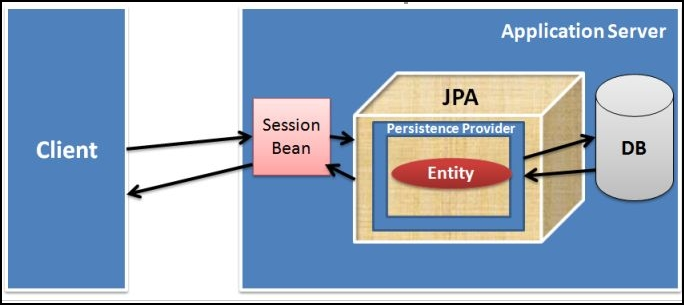
# **Projet :** TimerService

## Etape : Description du projet



**Etape : Création de la base**

Dans mysql, créez une base nommée : **form\_hibernate**

**Etape : Création de la table**

Dans la base **form\_hibernate**, éxécuter le script suivant :

**CREATE TABLE IF NOT EXISTS `project` (**

**`pnumber` int(11) NOT NULL AUTO\_INCREMENT,**

**`pname` varchar(255) NOT NULL,**

**`plocation` varchar(255) NOT NULL,**

**`dept\_no` int(11) NOT NULL,**

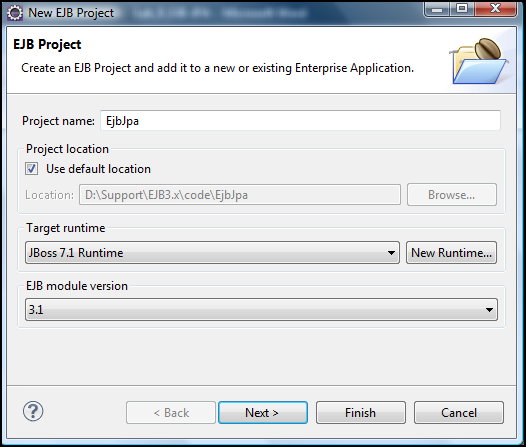
**PRIMARY KEY (`pnumber`)**

**) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ;**

## Etape

Créez un projet EJB dans eclipse

Project Name: **EjbJpa**

****

Finish

**Etape : Création de l’entité**

Créez une classe java **Project.java** dans le package **com.formation.entity**

**package** com.formation.entity;

**import** java.io.Serializable;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

**import** javax.persistence.Column;

@Entity(name = "project")

**public** **class** Project **implements** Serializable {

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

**public** Project() {

**super**();

}

@Id

**private** **int** pnumber;

**private** String pname;

**private** String plocation;

@Column(name = "dept\_no")

**private** **int** deptNo;

**public** **int** getPnumber() {

**return** pnumber;

}

**public** **void** setPnumber(**int** pnumber) {

**this**.pnumber = pnumber;

}

**public** String getPname() {

**return** pname;

}

**public** **void** setPname(String pname) {

**this**.pname = pname;

}

**public** String getPlocation() {

**return** plocation;

}

**public** **void** setPlocation(String plocation) {

**this**.plocation = plocation;

}

**public** **int** getDeptNo() {

**return** deptNo;

}

**public** **void** setDeptNo(**int** deptNo) {

**this**.deptNo = deptNo;

}

@Override

**public** String toString() {

**return** "Project [pnumber=" + pnumber + ", pname=" + pname

+ ", plocation=" + plocation + ", deptNo=" + deptNo + "]";

}

}

**Etape : Création de l’interface du EJB**

Créez une interface java **IProject.java** dans le package **com.formation.service**

**package** com.formation.service;

**import** java.util.List;

**import** javax.ejb.Remote;

**import** com.formation.entity.Project;

@Remote

**public** **interface** IProject {

**void** saveProject(Project project);

Project findProject(Project project);

List<Project> retrieveAllProjects();

}

**Etape : Création du EJB**

Créez une classe java **ProjectBean.java** dans le package **com.formation. service.implementation**

**package** com.formation.client;

**import** java.util.List;

**import** javax.naming.Context;

**import** javax.naming.NamingException;

**import** com.formation.entity.Project;

**import** com.formation.service.IProject;

**import** com.formation.service.implementation.ProjectBean;

**import** com.formation.util.ClientUtil;

**public** **class** EJBApplicationClient {

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

IProject bean = *doLookup*();

// 4. Appel du service

Project p1 = **new** Project();

p1.setPname("Banking App");

p1.setPlocation("Town City");

p1.setDeptNo(1);

Project p2 = **new** Project();

p2.setPname("Office Automation");

p2.setPlocation("Downtown");

p2.setDeptNo(2);

// 4. Call business logic

//Saving new Projects

bean.saveProject(p1);

bean.saveProject(p2);

//Find a Project

p1.setPnumber(1);

Project p3 = bean.findProject(p1);

System.*out*.println(p3);

//Retrieve all projects

System.*out*.println("List of Projects:");

List<Project> projects = bean.retrieveAllProjects();

**for**(Project project : projects)

System.*out*.println(project);

}

**private** **static** IProject doLookup() {

Context context = **null**;

IProject bean = **null**;

**try** {

// 1. Obtenir le context

context = ClientUtil.*getInitialContext*();

// 2. Generate JNDI Lookup name

String lookupName = *getLookupName*();

// 3. Lookup and cast

bean = (IProject) context.lookup(lookupName);

} **catch** (NamingException e) {

e.printStackTrace();

}

**return** bean;

}

**private** **static** String getLookupName() {

// nom ear

String appName = "";

//nom du jar ejb (nom du projet dans eclipse par défaut)

String moduleName = "EjbJpa";

//nom optionnel JBoss7

String distinctName = "";

// Nom de la classe implementant le EJB

String beanName = ProjectBean.**class**.getSimpleName();

// Nom complet de l'interface

**final** String interfaceName = IProject.**class**.getName();

// Create a look up string name

String name = "ejb:" + appName + "/" + moduleName + "/"

+ distinctName + "/" + beanName + "!" + interfaceName;

**return** name;

}

}

**Etape : Configuration de JPA**

Dans le projet EjbJpa

click droit sur le répertoire META-INF 🡺 New 🡺 Other 🡺 XML 🡺 XML file.

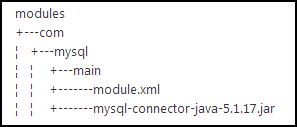
Entrez le nom : **persistence.xml**

**Etape : Configuration DataSource dans JBoss 7.1**

* **Télécharger le driver MYSQL**

<http://dev.mysql.com/downloads/connector/j/>

* **Structure de l’installation**

****

* Créez une structure de répertoire **JbossHome\modules\com\mysql\main**
* Copier le driver mysql dans ce répertoire **mysql-connector-java-5.1.12-bin.jar**
* Dans le même répertoire créer un fichier de configuration **module.xml**

Code :

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<module xmlns=*"urn:jboss:module:1.0"* name=*"com.mysql"*>

<resources>

<resource-root path=*"mysql-connector-java-5.1.12-bin.jar"*/>

</resources>

<dependencies>

<module name=*"javax.api"*/>

</dependencies>

</module>

* Créez une référence à la source de donnée

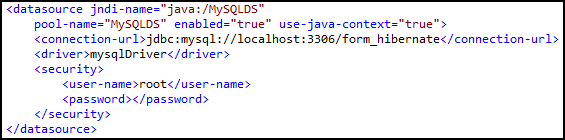
Editer le fichier **standalone.xml** du répertoire **JBossAS\_Home/standalone/configuration**

* Dans la partie **drivers** ajouter le code suivant



* Ajouter une source de donnée pour le driver

Dans la partie **datasources** ajouter le code suivant



## Etape : Deploy the EjbJpa

Lancer/Relancer JBoss

EJB project 🡺 Run As 🡺 Run On Server

## Etape : Configuration

Inclure le projet **EjbJpa** dans le path du projet **EjbJpaClient**

Inclure le jar **jboss\_client.jar**

Créez le fichier de configuration JBoss **jboss-ejb-client.properties**

Copier **ClientUtil.java** du projet **HelloEJBClient** vers le projet **EjbJpaClient** Attention : respectez l’arborescence des répertoires(**com.formation.util**).

## Etape : Créez d’un projet java Client

Créez un projet java simple

Nom Projet : **EjbJpaClient**

**Etape : Ecriture du client**

Créez une classe java **EJBApplicationClient.java** dans le package **com.formation.client**

**package** com.formation.client;

**import** java.util.List;

**import** javax.naming.Context;

**import** javax.naming.NamingException;

**import** com.formation.entity.Project;

**import** com.formation.service.IProject;

**import** com.formation.service.implementation.ProjectBean;

**import** com.formation.util.ClientUtil;

**public** **class** EJBApplicationClient {

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

IProject bean = *doLookup*();

// 4. Appel du service

Project p1 = **new** Project();

p1.setPname("Banking App");

p1.setPlocation("Town City");

p1.setDeptNo(1);

Project p2 = **new** Project();

p2.setPname("Office Automation");

p2.setPlocation("Downtown");

p2.setDeptNo(2);

// 4. Call business logic

//Saving new Projects

bean.saveProject(p1);

bean.saveProject(p2);

//Find a Project

p1.setPnumber(1);

Project p3 = bean.findProject(p1);

System.*out*.println(p3);

//Retrieve all projects

System.*out*.println("List of Projects:");

List<Project> projects = bean.retrieveAllProjects();

**for**(Project project : projects)

System.*out*.println(project);

}

**private** **static** IProject doLookup() {

Context context = **null**;

IProject bean = **null**;

**try** {

// 1. Obtenir le context

context = ClientUtil.*getInitialContext*();

// 2. Generate JNDI Lookup name

String lookupName = *getLookupName*();

// 3. Lookup and cast

bean = (IProject) context.lookup(lookupName);

} **catch** (NamingException e) {

e.printStackTrace();

}

**return** bean;

}

**private** **static** String getLookupName() {

// nom ear

String appName = "";

//nom du jar ejb (nom du projet dans eclipse par défaut)

String moduleName = "EjbJpa";

//nom optionnel JBoss7

String distinctName = "";

// Nom de la classe implementant le EJB

String beanName = ProjectBean.**class**.getSimpleName();

// Nom complet de l'interface

**final** String interfaceName = IProject.**class**.getName();

// Create a look up string name

String name = "ejb:" + appName + "/" + moduleName + "/"

+ distinctName + "/" + beanName + "!" + interfaceName;

**return** name;

}

}

## Etape : Test

Deployer le EJB

Lancer JBoss

Tester avec la classe **EJBApplicationClient.java** du projet **EjbJpaClient**

Dans la console du client

