**EJB WITH HIBERNATE IN NET BEANS**

* Create an EJB project in netbeans
* Create package com.hexa.entity
* Create a student entity class

package com.hexa.entity;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity()

@Table(name="student")

public class Student implements Serializable {

@Id

@Column(name="stu\_id")

private int stuId;

@Column(name="stu\_name", length=45)

private String stuName;

@Column(name="dept", length=20)

private String dept;

@Column(name="stu\_dob")

private Date dob;

@Column(name="stu\_marks")

private int marks;

public int getStuId() {

return stuId;

}

public void setStuId(int stuId) {

this.stuId = stuId;

}

public String getStuName() {

return stuName;

}

public void setStuName(String stuName) {

this.stuName = stuName;

}

public String getDept() {

return dept;

}

public void setDept(String dept) {

this.dept = dept;

}

public Date getDob() {

return dob;

}

public void setDob(Date dob) {

this.dob = dob;

}

public int getMarks() {

return marks;

}

public void setMarks(int marks) {

this.marks = marks;

}

@Override

public String toString() {

return stuId + " " + stuName + " " + marks

+ " " + dept + " " + dob;

}

}

* Create com.hex.service package
* Create a stateless session bean “**StudentService**” in the above package
* Click on service tab in net beans
* Click on databases -> create new Connection -> Select mysql(Connector/Jdriver)
* Provide the username and password credentials click on next.
* Provide the connection name as “jdbc/mysql
* Copy and paste the persistence.xml file under configuration folder
* Create interface StudentRemote
* Go to browser load localhost:4848
* Click on JDBC->Connection pool -> new and provide the following details
  + Pool Name : MySqlConnectionPool
  + Resource type : java.sql.Driver
  + Database driver vendor : MySql
  + Click Next -> Additional Properties
    - URL -> jdbc:mysql://localhost:3306/test
    - Port:3306
    - Server name : localhost
    - User : root
    - Password : root

Create JDBC Resource.

Provide the JNDi name and select the connection pool

Click on ok

package com.hexa.service;

import com.hexa.entity.Student;

import java.util.List;

import javax.ejb.Remote;

@Remote

public interface StudentRemote {

public List<Student> getStudents();

}

package com.hexa.service;

import com.hexa.entity.Student;

import java.util.List;

import javax.ejb.Stateless;

import javax.ejb.LocalBean;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

import javax.persistence.Query;

@Stateless(name = "stubean1",mappedName = "stubean2")

public class StudentService implements StudentRemote{

@PersistenceContext(name="EntityProjectPU")

private EntityManager em;

@Override

public List<Student> getStudents() {

Query qry = em.createQuery("from Student s ");

List<Student> lst = qry.getResultList();

return lst;

}

}

**Client Code :**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package entityclient;

import com.hexa.service.StudentRemote;

import java.util.Properties;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.naming.NamingException;

import javax.persistence.criteria.CriteriaBuilder;

/\*\*

\*

\* @author Hvuser

\*/

public class EntityClient {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) throws NamingException {

// TODO code application logic here

System.out.println("Client started");

Properties props = new Properties();

props.setProperty(Context.INITIAL\_CONTEXT\_FACTORY,

"com.sun.enterprise.naming.SerialInitContextFactory");

props.setProperty("org.omg.CORBA.ORBInitialHost", "localhost");

props.setProperty("org.omg.CORBA.ORBInitialPort", "3700");

Context ctx = new InitialContext(props);

StudentRemote ser = (StudentRemote) ctx.lookup("stubean2");

System.out.println(ser.getStudents());

}

}

**Output :**

Client started

[1001 Abhi 86 ECE Thu Oct 12 00:00:00 IST 1995, 1002 Benny 86 ECE Thu Oct 19 00:00:00 IST 1995, 1003 Edward 100 EEE Fri May 19 00:00:00 IST 1995, 1004 Stefan 95 EEE Fri May 12 00:00:00 IST 1995, 1005 Katherine 55 ECE Sun May 14 00:00:00 IST 1995, 1006 Bonnie 80 EEE Mon Sep 02 00:00:00 IST 1996, 1007 Ram Kumar 69 ECE Fri Jun 09 00:00:00 IST 1995, 1008 Damon 85 EEE Wed Jul 03 00:00:00 IST 1996, 1009 Ravi Kumar 95 ECE Sun Jun 04 00:00:00 IST 1995]

**EJB in Eclipse**

1. Add glass fish server
2. Create EJB project
3. Create a package com.hexa.service
4. Add the service files (Interface and implementations)
5. Right click on server : Add/Remove -> Add your created project
6. Start server
7. Check if the JNDI names are generated properly
8. Create new java project StatelessClient
9. Refer Libraries [c: -> Program Files -> GlassFish -> lib -> 4 jar files at last]
10. Copy the interface files of server into com.hexa.service package
11. If error comesgo to build path -> select java compiler -> select 1.7
12. Add the client code in com.hexa.client package
13. Add the build path library as jdk1.8
14. **package** com.hexa.client;

Client Code

1. **import** java.util.Properties;

**import** javax.naming.Context;

**import** javax.naming.InitialContext;

**import** javax.naming.NamingException;

**import** com.hexa.service.Product;

**import** com.hexa.service.ProductSerRemote;

**public** **class** ProductClient {

**public** **static** **void** main(String[] args) **throws** NamingException {

// **TODO** Auto-generated method stub

Properties props = **new** Properties();

props.setProperty(Context.*INITIAL\_CONTEXT\_FACTORY*,

"com.sun.enterprise.naming.SerialInitContextFactory");

props.setProperty("org.omg.CORBA.ORBInitialHost", "localhost");

props.setProperty("org.omg.CORBA.ORBInitialPort", "3700");

Context ctx = **new** InitialContext(props);

ProductSerRemote ser = (ProductSerRemote) ctx.lookup("pro2");

System.*out*.println(ser.getProduct(1));

}

}

**package** com.hexa.service;

**import** java.io.Serializable;

**public** **class** Product **implements** Serializable{

/\*\*

\*

\*/

**private** **static** **final** **long** *serialVersionUID* = 1L;

**private** **int** pid;

**private** String prodName;

**private** **double** price;

**public** Product(){

}

**public** **int** getPid() {

**return** pid;

}

**public** **void** setPid(**int** pid) {

**this**.pid = pid;

}

**public** String getProdName() {

**return** prodName;

}

**public** **void** setProdName(String prodName) {

**this**.prodName = prodName;

}

**public** **double** getPrice() {

**return** price;

}

@Override

**public** String toString() {

// **TODO** Auto-generated method stub

**return** pid + " " + prodName + " " + price;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

**public** Product(**int** pid, String prodName, **double** price) {

**super**();

**this**.pid = pid;

**this**.prodName = prodName;

**this**.price = price;

}

}

**package** com.hexa.service;

**import** java.util.HashMap;

**import** javax.ejb.Remote;

@Remote

**public** **interface** ProductSerRemote {

**public** Product getProduct(**int** productId);

}

Server Code

**package** com.hexa.service;

**import** java.util.HashMap;

**import** java.util.Map;

**import** javax.ejb.LocalBean;

**import** javax.ejb.Stateless;

/\*\*

\* Session Bean implementation class ProductService

\*/

@Stateless(name = "pro1", mappedName = "pro2")

**public** **class** ProductService **implements** ProductSerRemote{

**private** **static** Map<Integer, Product> *prodMap* = **new** HashMap<>();

/\*\*

\* Default constructor.

\*/

**static** {

*prodMap*.put(1, **new** Product(1, "Mouse", 1000));

*prodMap*.put(2, **new** Product(2, "Keyboard", 2500));

*prodMap*.put(3, **new** Product(3, "Monitor", 7000));

}

**public** ProductService() {

// **TODO** Auto-generated constructor stub

}

@Override

**public** Product getProduct(**int** productId) {

System.*out*.println("in productservice method");

**return** *prodMap*.get(productId);

}

}

Maven Project

1. open oxygen eclipse
2. create new ->