory service user.
564 #           You can also specify the password at the command prompt
      → instead of here.
565 # Mandatory : YES, if the value of registerWithDirService is TRUE
566 #
567 #DIRSERVICEPASSWORD= "password"
568
569 #
570 # Name : WALLETPASSWORD
571 # Datatype : String
572 # Description : The password for wallet to created or modified.
573 #           You can also specify the password at the command prompt
      → instead of here.
574 # Mandatory : YES, if the value of registerWithDirService is TRUE
575 #
576 #WALLETPASSWORD= "password"
577
578 #
579 # Name : LISTENERS
580 # Datatype : String
581 # Description : Specifies list of listeners to register the database with.
582 #           By default the database is configured for all the listeners
      → specified in the
583 #           $ORACLE_HOME/network/admin/listener.ora
584 # Valid values : The list should be comma separated like "listener1,listener2".
585 # Mandatory : NO
586 #

```

```

587 #LISTENERS = "listener1,listener2"
588 -----
589 # Name : VARIABLESFIL
590 # Datatype : String
591 # Description : Location of the file containing variable value pair
592 # Valid values : A valid file-system file. The variable value pair format in
593 #                 this file
594 #                   is <variable>=<value>. Each pair should be in a new line.
595 # Default value : None
596 # Mandatory : NO
597 -----
598 #VARIABLESFIL =
599
600 -----
601 # Name : VARIABLES
602 # Datatype : String
603 # Description : comma separated list of name=value pairs. Overrides variables
604 #                 defined in variablefile and templates
605 # Default value : None
606 # Mandatory : NO
607 -----
608 #VARIABLES =
609 -----
610 # Name : INITPARAMS
611 # Datatype : String
612 # Description : comma separated list of name=value pairs. Overrides
613 #                 initialization parameters defined in templates
614 # Default value : None
615 # Mandatory : NO
616 -----
617 #INITPARAMS =
618 -----
619 # Name : SAMPLESCHEMA
620 # Datatype : Boolean
621 # Description : Specifies whether or not to add the Sample Schemas to your
622 #                 database
623 # Valid values : TRUE \ FALSE
624 # Default value : FASLE
625 # Mandatory : No
626 -----
627 #SAMPLESCHEMA=TRUE
628 -----
629 # Name : MEMORYPERCENTAGE
630 # Datatype : String
631 # Description : percentage of physical memory for Oracle
632 # Default value : None
633 # Mandatory : NO
634 -----
635 #MEMORYPERCENTAGE = "40"
636
637 -----
638 # Name : DATABASESETYPE
639 # Datatype : String
640 # Description : used for memory distribution when MEMORYPERCENTAGE specified
641 # Valid values : MULTIPURPOSE|DATA_WAREHOUSING|OLTP
642 # Default value : MULTIPURPOSE
643 # Mandatory : NO
644 -----
645 #DATABASESETYPE = "MULTIPURPOSE"
646
647 -----
648 # Name : AUTOMATICMEMORYMANAGEMENT
649 # Datatype : Boolean
650 # Description : flag to indicate Automatic Memory Management is used
651 # Valid values : TRUE/FALSE
652 # Default value : TRUE
653 # Mandatory : NO
654 -----
655 #AUTOMATICMEMORYMANAGEMENT = "TRUE"

```

```

656 #-----
657 #-----#
658 # Name : TOTALMEMORY
659 # Datatype : String
660 # Description : total memory in MB to allocate to Oracle
661 # Valid values :
662 # Default value :
663 # Mandatory : NO
664 #-----
665 #TOTALMEMORY = "800"
666
667
668 #-----*** End of CREATEDATABASE section
   ↳ ***-----
669
670 #-----
671 # createTemplateFromDB section is used when OPERATION_TYPE is defined as
   ↳ "createTemplateFromDB".
672 #-----
673 [createTemplateFromDB]
674 #-----
675 # Name : SOURCEDB
676 # Datatype : String
677 # Description : The source database from which to create the template
678 # Valid values : The format is <host>:<port>:<sid>
679 # Default value : none
680 # Mandatory : YES
681 #-----
682 SOURCEDB = "myhost:1521:orcl"
683
684 #-----
685 # Name : SYSDBAUSERNAME
686 # Datatype : String
687 # Description : A user with DBA role.
688 # Default value : none
689 # Mandatory : YES
690 #-----
691 SYSDBAUSERNAME = "system"
692
693 #-----
694 # Name : SYSDBAPASSWORD
695 # Datatype : String
696 # Description : The password of the DBA user.
697 #           You can also specify the password at the command prompt
   ↳ instead of here.
698 # Default value : none
699 # Mandatory : YES
700 #-----
701 SYSDBAPASSWORD = "password"
702
703 #-----
704 # Name : TEMPLATENAME
705 # Datatype : String
706 # Description : Name for the new template.
707 # Default value : None
708 # Mandatory : Yes
709 #-----
710 TEMPLATENAME = "My Copy TEMPLATE"
711
712 #-----*** End of createTemplateFromDB section
   ↳ ***-----
713
714 #-----
715 # createCloneTemplate section is used when OPERATION_TYPE is defined as
   ↳ "createCloneTemplate".
716 #-----
717 [createCloneTemplate]
718 #
719 # Name : SOURCEDB
720 # Datatype : String
721 # Description : The source database is the SID from which to create the
   ↳ template.
722 #           This database must be local and on the same ORACLE_HOME.

```

```

723 # Default value : none
724 # Mandatory     : YES
725 #-----
726 SOURCEDB = "orcl"
727 #-----
728 #-----
729 # Name          : SYSDBAUSERNAME
730 # Datatype      : String
731 # Description   : A user with DBA role.
732 # Default value : none
733 # Mandatory    : YES, if no OS authentication
734 #-----
735 #SYSDBAUSERNAME = "sys"
736
737 #-----
738 # Name          : SYSDBAPASSWORD
739 # Datatype      : String
740 # Description   : The password of the DBA user.
741 #           You can also specify the password at the command prompt
742 #           ↗ instead of here.
743 # Default value : none
744 # Mandatory    : YES
745 #-----
746 #SYSDBAPASSWORD = "password"
747 #-----
748 # Name          : TEMPLATENAME
749 # Datatype      : String
750 # Description   : Name for the new template.
751 # Default value : None
752 # Mandatory    : Yes
753 #-----
754 TEMPLATENAME = "My Clone TEMPLATE"
755
756 #-----
757 # Name          : DATAFILEJARLOCATION
758 # Datatype      : String
759 # Description   : Location of the data file jar
760 # Valid values  : Directory where the new compressed datafile jar will be placed
761 # Default value : $ORACLE_HOME/assistants/dbca/templates
762 # Mandatory    : NO
763 #-----
764 #DATAFILEJARLOCATION =
765
766 #-----*** End of createCloneTemplate section
767 #           ↗ ***
768 #-----
769 # DELETEDATABASE section is used when DELETE_TYPE is defined as
770 #           ↗ "deleteDatabase".
771 #-----
772 [DELETEDATABASE]
773 #-----
774 # Name          : SOURCEDB
775 # Datatype      : String
776 # Description   : The source database is the SID
777 #           This database must be local and on the same ORACLE_HOME.
778 # Default value : none
779 # Mandatory    : YES
780 #-----
781 SOURCEDB = "orcl"
782 #-----
783 # Name          : SYSDBAUSERNAME
784 # Datatype      : String
785 # Description   : A user with DBA role.
786 # Default value : none
787 # Mandatory    : YES, if no OS authentication
788 #-----
789 #SYSDBAUSERNAME = "sys"
790 SYSDBAUSERNAME = "sys"
791
792 #-----

```

```

793 # Name : SYSDBAPASSWORD
794 # Datatype : String
795 # Description : The password of the DBA user.
796 # You can also specify the password at the command prompt
797 # instead of here.
798 # Default value : none
799 # Mandatory : YES, if no OS authentication
800 #-----
800 #SYSDBAPASSWORD = "password"
801 SYSDBAPASSWORD = "password"
802 #-----*** End of deleteDatabase section
803 #-----*
804 #-----
805 # GENERATESCRIPTS section
806 #-----
807 [generateScripts]
808 #-----
809 # Name : TEMPLATENAME
810 # Datatype : String
811 # Description : Name of the template
812 # Valid values : Template name as seen in DBCA
813 # Default value : None
814 # Mandatory : Yes
815 #-----
816 TEMPLATENAME = "New Database"
817 #-----
818 #-----
819 # Name : GDBNAME
820 # Datatype : String
821 # Description : Global database name of the database
822 # Valid values : <db_name>.<db_domain> - when database domain isn't NULL
823 # <db_name> - when database domain is NULL
824 # Default value : None
825 # Mandatory : Yes
826 #-----
827 #GDBNAME = "orcl12.us.oracle.com"
828 GDBNAME = "orcl"
829 #-----
830 #-----
831 # Name : SCRIPTDESTINATION
832 # Datatype : String
833 # Description : Location of the scripts
834 # Valid values : Directory for all the scripts
835 # Default value : None
836 # Mandatory : No
837 #-----
838 #SCRIPTDESTINATION =
839 #-----
840 #-----
841 # Name : EMCONFIGURATION
842 # Datatype : String
843 # Description : Enterprise Manager Configuration Type
844 # Valid values : CENTRAL
845 # Default value : NONE
846 # Mandatory : No
847 #-----
848 #EMCONFIGURATION = "NONE"
849 #-----
850 #-----
851 #-----
852 # Name : OMSHOST
853 # Datatype : String
854 # Description : EM management server host name
855 # Default value : None
856 # Mandatory : Yes, if CENTRAL is specified for EMCONFIGURATION
857 #-----
858 #OMSHOST =
859 #-----
860 #-----
861 # Name : OMSPORT
862 # Datatype : Number
863 # Description : EM management server port number

```

```

864 # Default value : None
865 # Mandatory      : Yes, if CENTRAL is specified for EMCONFIGURATION
866 #-----
867 #OMSPORT =
868
869 #-----
870 # Name          : EMUSER
871 # Datatype      : String
872 # Description   : EM Admin username to add or modify targets
873 # Default value : None
874 # Mandatory     : Yes, if CENTRAL is specified for EMCONFIGURATION
875 #-----
876 #EMUSER =
877
878 #-----
879 # Name          : EMPASSWORD
880 # Datatype      : String
881 # Description   : EM Admin user password
882 # Default value : None
883 # Mandatory     : Yes, if CENTRAL is specified for EMCONFIGURATION
884 #-----
885 #EMPASSWORD=
886
887 #-----*** End of deleteDatabase section
     ↔ ***-----
888
889 #-----
890 # CONFIGUREDATABASE section is used when OPERATION_TYPE is defined as
     ↔ "configureDatabase".
891 #-----
892 [CONFIGUREDATABASE]
893
894 #-----
895 # Name          : SOURCEDB
896 # Datatype      : String
897 # Description   : The source database is the SID
898 #           This database must be local and on the same ORACLE_HOME.
899 # Default value : none
900 # Mandatory     : YES
901 #-----
902 #SOURCEDB = "orcl"
903 SOURCEDB = "orcl"
904
905 #-----
906 # Name          : SYSDBAUSERNAME
907 # Datatype      : String
908 # Description   : A user with DBA role.
909 # Default value : none
910 # Mandatory     : YES, if no OS authentication
911 #-----
912 #SYSDBAUSERNAME = "sys"
913 SYSDBAUSERNAME = "sys"
914
915
916 #-----
917 # Name          : SYSDBAPASSWORD
918 # Datatype      : String
919 # Description   : The password of the DBA user.
920 #           You can also specify the password at the command prompt
     ↔ instead of here.
921 # Default value : none
922 # Mandatory     : YES, if no OS authentication
923 #-----
924 #SYSDBAPASSWORD =
925 SYSDBAPASSWORD = "password"
926
927 #-----
928 # Name          : REGISTERWITHDIRSERVICE
929 # Datatype      : Boolean
930 # Description   : Specifies whether to register with Directory Service.
931 # Valid values  : TRUE \ FALSE
932 # Default value : FALSE
933 # Mandatory     : No

```

```

934 #-----
935 #REGISTERWITHDIRSERVICE= TRUE
936 #
937 #-----
938 # Name : UNREGISTERWITHDIRSERVICE
939 # Datatype : Boolean
940 # Description : Specifies whether to unregister with Directory Service.
941 # Valid values : TRUE \ FALSE
942 # Default value : FALSE
943 # Mandatory : No
944 #-----
945 #UNREGISTERWITHDIRSERVICE= TRUE
946 #
947 #-----
948 # Name : REGENERATEDBPASSWORD
949 # Datatype : Boolean
950 # Description : Specifies whether regenerate database password in OID/Wallet
951 # Valid values : TRUE \ FALSE
952 # Default value : FALSE
953 # Mandatory : No
954 #-----
955 #REGENERATEDBPASSWORD= TRUE
956 #
957 #-----
958 # Name : DIRSERVICEUSERNAME
959 # Datatype : String
960 # Description : Specifies the name of the directory service user
961 # Mandatory : YES, if the any of the reg/unreg/regenPasswd options specified
962 #-----
963 #DIRSERVICEUSERNAME= "name"
964 #
965 #-----
966 # Name : DIRSERVICEPASSWORD
967 # Datatype : String
968 # Description : The password of the directory service user.
969 # You can also specify the password at the command prompt
# instead of here.
970 # Mandatory : YES, if the any of the reg/unreg/regenPasswd options specified
971 #-----
972 #DIRSERVICEPASSWORD= "password"
973 #
974 #-----
975 # Name : WALLETPASSWORD
976 # Datatype : String
977 # Description : The password for wallet to created or modified.
978 # You can also specify the password at the command prompt
# instead of here.
979 # Mandatory : YES, if the any of the reg/unreg/regenPasswd options specified
980 #-----
981 #WALLETPASSWORD= "password"
982 #
983 #
984 #-----
985 # Name : ENABLESECURITYCONFIGURATION
986 # Datatype : String
987 # Description : Database Security Settings
988 # Valid values : true|false
989 # Default value : true
990 # Mandatory : No
991 #-----
992 #ENABLESECURITYCONFIGURATION = "true"
993 #
994 #
995 #-----
996 # Name : DBSNMPPASSWORD
997 # Datatype : String
998 # Description : Password for DBSNMP user
999 # Valid values : Check Oracle12c Administrator's Guide
1000 # Default value : None
1001 # Mandatory : Yes, if EMCONFIGURATION is specified
1002 #-----
1003 #DBSNMPPASSWORD = "password"
1004

```

```

1005
1006 #-----
1007 # Name : DVCONFIGURATION
1008 # Datatype : Boolean
1009 # Description : Specify "True" to configure and enable Oracle Database vault
1010 # Valid values : True/False
1011 # Default value : False
1012 # Mandatory : No
1013 #-----
1014 #DVCONFIGURATION = "false"
1015
1016 #-----
1017 # Name : DVOWNERNAME
1018 # Datatype : String
1019 # Description : DataVault Owner
1020 # Valid values : Check Oracle12c Administrator's Guide
1021 # Default value : None
1022 # Mandatory : Yes, if DataVault option is chosen
1023 #-----
1024 #DVOWNERNAME = ""
1025
1026 #-----
1027 # Name : DVOWNERPASSWORD
1028 # Datatype : String
1029 # Description : Password for DataVault Owner
1030 # Valid values : Check Oracle12c Administrator's Guide
1031 # Default value : None
1032 # Mandatory : Yes, if DataVault option is chosen
1033 #-----
1034 #DVOWNERPASSWORD = ""
1035
1036 #-----
1037 # Name : DVACCOUNTMANAGERNAMESPACE
1038 # Datatype : String
1039 # Description : DataVault Account Manager
1040 # Valid values : Check Oracle12c Administrator's Guide
1041 # Default value : None
1042 # Mandatory : No
1043 #-----
1044 #DVACCOUNTMANAGERNAMESPACE = ""
1045
1046 #-----
1047 # Name : DVACCOUNTMANAGERPASSWORD
1048 # Datatype : String
1049 # Description : Password for DataVault Account Manager
1050 # Valid values : Check Oracle12c Administrator's Guide
1051 # Default value : None
1052 # Mandatory : No
1053 #-----
1054 #DVACCOUNTMANAGERPASSWORD = ""
1055
1056 #-----*** End of CONFIGUREDATABASE section
   → ***-----
1057
1058
1059 #-----
1060 # ADDINSTANCE section is used when OPERATION_TYPE is defined as "addInstance".
1061 #-----
1062 [ADDINSTANCE]
1063
1064 #-----
1065 # Name : DB_UNIQUE_NAME
1066 # Datatype : String
1067 # Description : DB Unique Name of the RAC database
1068 # Valid values : <db_unique_name>
1069 # Default value : None
1070 # Mandatory : Yes
1071 #-----
1072 #DB_UNIQUE_NAME = "orcl12c.us.oracle.com"
1073 DB_UNIQUE_NAME = "orcl"
1074
1075 #-----
1076 # Name : INSTANCENAME

```

472 VIRTUAL MACHINE - ORACLE RESPONSE FILES

```
1077 # Datatype      : String
1078 # Description   : RAC instance name to be added
1079 # Valid values  : Check Oracle12c Administrator's Guide
1080 # Default value : <sid_prefix>+<highest_current_thread+1>
1081 # Mandatory     : No
1082 #-----
1083 ##INSTANCENAME = "orcl1"
1084 INSTANCENAME = "orcl"
1085
1086 #-----
1087 # Name          : NODENAME
1088 # Datatype      : String
1089 # Description   : Node on which to add new instance
1090 #           (in 10gR2, instance addition is supported on 1 node at a time)
1091 # Valid values  : Cluster node name
1092 # Default value : None
1093 # Mandatory     : Yes
1094 #-----
1095 NODENAME=
1096
1097 #-----
1098 # Name          : OBFUSCATEDPASSWORDS
1099 # Datatype      : Boolean
1100 # Description   : Set to true if passwords are encrypted
1101 # Valid values  : TRUE\FALSE
1102 # Default value : FALSE
1103 # Mandatory     : No
1104 #-----
1105 #OBFUSCATEDPASSWORDS = FALSE
1106
1107 #-----
1108 # Name          : SYSDBAUSERNAME
1109 # Datatype      : String
1110 # Description   : A user with DBA role.
1111 # Default value : none
1112 # Mandatory     : YES
1113 #-----
1114 SYSDBAUSERNAME = "sys"
1115
1116 #-----
1117 # Name          : SYSDBAPASSWORD
1118 # Datatype      : String
1119 # Description   : The password of the DBA user.
1120 # Default value : none
1121 # Mandatory     : YES
1122 #-----
1123 #SYSDBAPASSWORD = "password"
1124 SYSDBAPASSWORD = "password"
1125
1126 #-----
1127 # Name          : SERVICEUSERPASSWORD
1128 # Datatype      : String
1129 # Description   : Password for Windows Service user
1130 # Default value : None
1131 # Mandatory     : If Oracle home is installed with windows service user
1132 #-----
1133 #SERVICEUSERPASSWORD = "password"
1134
1135 #-----*** End of ADDINSTANCE section
    ↳  * * *-----#
1136
1137
1138 #-----
1139 # DELETEINSTANCE section is used when OPERATION_TYPE is defined as
    ↳  "deleteInstance".
1140 #-----
1141 [DELETEINSTANCE]
1142
1143 #-----
1144 # Name          : DB_UNIQUE_NAME
1145 # Datatype      : String
1146 # Description   : DB Unique Name of the RAC database
1147 # Valid values  : <db_unique_name>
```

```

1148 # Default value : None
1149 # Mandatory     : Yes
1150 #-----
1151 #DB_UNIQUE_NAME = "orcl112c.us.oracle.com"
1152 DB_UNIQUE_NAME = "orcl"
1153 #-----
1154 # Name          : INSTANCENAME
1155 # Datatype      : String
1156 # Description   : RAC instance name to be deleted
1157 # Valid values  : Check Oracle112c Administrator's Guide
1158 # Default value : None
1159 # Mandatory    : Yes
1160 # -----
1161 #INSTANCENAME = "orcl112c"
1162 INSTANCENAME = "orcl"
1163 #-----
1164 # Name          : NODENAME
1165 # Datatype      : String
1166 # Description   : Node on which instance to be deleted (SID) is located
1167 # Valid values  : Cluster node name
1168 # Default value : None
1169 # Mandatory    : No
1170 # -----
1171 #NODENAME=
1172 #-----
1173 #OBFUSCATEDPASSWORDS
1174 #-----
1175 # Name          : OBFUSCATEDPASSWORDS
1176 # Datatype      : Boolean
1177 # Description   : Set to true if passwords are encrypted
1178 # Valid values  : TRUE\FALSE
1179 # Default value : FALSE
1180 # Mandatory    : No
1181 # -----
1182 #OBFUSCATEDPASSWORDS = FALSE
1183 #-----
1184 #-----
1185 # Name          : SYSDBAUSERNAME
1186 # Datatype      : String
1187 # Description   : A user with DBA role.
1188 # Default value : none
1189 # Mandatory    : YES
1190 # -----
1191 #SYSDBAUSERNAME = "sys"
1192 #-----
1193 # Name          : SYSDBAPASSWORD
1194 # Datatype      : String
1195 # Description   : The password of the DBA user.
1196 # Default value : none
1197 # Mandatory    : YES
1198 # -----
1199 ##SYSDBAPASSWORD = "password"
1200 SYSDBAPASSWORD = "password"
1201 #-----
1202 #-----*** End of DELETEINSTANCE section
1203 <-- ***-----
1204 #-----
1205 #-----
1206 #-----** CREATEPLUGGABLEDATABASE section is used when OPERATION_TYPE is defined as
1207 <-- "createPluggableDatabase".
1208 #-----
1209 [CREATEPLUGGABLEDATABASE]
1210 #-----
1211 # Name          : SOURCEDB
1212 # Datatype      : String
1213 # Description   : The source database is the SID
1214 #                 This database must be local and on the same ORACLE_HOME.
1215 # Default value : none
1216 # Mandatory    : YES
1217 # -----
1218 SOURCEDB = "orcl"

```

```

1219 #-----
1220 # Name : PDBNAME
1221 # Datatype : String
1222 # Description : The name of new pluggable database
1223 #           This pdb name must not be same as sourcedb name.
1224 # Default value : none
1225 # Mandatory : YES
1226 #-----
1227 PDBNAME = "PDB1"
1228 #-----
1229 #-----
1230 # Name : CREATEASCLONE
1231 # Datatype : Boolean
1232 # Description : specify true or false for PDB to be create as Clone.
1233 #           : When "true" is passed a new PDB GUID is generated for the
1234 #           ↪ plugged in PDB
1235 # Default value : true
1236 # Mandatory : NO
1237 #-----
1238 # CREATEASCLONE = "TRUE"
1239 #-----
1240 #-----
1241 #-----
1242 #-----
1243 # Name : CREATEPDBFROM
1244 # Datatype : String
1245 # Description : specify the source of pdb to be plugged
1246 # Valid values : DEFAULT | FILEARCHIVE | RMANBACKUP | USINGXML
1247 # Default value : DEFAULT
1248 # Mandatory : NO
1249 #-----
1250 # CREATEPDBFROM = "DEFAULT"
1251 #-----
1252 #-----
1253 # Name : PDBARCHIVEFILE
1254 # Datatype : String
1255 # Description : Full path and name for pdbArchive file
1256 # Default value : None
1257 # Mandatory : Mandatory when creating new PDB using FILEARCHIVE
1258 #-----
1259 # PDBARCHIVEFILE = ""
1260 #-----
1261 #-----
1262 # Name : PDBBACKUPFILE
1263 # Datatype : String
1264 # Description : Full path and name for pdb back up file
1265 # Default value : None
1266 # Mandatory : Mandatory when creating new PDB using RMANBACKUP
1267 #-----
1268 # PDBBACKUPFILE = ""
1269 #-----
1270 #-----
1271 # Name : PDBMETADATAFILE
1272 # Datatype : String
1273 # Description : Full path and name for pdb metadata file
1274 # Default value : None
1275 # Mandatory : Mandatory when creating new PDB using RMANBACKUP or USINGXML
1276 #-----
1277 # PDBMETADATAFILE = ""
1278 #-----
1279 #-----
1280 # Name : PDBUSEMULTIPLEBACKUP
1281 # Datatype : boolean
1282 # Description : Flag that can used to create PDB from single or multiple
1283 #           ↪ backupsets
1284 # Default value : true
1285 # Mandatory : Optional when creating new PDB using RMANBACKUP or USINGXML
1286 #-----
1287 # PDBUSEMULTIPLEBACKUP =
1288 #-----
1289 # Name : PDBADMINUSERNAME

```

```

1290 # Datatype      : String
1291 # Description   : PDB Administrator user name
1292 # Default value : None
1293 # Mandatory     : Mandatory only when creating new DEFAULT PDB
1294 #-----
1295 # PDBADMINUSERNAME = ""
1296
1297 #-----
1298 # Name          : PDBADMINPASSWORD
1299 # Datatype      : String
1300 # Description   : PDB Administrator user password
1301 # Default value : None
1302 # Mandatory     : Mandatory only when creating new DEFAULT PDB
1303 #-----
1304 # PDBADMINPASSWORD = ""
1305
1306 #-----
1307 # Name          : CREATENEWPDBADMINUSER
1308 # Datatype      : String
1309 # Description   : When Plugging a pdb from FILEARCHIVE or RMANBACKUP
1310 #           a new PDB Administrator user can be created using this option
1311 #           This option should be given along with pdbadmin username and
    ↳ password
1312 # Default value : False
1313 # Mandatory     : NO
1314 #-----
1315 # CREATENEWPDBADMINUSER = ""
1316
1317 #-----
1318 # Name          : SOURCEFILENAMECONVERT
1319 # Datatype      : String
1320 # Description   : This clause specifies how to locate files listed in an XML
    ↳ file
1321 #           describing a Pluggable Database if they reside in a location
    ↳ different
1322 #           from that specified in the XML file.
1323 #           This clause is valid when creating Pluggable database in
    ↳ USINGXML option
1324 # Valid values  : (<filename_pattern>, <replacement_filename_pattern>,....)
1325 # Default value : "NONE"
1326 # Mandatory     : NO
1327 #-----
1328 # SOURCEFILENAMECONVERT = ""
1329
1330 #-----
1331 # Name          : FILENAMECONVERT
1332 # Datatype      : String
1333 # Description   : This clause specifies how to generate names of files
    ↳ for the Pluggable Database being created using names of
    ↳ existing files
1335 #           This clause is valid when creating Pluggable database in
    ↳ USINGXML option
1336 # Valid values  : (<filename_pattern>, <replacement_filename_pattern>,....)
1337 # Default value : "NONE"
1338 # Mandatory     : NO
1339 #-----
1340 # FILENAMECONVERT = ""
1341
1342 #-----
1343 # Name          : COPYPDBFILES
1344 # Datatype      : Boolean
1345 # Description   : If COPY is specified, it will indicate that datafiles need to
    ↳ be copied.
1346 #           : This option can be true only when FILENAMECONVERT is
    ↳ specified or
1347 #           CDB files are Oracle Managed Files(OMF)
1348 # Valid values  : TRUE \ FALSE
1349 # Default value : False
1350 # Mandatory     : NO
1351 #-----
1352 # COPYPDBFILES = ""
1353
1354 #-----

```

```

1355 # Name : PDBDATAFILEDESTINATION
1356 # Datatype : String
1357 # Description : common location for PDB datafile area
1358 # Default value : None
1359 # Mandatory : NO
1360 #-----
1361 # PDBDATAFILEDESTINATION = ""
1362
1363 #-----
1364 # Name : USEMETADATAFILELOCATION
1365 # Datatype : Boolean
1366 # Description : Specify true if datafile path defined in Meta datafile within
    → PDB
1367 #           archive file is to be used to un-archive datafile.
1368 # Valid values : TRUE \ FALSE
1369 # Default value : FALSE
1370 # Mandatory : NO
1371 #-----
1372 # USEMETADATAFILELOCATION = ""
1373
1374 #-----
1375 # Name : REGISTERWITHDIRSERVICE
1376 # Datatype : Boolean
1377 # Description : Specifies whether to register with Directory Service.
1378 # Valid values : TRUE \ FALSE
1379 # Default value : FALSE
1380 # Mandatory : No
1381 #-----
1382 #REGISTERWITHDIRSERVICE= TRUE
1383
1384 #-----
1385 # Name : DIRSERVICEUSERNAME
1386 # Datatype : String
1387 # Description : Specifies the name of the directory service user
1388 # Mandatory : YES, if the value of registerWithDirService is TRUE
1389 #-----
1390 #DIRSERVICEUSERNAME= "name"
1391
1392 #-----
1393 # Name : DIRSERVICEPASSWORD
1394 # Datatype : String
1395 # Description : The password of the directory service user.
    → You can also specify the password at the command prompt
1396 #           instead of here.
1397 # Mandatory : YES, if the value of registerWithDirService is TRUE
1398 #-----
1399 #DIRSERVICEPASSWORD= "password"
1400
1401 #-----
1402 # Name : WALLETPASSWORD
1403 # Datatype : String
1404 # Description : The password for wallet to created or modified.
    → You can also specify the password at the command prompt
1405 #           instead of here.
1406 # Mandatory : YES, if the value of registerWithDirService is TRUE
1407 #-----
1408 #WALLETPASSWORD= "password"
1409
1410 #-----
1411 # Name : LBACSYSPASSWORD
1412 # Datatype : String
1413 # Description : Password for LBACSYS user
    → You can also specify the password at the command prompt
1414 #           instead of here.
1415 # Mandatory : YES, if the value of registerWithDirService are TRUE
1416 #-----
1417 #LBACSYSPASSWORD= "password"
1418
1419 #-----
1420 # Name : CREATEUSERTABLESPACE
1421 # Datatype : Boolean
1422 # Description : Specify true if a default user tablespace need to be created
    → in new PDB

```

```

1423 #
1424 # Valid values : TRUE \ FALSE
1425 # Default value : TRUE
1426 # Mandatory : NO
1427 #-----
1428 # CREATEUSERTABLESPACE = ""
1429
1430 #-----
1431 # Name : DVCONFIGURATION
1432 # Datatype : Boolean
1433 # Description : Specify "True" to configure and enable Oracle Database vault
1434 # Valid values : True/False
1435 # Default value : False
1436 # Mandatory : No
1437 #-----
1438 #DVCONFIGURATION = "false"
1439
1440 #-----
1441 # Name : DVOWNERNAME
1442 # Datatype : String
1443 # Description : DataVault Owner
1444 # Valid values : Check Oracle12c Administrator's Guide
1445 # Default value : None
1446 # Mandatory : Yes, if DataVault option is chosen
1447 #-----
1448 #DVOWNERNAME = ""
1449
1450 #-----
1451 # Name : DVOWNERPASSWORD
1452 # Datatype : String
1453 # Description : Password for DataVault Owner
1454 # Valid values : Check Oracle12c Administrator's Guide
1455 # Default value : None
1456 # Mandatory : Yes, if DataVault option is chosen
1457 #-----
1458 #DVOWNERPASSWORD = ""
1459
1460 #-----
1461 # Name : DVACCOUNTMANAGERNAME
1462 # Datatype : String
1463 # Description : DataVault Account Manager
1464 # Valid values : Check Oracle12c Administrator's Guide
1465 # Default value : None
1466 # Mandatory : No
1467 #-----
1468 #DVACCOUNTMANAGERNAME = ""
1469
1470 #-----
1471 # Name : DVACCOUNTMANAGERPASSWORD
1472 # Datatype : String
1473 # Description : Password for DataVault Account Manager
1474 # Valid values : Check Oracle12c Administrator's Guide
1475 # Default value : None
1476 # Mandatory : No
1477 #-----
1478 #DVACCOUNTMANAGERPASSWORD = ""
1479
1480 #-----*** End of createPluggableDatabase section
1481   <-- ***-----
1482
1483 # UNPLUGDATABASE section is used when OPERATION_TYPE is defined as
1484   <-- "unplugDatabase".
1485
1486 #-----
1487 [UNPLUGDATABASE]
1488
1489 # Name : SOURCEDB
1490 # Datatype : String
1491 # Description : The source database is the SID
1492   <-- This database must be local and on the same ORACLE_HOME.
1493 # Default value : none
1494 # Mandatory : YES
1495

```

478 VIRTUAL MACHINE - ORACLE RESPONSE FILES

```
1494 SOURCEDB = "orcl"
1495 -----
1496 # Name          : PDBNAME
1497 # Datatype      : String
1498 # Description   : The name of new pluggable database
1499 #           This pdb name must not be same as sourcedb name.
1500 #
1501 # Default value : none
1502 # Mandatory     : YES
1503 #
1504 PDBNAME = "PDB1"
1505 -----
1506 # -----
1507 # Name          : ARCHIVETYPE
1508 # Datatype      : String
1509 # Description   : The unplugged database datafile backup will in tar.gz or rman
1510 #           ↪ backup
1511 #           This pdb name must not be same as sourcedb name.
1512 # Valid values  : TAR | RMAN
1513 # Default value : TAR
1514 # Mandatory    : NO
1515 #
1516 ARCHIVETYPE = "TAR"
1517 #
1518 # -----
1519 # Name          : PDBARCHIVEFILE
1520 # Datatype      : String
1521 # Description   : Full path and name for pdbArchive file
1522 # Default value : None
1523 # Mandatory    : NO
1524 #
1525 # PDBARCHIVEFILE = ""
1526 #
1527 # -----
1528 # Name          : PDBBACKUPFILE
1529 # Datatype      : String
1530 # Description   : Full path and name for pdb back up file
1531 # Default value : None
1532 # Mandatory    : No
1533 #
1534 # PDBBACKUPFILE = ""
1535 #
1536 # -----
1537 # Name          : PDBMETADATAFILE
1538 # Datatype      : String
1539 # Description   : Full path and name for pdb metadata file
1540 # Default value : None
1541 # Mandatory    : No
1542 #
1543 #
1544 #
1545 # Name          : UNREGISTERWITHDIRSERVICE
1546 # Datatype      : Boolean
1547 # Description   : Specifies whether to unregister with Directory Service.
1548 # Valid values  : TRUE \ FALSE
1549 # Default value : FALSE
1550 # Mandatory    : No
1551 #
1552 #UNREGISTERWITHDIRSERVICE= TRUE
1553 #
1554 #
1555 # Name          : DIRSERVICEUSERNAME
1556 # Datatype      : String
1557 # Description   : Specifies the name of the directory service user
1558 # Mandatory     : YES, if the value of unregisterWithDirService is TRUE
1559 #
1560 #DIRSERVICEUSERNAME= "name"
1561 #
1562 #
1563 # Name          : DIRSERVICEPASSWORD
1564 # Datatype      : String
1565 # Description   : The password of the directory service user.
```

```

1566 # You can also specify the password at the command prompt
1567 # instead of here.
1568 # Mandatory : YES, if the value of unregisterWithDirService is TRUE
1569 #-----#
1570 #DIRSERVICEPASSWORD= "password"
1571 #-----#
1572 # Name : WALLETPASSWORD
1573 # Datatype : String
1574 # Description : The password for wallet to created or modified.
1575 # You can also specify the password at the command prompt
1576 # instead of here.
1577 # Mandatory : YES, if the value of unregisterWithDirService is TRUE
1578 #-----#
1579 #WALLETPASSWORD= "password"
1580 #-----*** End of unplugDatabase section
1581 #-----#
1582 #-----#
1583 # DELETEPLUGGABLEDATABASE section is used when OPERATION_TYPE is defined as
1584 # "deletePluggableDatabase".
1585 #[DELETEPLUGGABLEDATABASE]
1586 #-----#
1587 # Name : SOURCEDB
1588 # Datatype : String
1589 # Description : The source database is the SID
1590 # This database must be local and on the same ORACLE_HOME.
1591 # Default value : none
1592 # Mandatory : YES
1593 #-----#
1594 SOURCEDB = "orcl"
1595 #-----#
1596 # Name : PDBNAME
1597 # Datatype : String
1598 # Description : The name of new pluggable database
1599 # This pdb name must not be same as sourcedb name.
1600 # Default value : none
1601 # Mandatory : YES
1602 #-----#
1603 PDBNAME = "PDB1"
1604 #-----*** End of deletePluggableDatabase section
1605 #-----#
1606 #-----#
1607 #-----#
1608 #-----#
1609 # CONFIGUREPLUGGABLEDATABASE section is used when OPERATION_TYPE is defined as
1610 # "configurePluggableDatabase".
1611 #[CONFIGUREPLUGGABLEDATABASE]
1612 #-----#
1613 # Name : SOURCEDB
1614 # Datatype : String
1615 # Description : The source database is the SID
1616 # This database must be local and on the same ORACLE_HOME.
1617 # Default value : none
1618 # Mandatory : YES
1619 #-----#
1620 SOURCEDB = "orcl"
1621 #-----#
1622 # Name : PDBNAME
1623 # Datatype : String
1624 # Description : The name of new pluggable database
1625 # This pdb name must not be same as sourcedb name.
1626 # Default value : none
1627 # Mandatory : YES
1628 #-----#
1629 PDBNAME = "PDB1"
1630 #-----#
1631 #-----#
1632 #-----#

```

```

1633 # Name : DVCONFIGURATION
1634 # Datatype : Boolean
1635 # Description : Specify "True" to configure and enable Oracle Database vault
1636 # Valid values : True/False
1637 # Default value : False
1638 # Mandatory : No
1639 #-----
1640 #DVCONFIGURATION = "false"
1641
1642 #-----
1643 # Name : DVOWNERNAME
1644 # Datatype : String
1645 # Description : DataVault Owner
1646 # Valid values : Check Oracle12c Administrator's Guide
1647 # Default value : None
1648 # Mandatory : Yes, if DataVault option is chosen
1649 #-----
1650 #DVOWNERNAME = ""
1651
1652 #-----
1653 # Name : DVOWNERPASSWORD
1654 # Datatype : String
1655 # Description : Password for DataVault Owner
1656 # Valid values : Check Oracle12c Administrator's Guide
1657 # Default value : None
1658 # Mandatory : Yes, if DataVault option is chosen
1659 #-----
1660 #DVOWNERPASSWORD = ""
1661
1662 #-----
1663 # Name : DVACCOUNTMANAGERNAMESPACE
1664 # Datatype : String
1665 # Description : DataVault Account Manager
1666 # Valid values : Check Oracle12c Administrator's Guide
1667 # Default value : None
1668 # Mandatory : No
1669 #-----
1670 #DVACCOUNTMANAGERNAMESPACE = ""
1671
1672 #-----
1673 # Name : DVACCOUNTMANAGERPASSWORD
1674 # Datatype : String
1675 # Description : Password for DataVault Account Manager
1676 # Valid values : Check Oracle12c Administrator's Guide
1677 # Default value : None
1678 # Mandatory : No
1679 #-----
1680 #DVACCOUNTMANAGERPASSWORD = ""
1681
1682 #-----
1683 # Name : OLSCONFIGURATION
1684 # Datatype : Boolean
1685 # Description : Specify "True" to configure and enable Oracle Label Security
1686 # Valid values : True/False
1687 # Default value : False
1688 # Mandatory : No
1689 #-----
1690 #OLSCONFIGURATION = "false"
1691
1692 #-----
1693 # Name : REGISTERWITHDIRSERVICE
1694 # Datatype : Boolean
1695 # Description : Specifies whether to register with Directory Service.
1696 # Valid values : TRUE \ FALSE
1697 # Default value : FALSE
1698 # Mandatory : No
1699 #-----
1700 #REGISTERWITHDIRSERVICE= TRUE
1701
1702 #-----
1703 # Name : DIRSERVICEUSERNAME
1704 # Datatype : String
1705 # Description : Specifies the name of the directory service user

```

```

1706 # Mandatory      : YES, if the value of registerWithDirService is TRUE
1707 #-----
1708 #DIRSERVICEUSERNAME= "name"
1709 #-----
1710 #-----
1711 # Name           : DIRSERVICEPASSWORD
1712 # Datatype       : String
1713 # Description    : The password of the directory service user.
1714 #                   You can also specify the password at the command prompt
1714 #                   ↳ instead of here.
1715 # Mandatory      : YES, if the value of registerWithDirService is TRUE
1716 #-----
1717 #DIRSERVICEPASSWORD= "password"
1718 #-----
1719 #-----
1720 # Name           : WALLETPASSWORD
1721 # Datatype       : String
1722 # Description    : The password for wallet to created or modified.
1723 #                   You can also specify the password at the command prompt
1723 #                   ↳ instead of here.
1724 # Mandatory      : YES, if the value of registerWithDirService is TRUE
1725 #-----
1726 #WALLETPASSWORD= "password"
1727 #-----
1728 #-----
1729 # Name           : LBACSYSPASSWORD
1730 # Datatype       : String
1731 # Description    : Password for LBACSYS user
1732 #                   You can also specify the password at the command prompt
1732 #                   ↳ instead of here.
1733 # Mandatory      : YES, if the value of olsConfiguration and
1733 #                   ↳ registerWithDirService are TRUE
1734 #-----
1735 #LBACSYSPASSWORD= "password"
1736 #-----
1737 #-----
1738 #-----
1739 # Name           : EMCONFIGURATION
1740 # Datatype       : String
1741 # Description    : Enterprise Manager Configuration Type
1742 # Valid values   : DBEXPRESS|NONE
1743 # Default value  : NONE
1744 # Mandatory      : No
1745 #-----
1746 #EMCONFIGURATION = "NONE"
1747 #-----
1748 #-----
1749 #-----
1750 # Name           : EMEXPRESSPORT
1751 # Datatype       : Number
1752 # Description    : Enterprise Manager Configuration Type
1753 # Valid values   : Check Oracle12c Administrator's Guide
1754 # Default value  : NONE
1755 # Mandatory      : No, will be picked up from DBEXPRESS_HTTPS_PORT env variable
1756 #                   or auto generates a free port between 5500 and 5599
1757 #-----
1758 #EMEXPRESSPORT = ""
1759 #-----
1760 #-----*** End of configurePluggableDatabase section
1760 #-----**-
1761 #-----
1762 [root@localhost ~]#

```