UNIVERSITY OF BUEA

Faculty of Engineering and Technology

Department of Computer Engineering

REPORT ON EJB TUTORIAL

CEF 502 (J2EE AND APPLICATION SERVERS)

BY

CHICK DELVERI MUNANG (FE12A028)

May 2016

DEFINITION OF THE FOLDER STRUCTURE AND FILES

The Code directory contains the following sub directories

• EjbComponent
o   This folder contains NetBeans project for creating the Library

Statefull and Stateless remote session bean.
o   This folder also contains the persistence session bean and the

Book entity.

• EjbModule
o   This folder contains the NetBeans project for creating the Timer

Session Bean, The LibraryBookssPersistentBean , The Books Model, the Library Message Bean etc

• EjbModuleTester
o   This is a NetBeans project that is I used to create client which

would late connect to the beans defined in the EjbComponent

project.
o   It contains the EjbTester for the Persistence, Satefull and

Stateless beans. • EjbTester

o   This folder contains the NetBeans project with the clients that are used to test the beans defined in the EjbModule folder
o   It contains the EbjTester for the Persistent, Stateless and

Statefull beans.

• WebServiceClient
o   This folder contains the NetBeans project for creating a

WebService clients.

28-05-2016

The complete code for this tutorial is available on my **github** repository at
https://github.com/chidelmun/EJB-Project

Some modifications I did for the code to work

1. Instead of using the **jndi.properties** file as per the tutorial I inserted the code from the jndi file directly into the tester classes using the class **Properties** from the **java.util** package.
2. To use the Properties class successfully I had to change the the value of the *java.naming.factory.url.pkgs* in the **jndi.properties** file from *org.jboss.naming:org.jnp.interfaces* to *org.jboss.ejb.client.naming.*
3. In addition to the libraries used In the tutorial I need the following two libraries for queries to work.
   1. hibernate-core.jar
   2. jpa-api-2.0-cr-1.jar
4. I had to create the jboss-ds.xml datasource and the persistent unit automatically by linking my database to my project directly using netbeans instead of doing it manually as per the tutorial.
5. All if not most of the queries in the tutorial were not working for me so I had to modify the queries. Eg
   1. From chapter 6 onwards I had to change
      return entityManager.createQuery("From Book").getResultList();
      to
      return entityManager.createNamedQuery("Book.findAll").getResultList();

Definition of Folder Structure and Files

EJB Module

- 1 EJB Module project with 12 packages.

- Each package addressing one section of the tutorial that needed to be test.
- Each package contains session beans for addressing a single part of the tutorial.
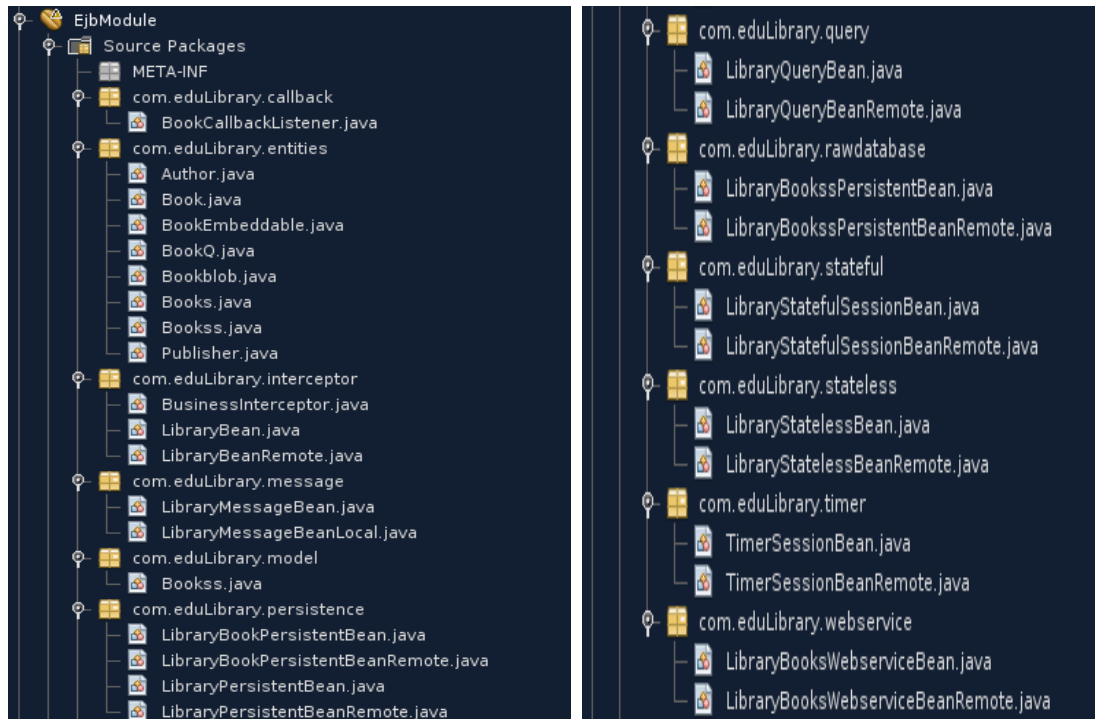
Package Description

Package	Bean Description
com.eduLibrary.callback	Chapter 9: Bean to handle EJB – callbacks
com.eduLibrary.interceptor	Chapter 12: EJB Interceptors
com.eduLibrary.message	Chapter 11: Message driven bean and Dependency injection
com.eduLibrary.model	DAO for Chapter 19.
com.eduLibrary.persistence	Chapter 6, 8, 13, 14, 18: Persistence, Annotation, Embeddable Objects, Clobs and Blobs, Entities Relationship
com.eduLibrary.query	Chapter : EJB -query languages

com.eduLibrary.rawdatabase	Chapter 19: EJB Access Database
com.eduLibrary.stateful	Chapter 5: EJB Stateful Bean
com.eduLibrary.stateless	Chapter 4: EJB Stateless bean
com.eduLibrary.timer	Chapter 10: Timer session bean
com.eduLibrary.webservice	Chapter 22: Webservice bean class.
com.eduLibrary.entities	Contains all the entity classes used in the tutorial e.g author, books, book, and publisher

EJB Module Tester

- 1 EJB Module Tester project with 1 package containing 12 executable classes.
- Each class test one section of the tutorial that needed to be test.
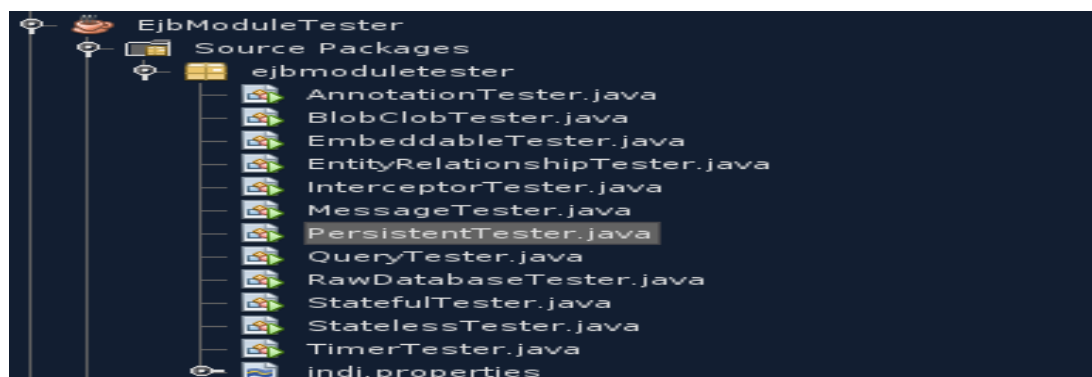
Executable Class	Class Description

AnnotationTester	Test EJB Annotation
BlobClobTester	Test EJB Blob and Clob Data types
EmbeddableTester	Test EJB Embeddable objects
EntityRelationshipTester	Test EJB entity relationships
InterceptorTester	Test EJB Interceptor
MessageTester	Test EJB message driven Bean
PersistentTester	Test EJB persistence
QueryTester	Test EJB query languages
RawDatabaseTester	Test EJB database access
StatelessTester	Test EJB Stateless bean
StatefulTester	Test EJB Stateful bean
TimerTester	Test EJB Timer functionalities

Requirements

1. Java 8.
2. Postgresql Database.
3. Jboss 5 application server.
4. Linux operating system.
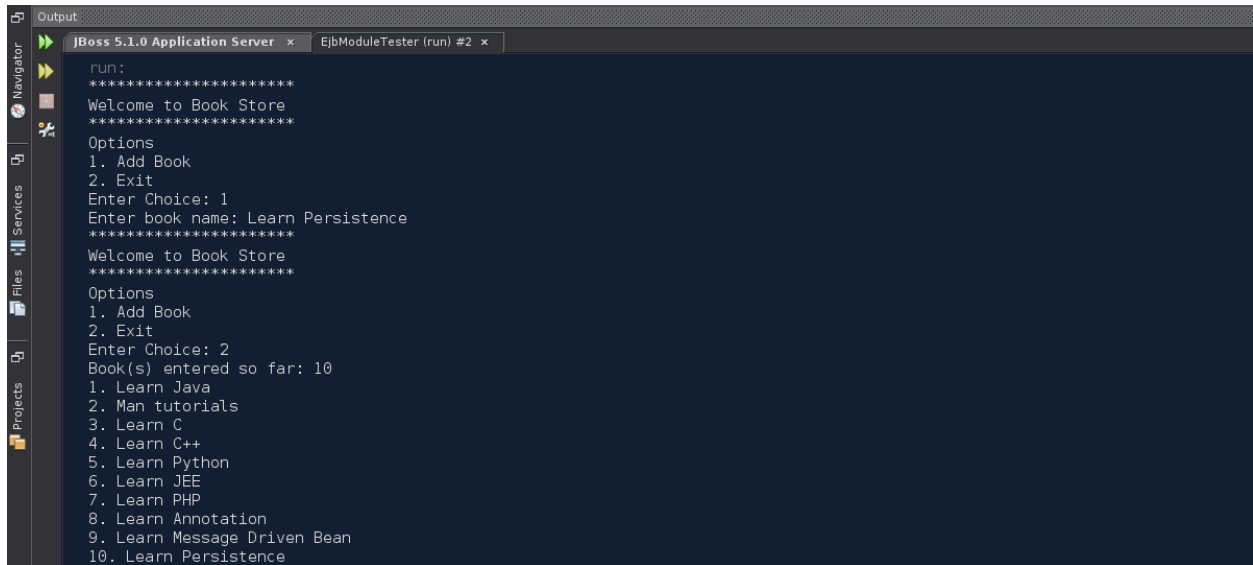
Deployment and Testing

1. Create the following super user, that will have full control over postgres database.
   create user root with password 'root' createdb createuser;
2. Clone the code from my github(**https://github.com/chidelmun/EJB-Project**)
3. Copy the XML file "jbossmq-destinations-service.xml" into /Your_Jboss_5/server/default/deploy directory.

Running from Code

1. Open the projects EjbModule, EjbModuleTester and WebServiceClientEjb in your netbeans IDE
2. Clean and Build each of the above mentioned projects
3. Deploy EjbModule on your server by Right Clicking on EjbModule - **>** Deploy
4. Execute the testers from Netbeans, Click on EjbModuleTester -> Source Packages -> *ejbmoduletester*
5. The package *ejbmoduletester* has 12 executable classes based on the various chapters with examples.

Sample output

```
run:
*********************
Welcome to Book Store
*********************
Options
1. Add Book
2. Exit
Enter Choice: 1
Enter book name: Learn Persistence
*********************
Welcome to Book Store
*********************
Options
1. Add Book
2. Exit
Enter Choice: 2
Book(s) entered so far: 10
1. Learn Java
2. Man tutorials
3. Learn C
4. Learn C++
5. Learn Python
6. Learn JEE
7. Learn PHP
8. Learn Annotation
9. Learn Message Driven Bean
10. Learn Persistence
```

Running from the Terminal

- Launch your server
- cd /Path_to_EjbModuleTester/dist
- Type the following commands based on what chapter or file you want to test

Chapter 3 & 4 Stateless Bean

Run: java -cp EjbModuleTester.jar ejbmoduletester.StatelessTester

Chapter 5 Stateful Bean

Run: java -cp EjbModuleTester.jar ejbmoduletester.StatefulTester

```
ulrich@ulrich-HP-Pavilion-g6-Notebook-PC:~/NetBeansProjects/EJB/EjbModuleTester/dist$ java -cp EjbModuleTester.jar ejbmoduletester.StatefulTester
************************
Welcome to Book Store
************************
Options
1. Add Book
2. Exit
Enter Choice: 1
Enter book name: Learn COBOL
************************
Welcome to Book Store
************************
Options
1. Add Book
2. Exit
Enter Choice: 2
Book(s) entered so far: 1
1. Learn COBOL
***Using second lookup to get library stateful object***
Book(s) entered so far: 0
```

Chapter 6 Persistence

Run: java -cp EjbModuleTester.jar ejbmoduletester.PersistentTester

```
ulrich@ulrich-HP-Pavilion-g6-Notebook-PC:~/NetBeansProjects/EJB/EjbModuleTester/dist$ java -cp EjbModuleTester.jar ejbmoduletester.PersistentTester
************************
Welcome to Book Store
************************
Options
1. Add Book
2. Exit
Enter Choice: 1
Enter book name: Learn PHP
************************
Welcome to Book Store
************************
Options
1. Add Book
2. Exit
Enter Choice: 2
Book(s) entered so far: 7
1. Learn Java
2. Man tutorials
3. Learn C
4. Learn C++
5. Learn Python
6. Learn JEE
7. Learn PHP
```

Chapter 7 Message driven Bean

Run: java -cp EjbModuleTester.jar ejbmoduletester.MessageTester

```
ulrich@ulrich-HP-Pavilion-g6-Notebook-PC:~/NetBeansProjects/EJB/EjbModuleTester/dist$ java -cp EjbModuleTester.jar ejbmoduletester.MessageTester
***********************
Welcome to Book Store
***********************
Options
1. Add Book
2. Exit
Enter Choice: 1
Enter book name: Learn Message Driven Bean
***********************
Welcome to Book Store
***********************
Options
1. Add Book
2. Exit
Enter Choice: 2
Book(s) entered so far: 9
1. Learn Java
2. Man tutorials
3. Learn C
4. Learn C++
5. Learn Python
6. Learn JEE
7. Learn PHP
8. Learn Annotation
9. Learn Message Driven Bean
```

Chapter 8 Annotations

Run: java -cp EjbModuleTester.jar ejbmoduletester.AnnotationTester

```
ulrich@ulrich-HP-Pavilion-g6-Notebook-PC:~/NetBeansProjects/EJB/EjbModuleTester/dist$ java -cp EjbModuleTester.jar ejbmoduletester.AnnotationTester
***********************
Welcome to Book Store
***********************
Options
1. Add Book
2. Exit
Enter Choice: 1
Enter book name: Learn Annotation
***********************
Welcome to Book Store
***********************
Options
1. Add Book
2. Exit
Enter Choice: 2
Book(s) entered so far: 8
1. Learn Java
2. Man tutorials
3. Learn C
4. Learn C++
5. Learn Python
6. Learn JEE
7. Learn PHP
8. Learn Annotation
```

Chapter 9 Callbacks

Didn't work as per the tutorial. I Implemented this otherwise

I copied the annotations (@PostPersist & @PrePersist) into the Books Entity class directly and had the same results as per the tutorial.

```
15:13:35,280 INFO  [STDOUT] BookCallbackListener.prePersist:Book creation in progress: Id is  null
15:13:35,292 INFO  [STDOUT] BookCallbackListener.postPersist::Book created with book id 159
```

Chapter 10 Timer Service

Run: java -cp EjbModuleTester.jar ejbmoduletester.TimerTester

```
ulrich@ulrich-HP-Pavilion-g6-Notebook-PC:~/NetBeansProjects/EJB/EjbModuleTester/dist$ java -cp EjbModuleTester.jar ejbmoduletester.TimerTester
[Fri Jun 10 15:01:02 WAT 2016] timercreated.
```

Chapter 11 Dependency Injection

EJB 3.0 annotations applied on fields or setter methods to inject dependencies. Two annotations exist
@EJB - used to inject other EJB reference
@Resource - used to inject datasource or singleton services like sessionContext, timerService etc.
com.eduLibrary.message package of the EjbModule project makes use of these features

Run: java -cp EjbModuleTester.jar ejbmoduletester.MessageTester

```
15:18:05,796 INFO  [STDOUT] postConstruct:: LibraryPersistentBean session bean created with entity Manager object:
15:18:05,796 INFO  [STDOUT] BookCallbackListener.prePersist:Book creation in progress: Id is  null
15:18:05,805 INFO  [STDOUT] BookCallbackListener.postPersist::Book created with book id 160
15:18:37,226 WARN  [SimpleConnectionManager] A problem has been detected with the connection to remote client 3j011-98e210-ip9tif97-1-ip9ti
g1j-4, jmsClientID=b-r9git9pi-1-79fit9pi-012e89-110j3. It is possible the client has exited without closing its connection(s) or the networ
k has failed. All associated connection resources will be cleaned up.
```

Chapter 12 Interceptors

Run: java -cp EjbModuleTester.jar ejbmoduletester.InterceptorTester

Chapter 13 Embeddable Objects

Run: java -cp EjbModuleTester.jar ejbmoduletester.EmbeddableTester

Chapter 14 BLOBS/CLOBS

Run: java -cp EjbModuleTester.jar ejbmoduletester.BlobClobTester

```
ulrich@ulrich-HP-Pavilion-g6-Notebook-PC:~/NetBeansProjects/EJB/EjbModuleTester/dist$ java -cp EjbModuleTester.jar ejbmoduletester.BlobClobTester
***********************
Welcome to Book Store
***********************
Options
1. Add Book
2. Exit
Enter Choice: 1
Enter book name: Learn BlobClob
***********************
Welcome to Book Store
***********************
Options
1. Add Book
2. Exit
Enter Choice: 2
Book(s) entered so far: 2
1. Learn Python
image bytes: [0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 ]
<book><name>Learn Python</name></book>
2. Learn BlobClob
image bytes: [0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32 ]
<book><name>Learn BlobClob</name></book>
```

Chapter 15 Transactions

A transaction is a single unit of work items, which follows the ACID properties ie Atomic, Consistent, Isolated, and Durable.

Container Managed Transactions Container manages the transaction states.
Bean Managed Transactions Developer manages the life cycle of transaction states.
No code to run here

Chapter 16 Security

Major concern of any enterprise level application. Includes identification of user(s) or system accessing the application.
- Authentication
- Authorization
- User
- User Groups
- User Roles

Container Managed Security

EJB 3.0 has specified following attributes/annotations of security, which EJB containers implement.
- DeclareRoles - Indicates that class will accept the declared roles. Annotations are applied at class level.
- RolesAllowed - Indicates that a method can be accessed by the user of specified role. It can be applied at class level where all methods of a class can be accessed by the user of specified role.
- PermitAll - Indicates that a business method is accessible to all. It can be applied at class as well as at method level.
- DenyAll - Indicates that a business method is not accessible to any of the user specified at class or at method level.
No code to run here

Chapter 17 JNDI BINDINGS

JNDI Java Naming and Directory Interface used for naming and directory services.
Binding - Assigning a name to an EJB object, which can be used later
Lookup - Looking up and getting an object of EJB.
In Jboss, session beans are bound in JNDI in the following format by default.
- local   - EJB-name/local
- remote - EJB-name/remote

Chapter 18 Entity Relationships

EJB 3.0 provides option to define database entity relationships/mappings like one-to-one, one-to-many, many-to-one, and many-to-many relationships.

Run: java -cp EjbModuleTester.jar ejbmoduletester.EntityRelationshipTester

Chapter 19 Access Database

Run: java -cp EjbModuleTester.jar ejbmoduletester.RawDatabaseTester

In EJB 3.0, persistence mechanism is used to access the database in which the container manages the database related operations. Developers can access database using JDBC API call directly in EJB business methods.

Chapter 20 Query Language

Run: java -cp EjbModuleTester.jar ejbmoduletester.QueryTester

```
ulrich@ulrich-HP-Pavilion-g6-Notebook-PC:~/NetBeansProjects/EJB/EjbModuleTester/dist$ java -cp EjbModuleTester.jar ejbmoduletester.QueryTester
***********************
Welcome to Book Store
***********************
Options
1. Add Book
2. Exit
Enter Choice: 1
Enter book name: Learn test automation
***********************
Welcome to Book Store
***********************
Options
1. Add Book
2. Exit
Enter Choice: 2
Book(s) entered so far: 2
1. Learn Selenium testing
2. Learn test automation
```

Chapter 21 Exception Handling

EJBs are a part of enterprise applications which are normally based on distributed environments. So, apart from the normal exceptions that can occur, there can be exceptions like communication failure, security permissions, server down, etc.

EJB container considers exceptions in two ways:

⍰ **Application Exception** - If business rule is violated or exception occurs while executing the business logic.

⍰ **System Exception**- Any exception, which is not caused by business logic or business code. For example, error during EJB lookup. RuntimeException, RemoteException are SystemException.

Chapter 22 Webservices

cd /EJB/WebServiceClientEjb/dist

Run: java -jar "WebServiceClientEjb.jar"