# CEF 508- JAVA EE EJB PROJECT REPORT

**BY**

**NKEANGNYI TONIA AKWANGHA- FE12A138**

COURSE INSTRUCTOR: ENG. BRINARD ELINGESE

## Definition of Folder Structures and Files

1) **EJB Module**

   This module (project) had 12 packages, with all but one package containing session beans to store data for a particular user, for that session. This is as defined in the tutorial. The various packages are:

   - **Callback:** This is the bean to handle EJB callbacks (interception to EJB lifecycle).
   - **Entities:** This package doesn't contain any session bean. However, it contains entity classes for storage in the database (postgresql). So it manages the different database entries.
   - **Interceptor:** This contains the bean for intercepting business method calls.
   - **Message:** This contains the bean used to do tasks asynchronously.
   - **Model:** This package is used to demonstrate database access. It contains a database Object for this.
   - **Persistence:** Persistence is all about incorporating a database in our application.
   - **Query:** This package is meant to ease database querying by making use of the EJB Querying Language.
   - **RawDatabase:** This package too is used to ease and demonstrate database access.
   - **Stateful:** This package contains session beans which preserve the conversational state of server with client.
   - **Stateless:** This bean is used to perform independent operations.
   - **Timer:** This package contains a session bean for building scheduled applications.
   - **WebService:** This last package is used to expose our EJB application as a web service.

2) **EJB Module Tester**

   This project contains a single package and several executable test classes for each of the sections of the tutorial needing testing. These classes are:

   - **AnnotationTester.java:** This tests EJB annotation
   - **BlobClobTester.java:** This is used to test the EJB Blob and Clob data types.
   - **EmbeddableTester.java:** This tests all the EJB embeddable objects
   - **EntityRelationshipTester.java:** This is used to test the EJB entity relationships.
   - **InterceptorTester.java:** This tests EJB business method interception.

- **MessageTester.java:** This tests the message Driven Beans (asynchronous tasks).
- **PersistentTester.java:** This tests EJB persistence (database integration).
- **QueryTester.java:** Tests EJB querying language.
- **RawDatabaseTester.java:** This tests EJB database access.
- **StatefulTester.java:** Tests EJB stateful bean
- **StatelessTester.java:** Tests EJB stateless bean.
- **TimeTester.java:** Tests EJB timer.

3) **WebServiceClientejb**
   This project contains a single executable ***WebServiceClientejb.java*** class to manage our EJB webservice exposure.

## Requirements

To run this project, we need:
- Netbeans IDE 7.x or 8.x
- JDK 1.7 or 1.8
- Jboss-5.1.0.GA
- Postgresql 9.3
- Linux operating system
  The following resources need to included as libraries for proper running of this project:
  - hibernate-core.jar.zip
  - jpa-api-2.0-cr-1.jar
  - postgresql-9.3-1104.jdbc41.jar

## Project Configuration, Deployment and Testing

a) Start the postgresql service and create a postgresql superuser will full access control to postgres database. From the postgresql command line, use the following command:
   *create user root with password 'root' createdb createuser;*
b) Clone the code from github: https://github.com/tonietonia/EjbTutorials
c) Open NetBeans IDE and open the projects EjbModule, EjbModuleTester and WebServiceClientejb from there.
d) Under the services tab right-click on database and select Choose new Connection, then follow the steps to create a Postgresql database connection.

e) Still under the services tab, right click on server and select Add Server, then follow the steps to add a JBOSS application server.
f) Right-click on the EJBModule project and select the properties option. In the properties window, select Libraries -> Add jar/Folder then locate and select the resources mentioned above.
g) Right-click on the EJBModuleTester project and select the properties option. In the properties window, select Libraries ->Add jar/Folder and proceed as done in (f) above. Also, select Add Project and add the EJBModule project.
h) Clean and build each of the projects.
i) When it is built, right-click on the EjbModule project and select deploy.
j) When it is done, go to the EjbModuleTester and run any of the test applications.

## Contact Information for Issues Reporting

Email: toniscus@gmail.com

Github: @tonietonia

Telephone : 677884885