

Session 1

EXERCISE #1: Installing and Validating Java	1
EXERCISE #2: Installing and Validating Cassandra	2
EXERCISE #3: Download And Run the Mini Class Project.....	3
EXERCISE #4: Open the Project in Eclipse	5

EXERCISE #1: Installing and Validating Java

1. Install Java 7 Java Development Kit (JDK) for your platform. According to the instructions in the lesson, ensure that Java 7 is the default version of Java on your system.

```
java -version
```

2. If you intend to install Cassandra using an RPM package or Debian package in [Exercise 2](#), it's not sufficient that the Java version 7 command is simply in your path, but rather is the default Java system-wide. That way when you start Cassandra as a service, the correct version of Java will be picked up by the service. The output should be something like this:

```
java version "1.7.0_40"  
Java(TM) SE Runtime Environment (build 1.7.0_40-b43)  
Java HotSpot(TM) 64-Bit Server VM (build 24.0-b56, mixed  
mode)
```

3. Validate that the Java compiler is working:

```
javac -version
```

Again, it should show you the same version number.

```
javac 1.7.0_40
```

EXERCISE #2: Installing and Validating Cassandra

4. Install Cassandra in a manner appropriate to your system. You may choose to do a apt-get or yum install if your system supports those, a .dmg install on a Macintosh, a .msi install for a Windows system, or a tarball install.
5. Start Cassandra in a manner appropriate to your installation type. This may involve starting the Casandra service, or running the "cassandra" command from the command line. Refer to the install videos for a complete tutorial on starting Cassandra.
6. Validate that Cassandra is running by using connecting to it with the CQL shell program. On Unix machines, invoke cqlsh:

```
cqlsh
```

It will automatically connect to a Cassandra instance on localhost. You should see the connection:

```
Connected to Test Cluster at localhost:9160.  
[cqlsh 4.0.1 | Cassandra 2.0.1 | CQL spec 3.1.1 | Thrift  
protocol 19.37.0] Use HELP for help.  
cqlsh>
```

On Windows machines, start the CQL shell by clicking the "Cassandra CQL Shell" icon in the "DataStax Community Edition" program group.

Ensure the Cassandra version is at least 2.0.0. The output above shows version 2.0.1.

7. Use the cqlsh tool to examine the system tables that are already in the database. In addition, familiarize yourself with two key features of this tool: HELP, and auto complete.
8. If Cassandra is not running you will get output similar to this:

```
Connection error: Could not connect to localhost:9160
```

To use HELP, type HELP, and several commands are presented. A subset is shown below:

```
Documented shell commands:  
=====
```

CAPTURE	COPY	DESCRIBE	EXPAND	SHOW	TRACING
CONSISTENCY	DESC	EXIT	HELP	SOURCE	

CQL help topics:

=====

ALTER	CREATE_TABLE_OPTIONS	REVOKE
ALTER_ADD	CREATE_TABLE_TYPES	SELECT
ALTER_ALTER	CREATE_USER	

[Examine the system tables by describing the schema \(you may type lower case as well\).](#) [You can run many internal commands as well as execute CQL statements.](#)

SHOW HOST;

If you would like to select use auto complete, type up-to a few characters of each word and press the Tab key. The tool either auto-completes your word, or presents a list of possible completions. Auto-complete is not available on all platforms.

SELECT * from system.<press tab key>

```
cqlsh> select * from system.
"IndexInfo"          local          schema_columnfamilies
"NodeIdInfo"         paxos          schema_columns
batchlog             peer_events    schema_keyspaces
compactions_in_progress peers          schema_triggers
hints                range_xfers
```

[End your command with a semi-colon, and press <enter>.](#)

EXERCISE #3: Download And Run the Mini Class Project

The class project consists of a Playlist web application. It runs with a self-contained version of the Jetty application server, and does not need an external server to run it. Once it's running, you can visit the application using a web browser.

- Unpack the session1.zip or session1.tar.gz file.

Zip:

unzip session1.zip

Tar:

tar -xvf session1.tar.gz

10. Let's run it from the command line – all of the necessary java classes are included to run the project. Go into the target subdirectory:

```
cd session1
cd target
```

Notice that there is a .jar file called playlist-1.0-SNAPSHOT.jar, which contains a main class is called "StartJetty". In addition, all of the necessary dependent libraries are in the lib directory.

To [run](#) the program:

Unix:

```
java -cp 'playlist-1.0-SNAPSHOT.jar:lib/*' StartJetty
```

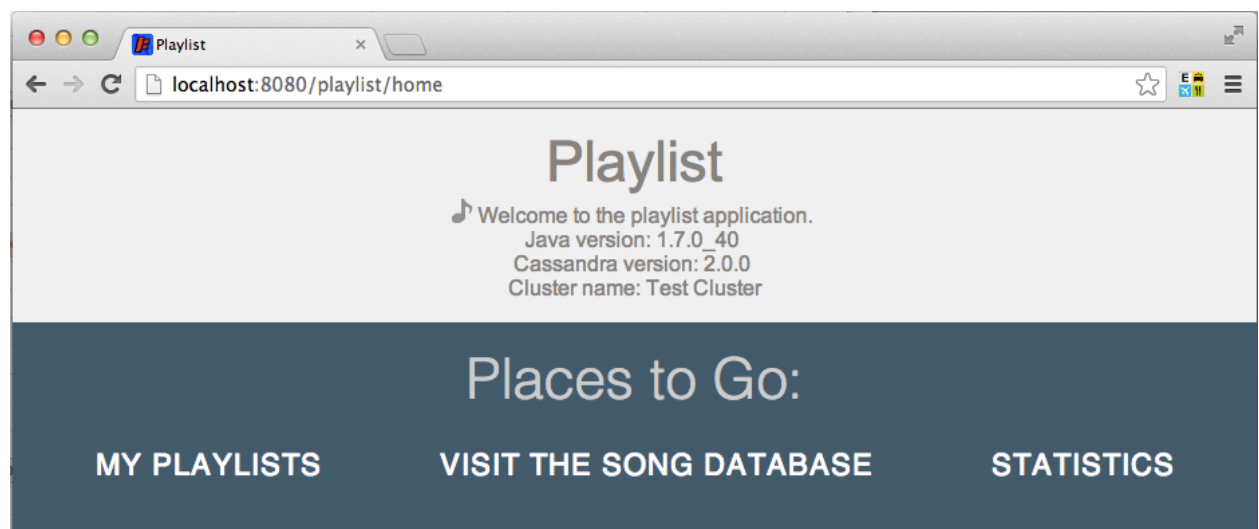
Windows:

```
java -cp 'playlist-1.0-SNAPSHOT.jar;lib\*' StartJetty
```

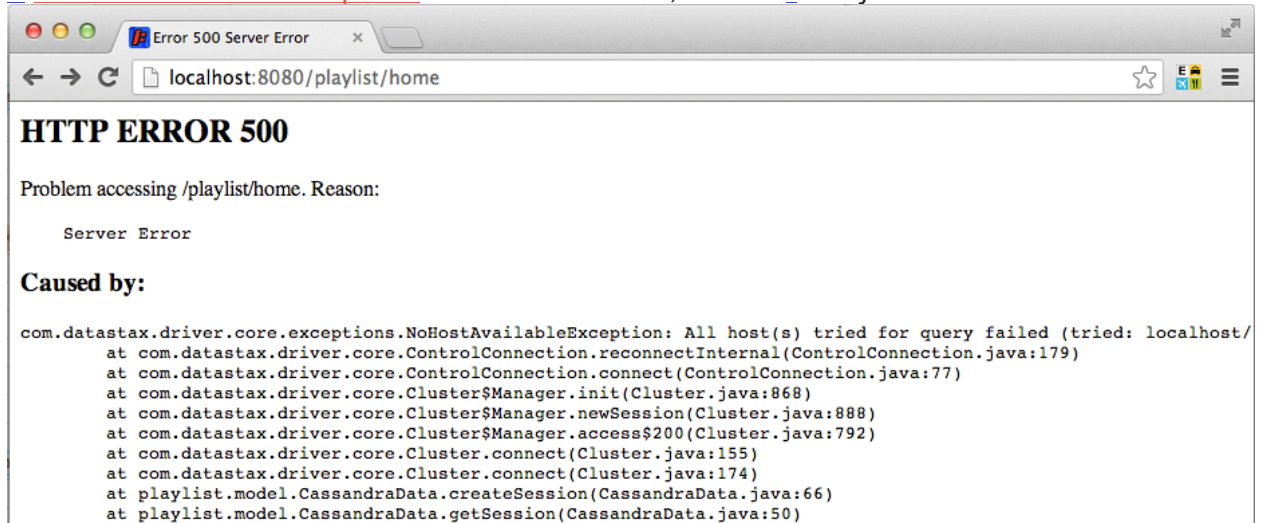
You should see it start:

```
0 [main] INFO StartJetty - Web Resources Directory:
jar:file:/Users/stevelowenthal/teste/session1/target/playli
st-1.0-SNAPSHOT.jar!/webapp
202 [main] INFO org.eclipse.jetty.server.Server - jetty-
8.1.12.v20130726
570 [main] INFO org.eclipse.jetty.server.AbstractConnector
- Started SelectChannelConnector@0.0.0.0:8080
```

11. Visit the Playlist application in your web browser at <http://localhost:8080/playlist>. It will connect to Cassandra, and print out some cluster information.



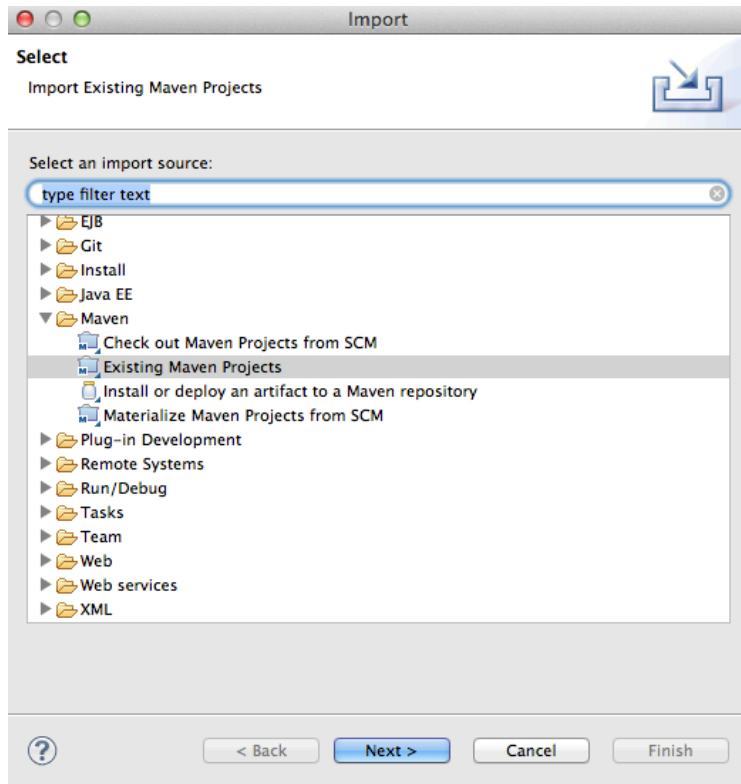
12. If Cassandra is not running, or a connection cannot be made, you are likely to get a [NoHostAvailableException](#). Start Cassandra, and refresh your browser.



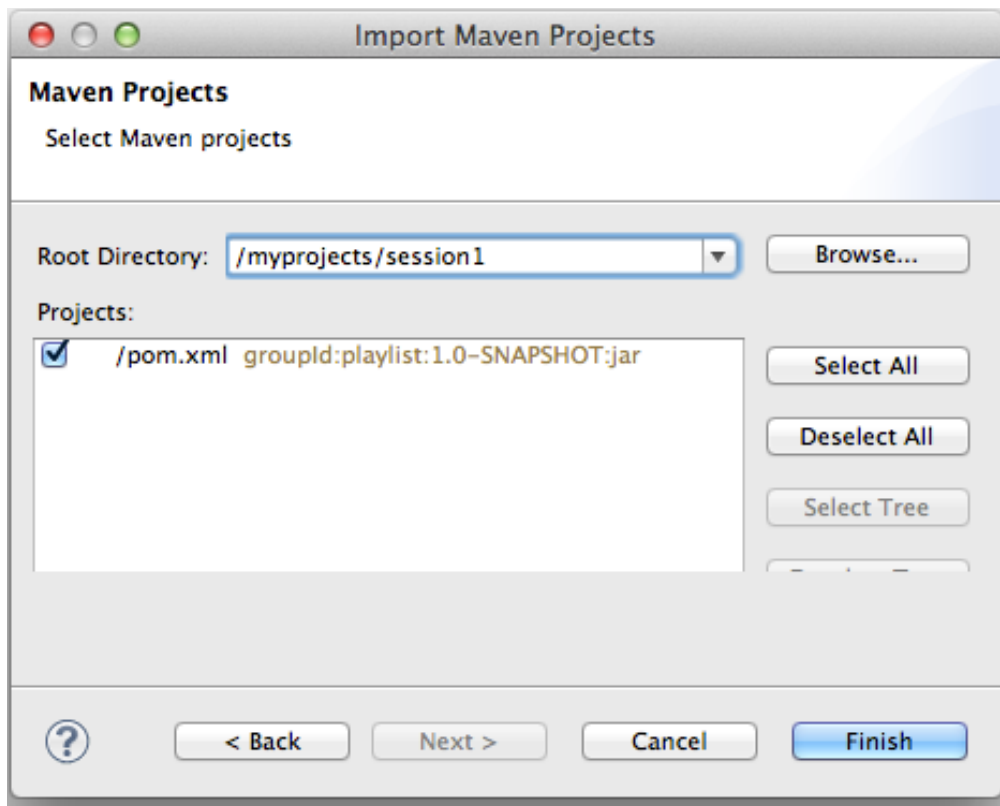
EXERCISE #4: Open the Project in Eclipse

13. Install Eclipse Java EE for Developers and start it. Follow the prompts to set up your workspace. Please refer to the videos for a complete tutorial on installing Eclipse.
14. Import the project into eclipse as a Maven project.

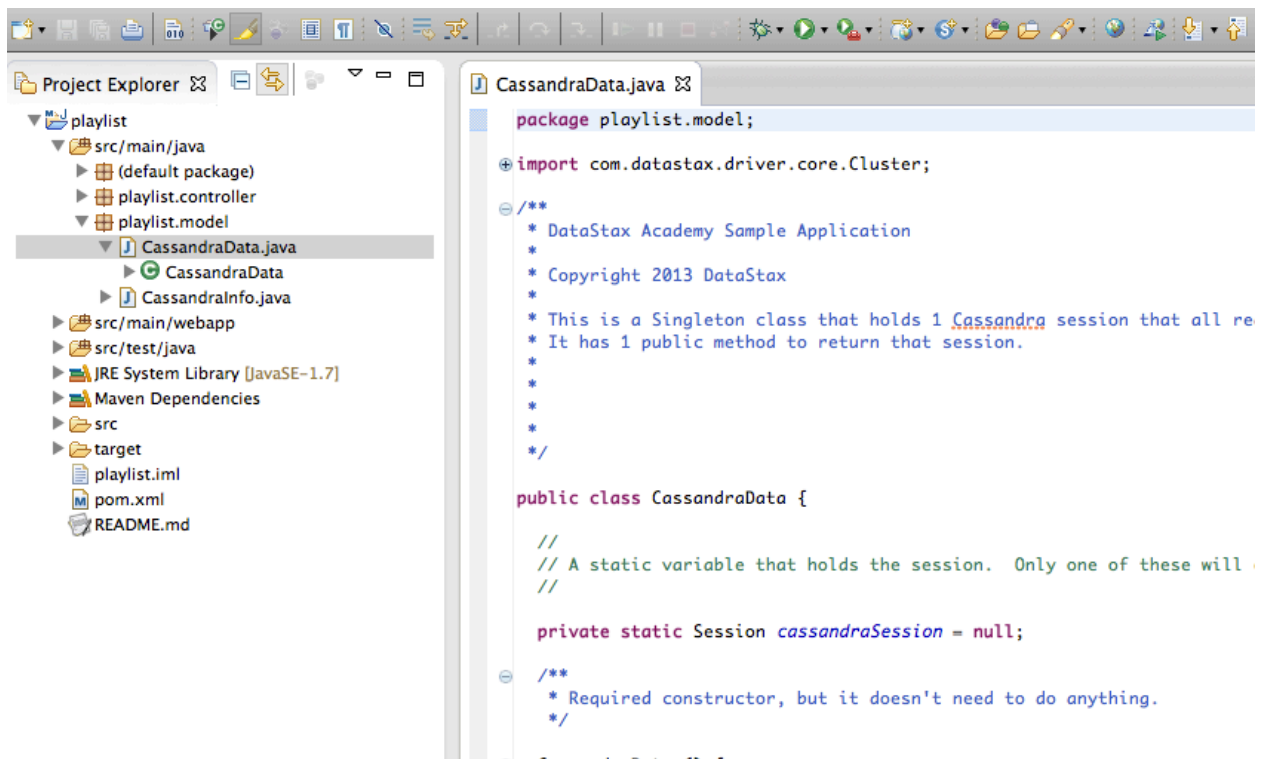
Click File -> Import. Open the Maven Tab. Click "Existing Maven Projects"



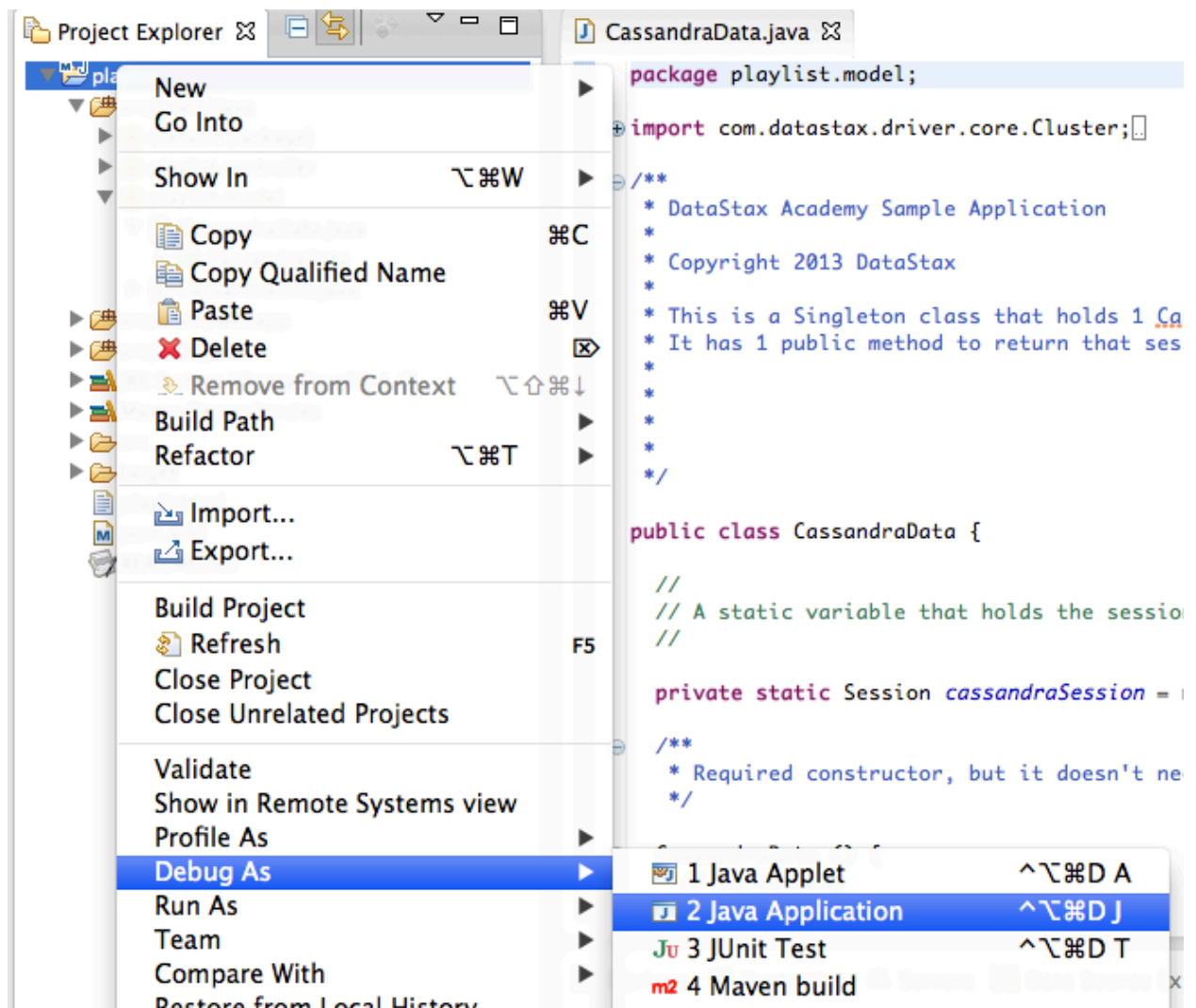
Put the path to your “session1” directory in the “Root Directory” box. Ensure pom.xml is checked [and click Finish](#).



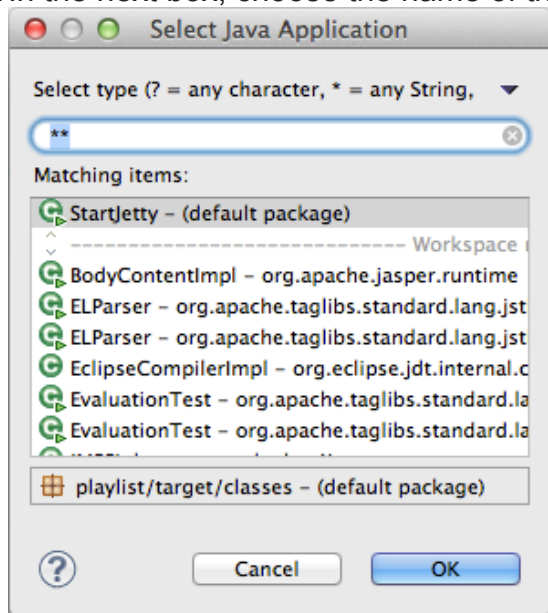
15. Once the project imports, you should see the files in the project explorer pane. Click the triangles, and open the java sources:



16. Click on the `CassandraData.java` file, and browse the source code in the window. You'll see the methods that create the Cassandra Session object, and store it in a static variable, which allows all queries in the application to share the same instance of the Session.
17. You can run or debug the project in Eclipse. Before starting the project, ensure to kill the project you ran from the command line in exercise #3, or you will get an "Address already in use" error. To debug the application, right click the project "playlist", select Debug As. Choose "Java Application".

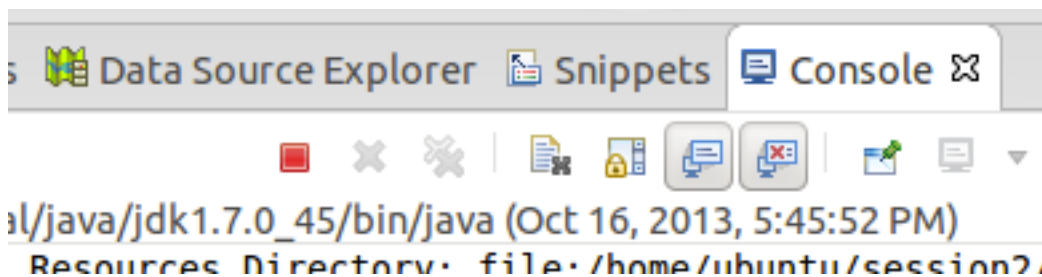


18. In the next box, choose the name of the main class “StartJetty”, and click “OK”.



19. When the application is running in the debugger, you may set break points anywhere in it. You can now visit the application at <http://localhost:8080/playlist>.

20. To stop the application in Eclipse, click the red square in the “Console” tab on the bottom of the screen.



21. If you restart the application, Eclipse remembers your run configuration. You can restart the application by clicking the run or debug buttons in the toolbar:

