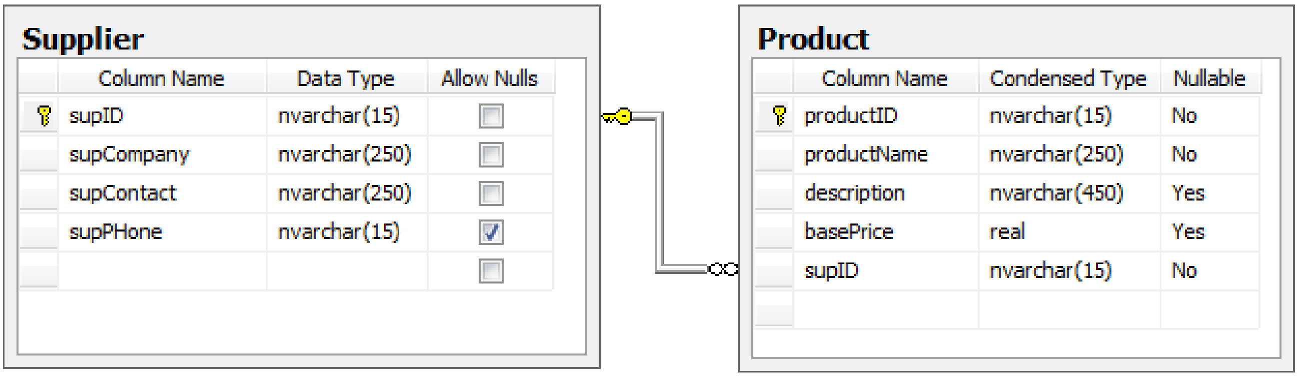
CMR – One-to-Many

Trong phần này chúng ta làm các bài demo liên quan đến CMR loại 1-1

1. Tạo database có tên CMR_EJB_Demos. Tạo 2 bảng Customer và CustomerDetails có cấu trúc như hình sau.



Supplier

Column Name	Data Type	Allow Nulls
supID	nvarchar(15)	□
supCompany	nvarchar(250)	□
supContact	nvarchar(250)	□
supPHone	nvarchar(15)	☑
		□

Product

Column Name	Condensed Type	Nullable
productID	nvarchar(15)	No
productName	nvarchar(250)	No
description	nvarchar(450)	Yes
basePrice	real	Yes
supID	nvarchar(15)	No

Nhập liệu 1 số mẫu tin thử nghiệm.

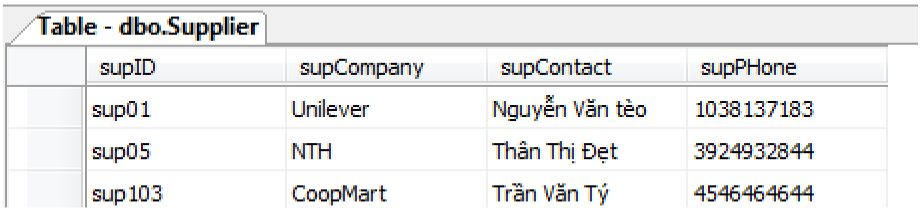


Table - dbo.Supplier

supID	supCompany	supContact	supPHone
sup01	Unilever	Nguyễn Văn tèo	1038137183
sup05	NTH	Thân Thị Đẹt	3924932844
sup103	CoopMart	Trần Văn Tý	4546464644

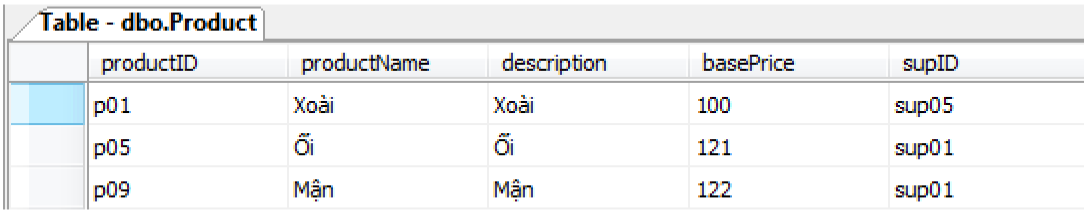
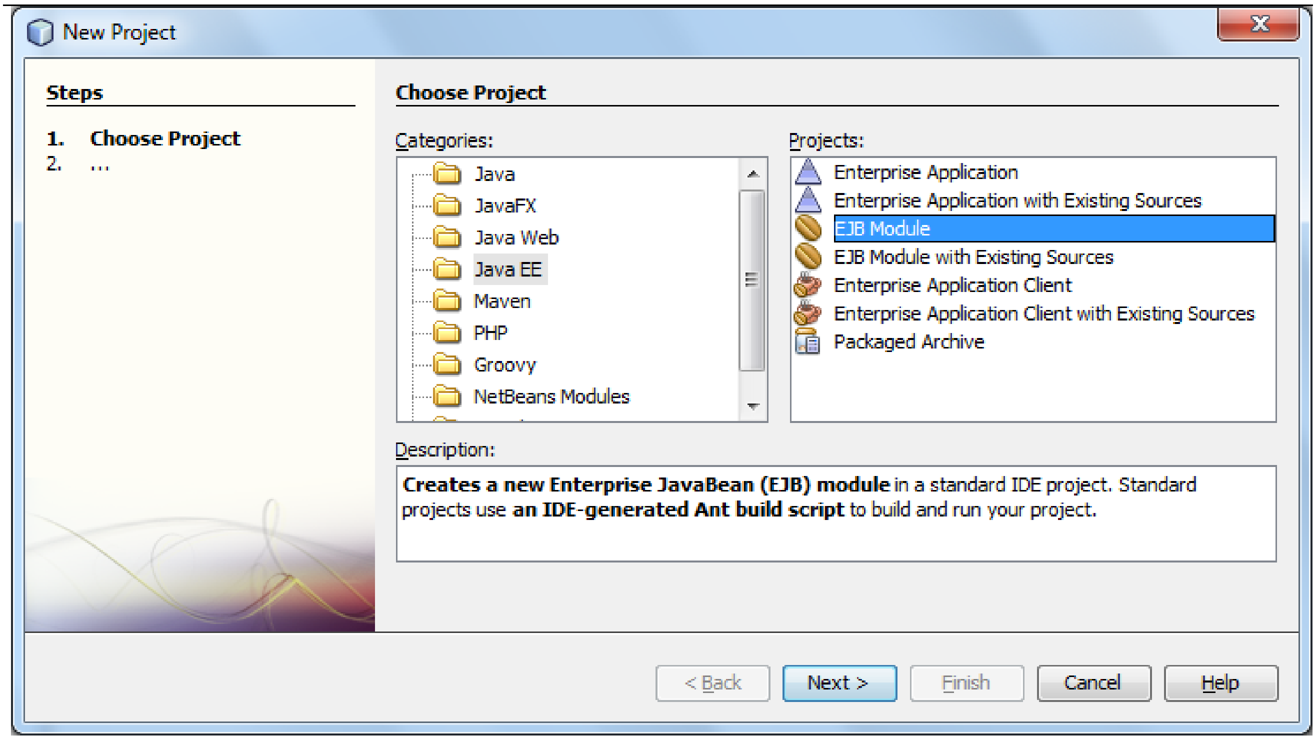


Table - dbo.Product

productID	productName	description	basePrice	supID
p01	Xoài	Xoài	100	sup05
p05	Ổi	Ổi	121	sup01
p09	Mận	Mận	122	sup01

2. Tạo project trong netbeans



New Project

Steps

1. Choose Project
2. ...

Choose Project

Categories:

- Java
- JavaFX
- Java Web
- Java EE
- Maven
- PHP
- Groovy
- NetBeans Modules

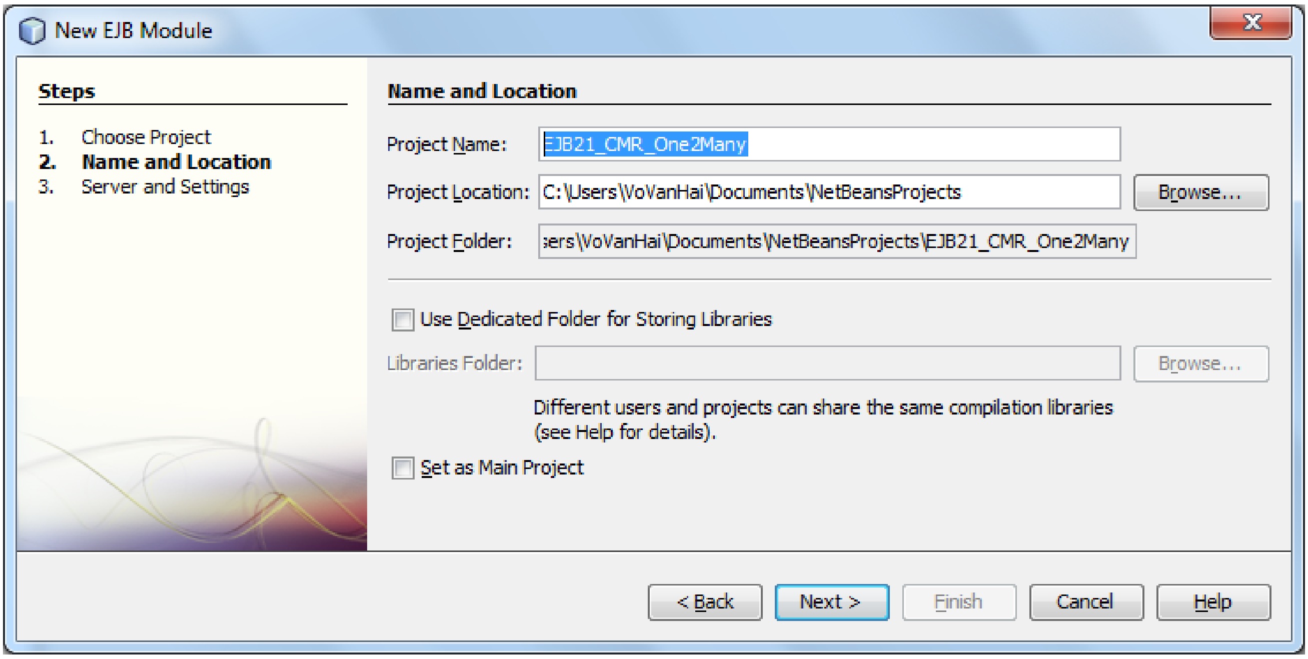
Projects:

- Enterprise Application
- Enterprise Application with Existing Sources
- EJB Module
- EJB Module with Existing Sources
- Enterprise Application Client
- Enterprise Application Client with Existing Sources
- Packaged Archive

Description:

Creates a new Enterprise JavaBean (EJB) module in a standard IDE project. Standard projects use **an IDE-generated Ant build script** to build and run your project.

Đặc tên project là: **EJB21_CMR_One2Many**



New EJB Module

Steps

1. Choose Project
2. Name and Location
3. Server and Settings

Name and Location

Project Name: EJB21_CMR_One2Many

Project Location: C:\Users\VoVanHai\Documents\NetBeansProjects

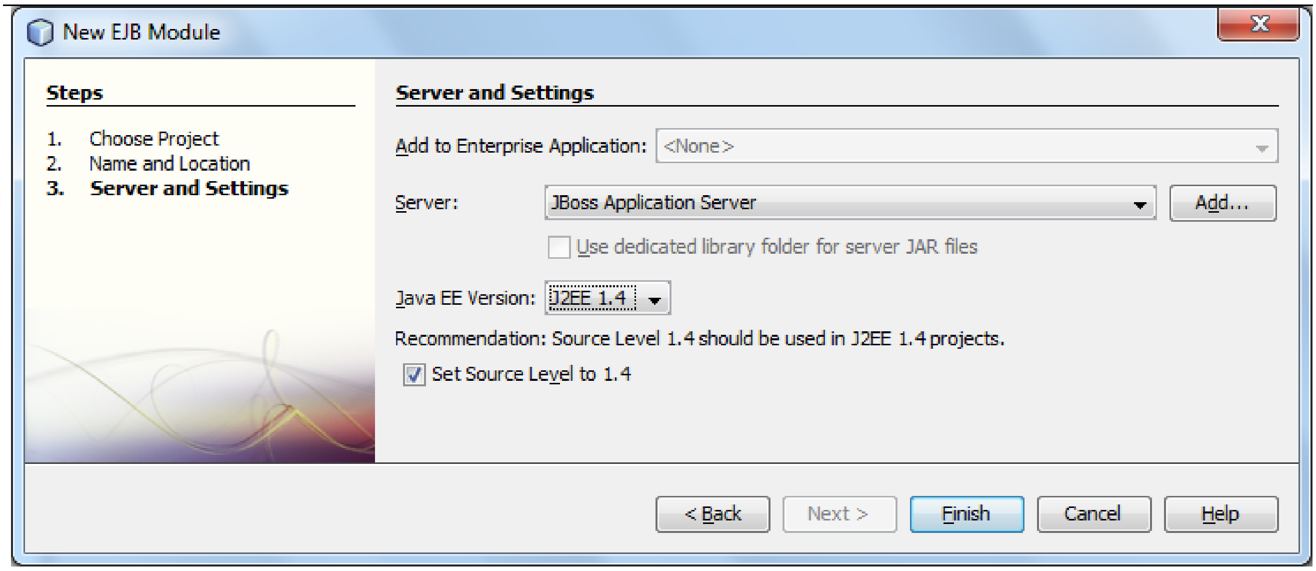
Project Folder: :ers\VoVanHai\Documents\NetBeansProjects\EJB21_CMR_One2Many

□ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

□ Set as Main Project



New EJB Module

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**

Server and Settings

Add to Enterprise Application: <None>

Server: JBoss Application Server

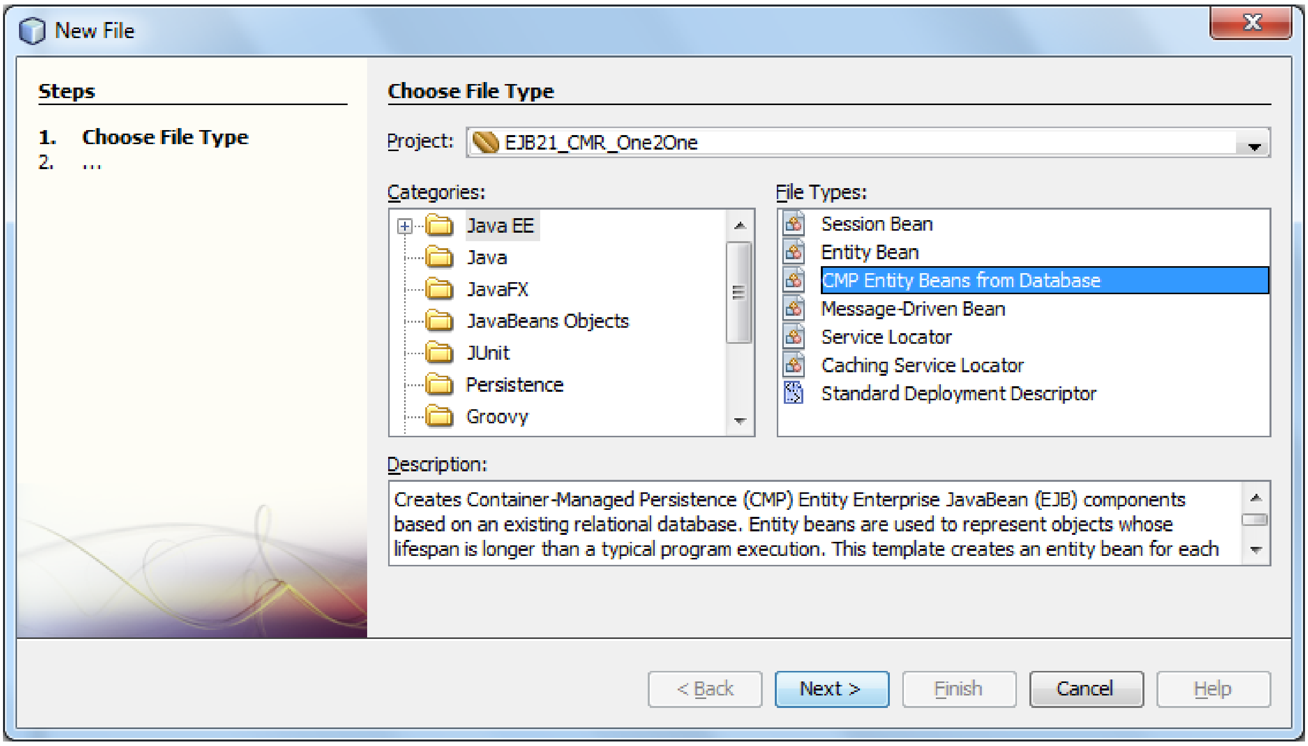
□ Use dedicated library folder for server JAR files

Java EE Version: J2EE 1.4

Recommendation: Source Level 1.4 should be used in J2EE 1.4 projects.

☑ Set Source Level to 1.4

Thêm vào project “CMP Entities Beans from Database” như hình



New File

Steps

1. **Choose File Type**
2. ...

Choose File Type

Project: EJB21_CMR_One2One

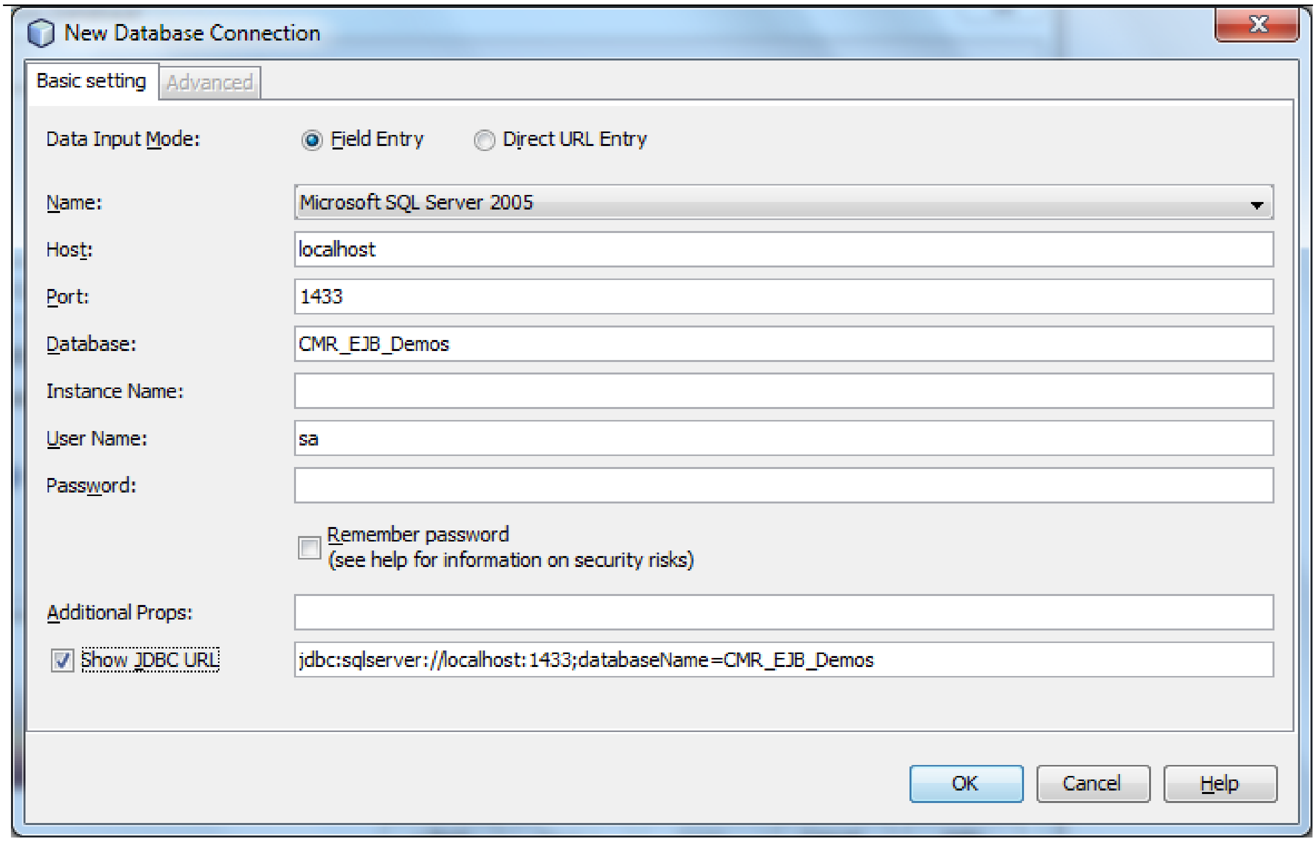
Categories: Java EE, Java, JavaFX, JavaBeans Objects, JUnit, Persistence, Groovy

File Types: Session Bean, Entity Bean, CMP Entity Beans from Database, Message-Driven Bean, Service Locator, Caching Service Locator, Standard Deployment Descriptor

Description:

Creates Container-Managed Persistence (CMP) Entity Enterprise JavaBean (EJB) components based on an existing relational database. Entity beans are used to represent objects whose lifespan is longer than a typical program execution. This template creates an entity bean for each

Nếu bạn đã làm bài CMR_One2One thì bạn không cần tạo 1 Connection mới mà bạn chỉ cần chọn trong danh sách các Connection.



New Database Connection

Data Input Mode: ● Field Entry ○ Direct URL Entry

Name: Microsoft SQL Server 2005

Host: localhost

Port: 1433

Database: CMR_EJB_Demos

Instance Name:

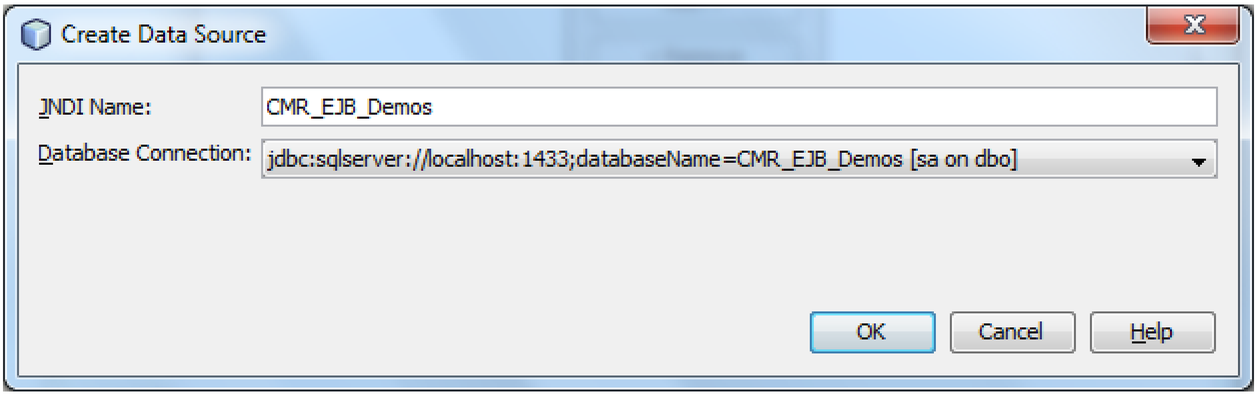
User Name: sa

Password:

□ Remember password
(see help for information on security risks)

Additional Props:

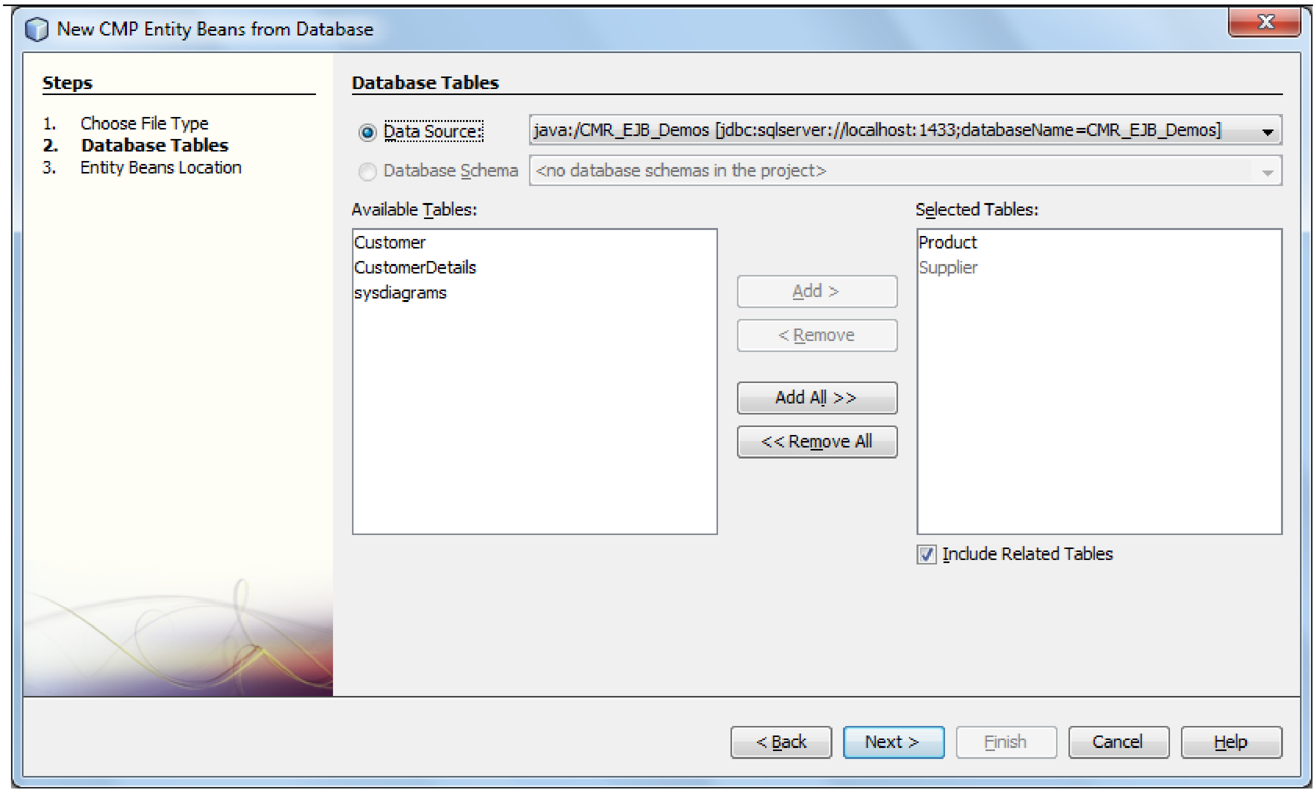
☑ Show JDBC URL jdbc:sqlserver://localhost:1433;databaseName=CMR_EJB_Demos



Create Data Source

JNDI Name: CMR_EJB_Demos

Database Connection: jdbc:sqlserver://localhost:1433;databaseName=CMR_EJB_Demos [sa on dbo]



New CMP Entity Beans from Database

Steps
1. Choose File Type
2. **Database Tables**
3. Entity Beans Location

Database Tables

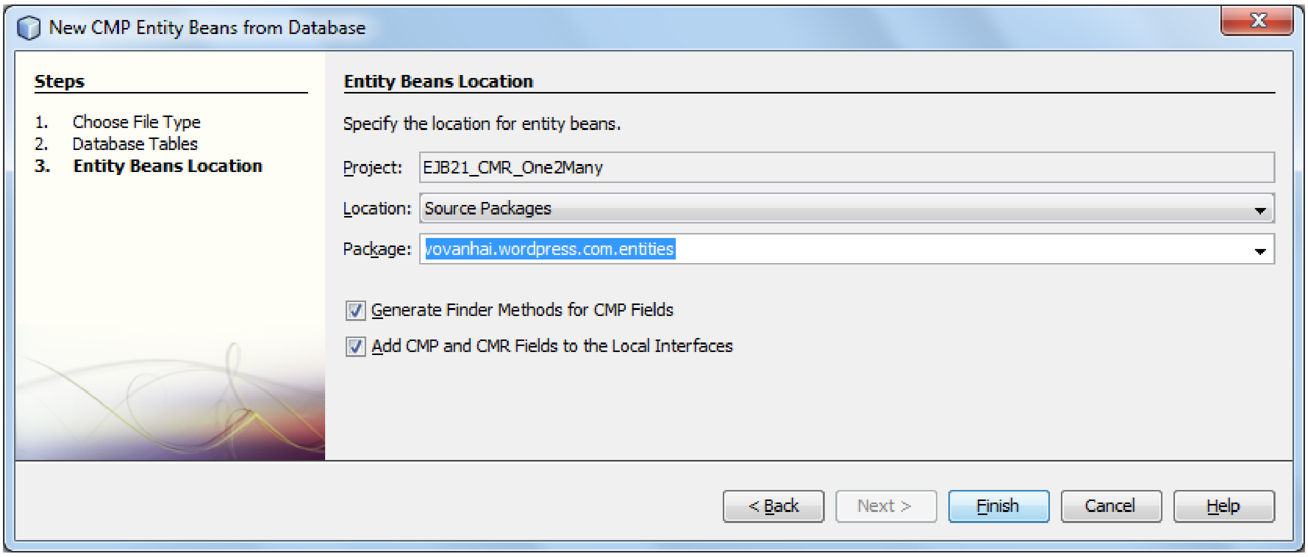
● Data Source: java:/CMR_EJB_Demos [jdbc:sqlserver://localhost:1433;databaseName=CMR_EJB_Demos]

○ Database Schema <no database schemas in the project>

Available Tables: Customer, CustomerDetails, sysdiagrams

Selected Tables: Product, Supplier

☑ Include Related Tables



New CMP Entity Beans from Database

Steps
1. Choose File Type
2. Database Tables
3. **Entity Beans Location**

Entity Beans Location

Specify the location for entity beans.

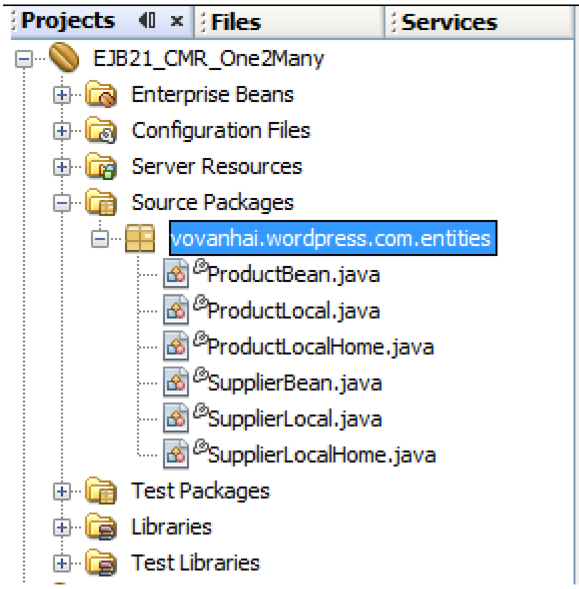
Project: EJB21_CMR_One2Many

Location: Source Packages

Package: vovanhai.wordpress.com.entities

☑ Generate Finder Methods for CMP Fields

☑ Add CMP and CMR Fields to the Local Interfaces



Code cho các lớp

Lớp SupplierLocalHome.java

```
package vovanhai.wordpress.com.entities;

import java.util.Collection;
import javax.ejb.CreateException;
import javax.ejb.EJBLocalHome;
import javax.ejb.FinderException;

/**
 *
 * @author VoVanHai
 */
public interface SupplierLocalHome extends EJBLocalHome {

    vovanhai.wordpress.com.entities.SupplierLocal findByPrimaryKey(String supID)
            throws FinderException;

    vovanhai.wordpress.com.entities.SupplierLocal create(String supID, String supCompany,
            String supContact, String supPHone) throws CreateException;

    Collection findBySupID(String supID) throws FinderException;

    Collection findBySupCompany(String supCompany) throws FinderException;

    Collection findBySupContact(String supContact) throws FinderException;

    Collection findBySupPHone(String supPHone) throws FinderException;

    Collection findAll() throws FinderException;
}
```

Lớp SupplierLocal.java

```
package vovanhai.wordpress.com.entities;
```

```
import java.util.Collection;
import javax.ejb.EJBLocalObject;

/**
 *
 * @author VoVanHai
 */
public interface SupplierLocal extends EJBLocalObject {

    String getSupID();

    void setSupID(String supID);

    String getSupCompany();

    void setSupCompany(String supCompany);

    String getSupContact();

    void setSupContact(String supContact);

    String getSupPHone();

    void setSupPHone(String supPHone);

    Collection getProductCollection();

    void setProductCollection(Collection productCollection);


}
```

Lớp SupplierBean.java. Sửa 2 phương thức ejbCreate và ejbPostCreate như sau

```
…
public java.lang.String ejbCreate(String supID, String supCompany,
            String supContact, String supPHone)  throws CreateException {
        if (supID == null) {
            throw new CreateException("The field \"key\" must not be null");
        }

        setSupID(supID);
        setSupCompany(supCompany);
        setSupContact(supContact);
        setSupPHone(supPHone);
        return null;
    }

    public void ejbPostCreate(String supID, String supCompany,
            String supContact, String supPHone) {
    }
…
```

Lớp ProductLocalHome.java

```
package vovanhai.wordpress.com.entities;

import java.util.Collection;
import javax.ejb.CreateException;
import javax.ejb.EJBLocalHome;
import javax.ejb.FinderException;

/**
 *
 * @author VoVanHai
 */
public interface ProductLocalHome extends EJBLocalHome {

    vovanhai.wordpress.com.entities.ProductLocal findByPrimaryKey(java.lang.String key) throws
FinderException;

    vovanhai.wordpress.com.entities.ProductLocal create(String productID,
            String productName, String description, Float basePrice) throws CreateException;

    vovanhai.wordpress.com.entities.ProductLocal create(String productID,
            String productName, String description, Float basePrice, SupplierLocal local) throws
CreateException;

    Collection findByProductID(String productID) throws FinderException;

    Collection findByProductName(String productName) throws FinderException;

    Collection findByDescription(String description) throws FinderException;

    Collection findByBasePrice(Float basePrice) throws FinderException;

    Collection findAll() throws FinderException;
}
```

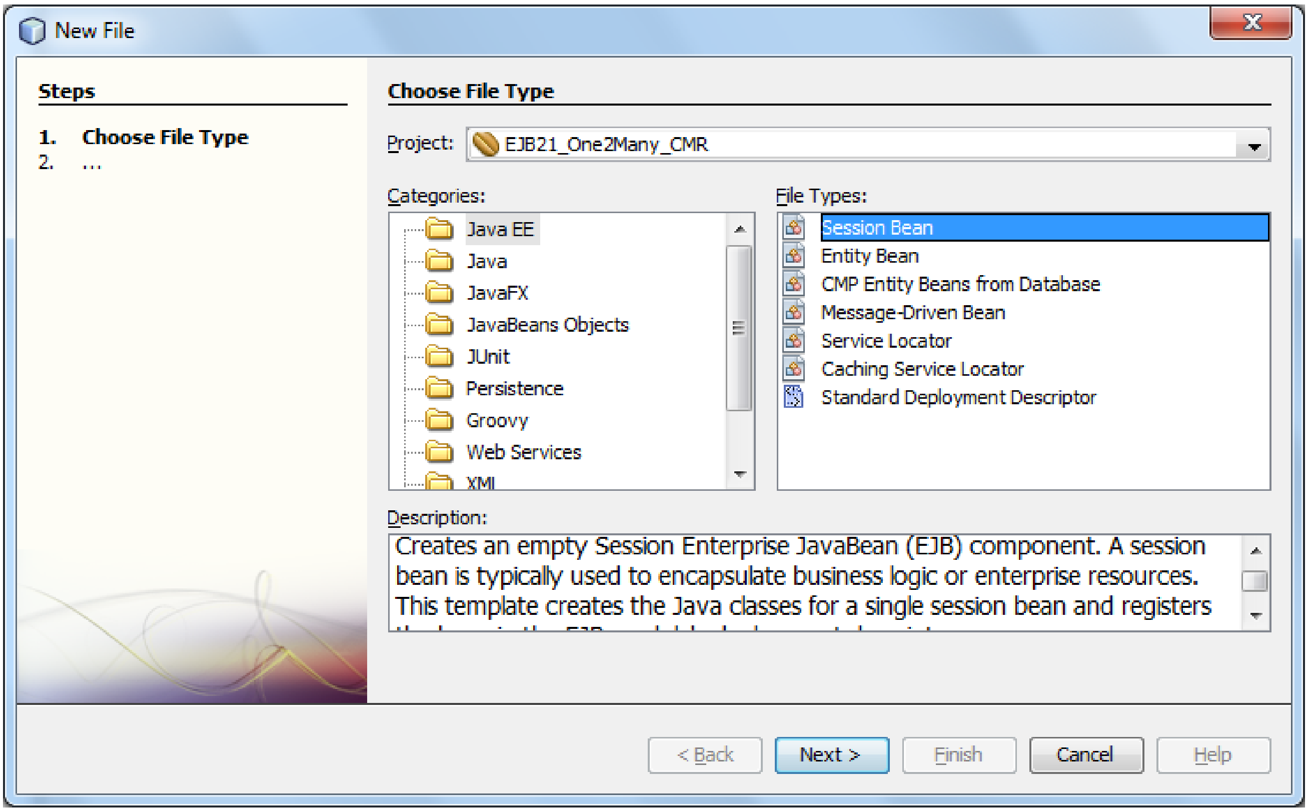
Lớp ProductLocal.java

```
package vovanhai.wordpress.com.entities;

import javax.ejb.EJBLocalObject;

/**
 *
 * @author VoVanHai
 */
public interface ProductLocal extends EJBLocalObject {

    String getProductID();

    void setProductID(String productID);

    String getProductName();

    void setProductName(String productName);
    String getDescription();

    void setDescription(String description);

    Float getBasePrice();

    void setBasePrice(Float basePrice);

    SupplierLocal getSupID();

    void setSupID(SupplierLocal supID);
}
```

Lớp ProductBean.java

```
…
public java.lang.String ejbCreate(String productID,
            String productName, String description, Float basePrice) throws CreateException {
        if (productID == null) {
            throw new CreateException("The field \"key\" must not be null");
        }

        setProductID(productID);
        setProductName(productName);
        setDescription(description);
        setBasePrice(basePrice);
        return null;
    }

    public void ejbPostCreate(String productID,
            String productName, String description, Float basePrice) {
        // TODO populate relationships here if appropriate
    }

    public java.lang.String ejbCreate(String productID,
            String productName, String description, Float basePrice, SupplierLocal local) throws
CreateException {
        if (productID == null) {
            throw new CreateException("The field \"key\" must not be null");
        }

        setProductID(productID);
        setProductName(productName);
        setDescription(description);
        setBasePrice(basePrice);

        return null;
    }

    public void ejbPostCreate(String productID,
            String productName, String description, Float basePrice, SupplierLocal local) {
        setSupID(local);
    }
…
```

Ở đây các entity đều là Local nên việc truy xuất nó từ client là không được. Ta tiến hành viết các SessionBean cho việc truy xuất dữ liệu từ EntityBean



New File

Steps

1. **Choose File Type**
2. ...

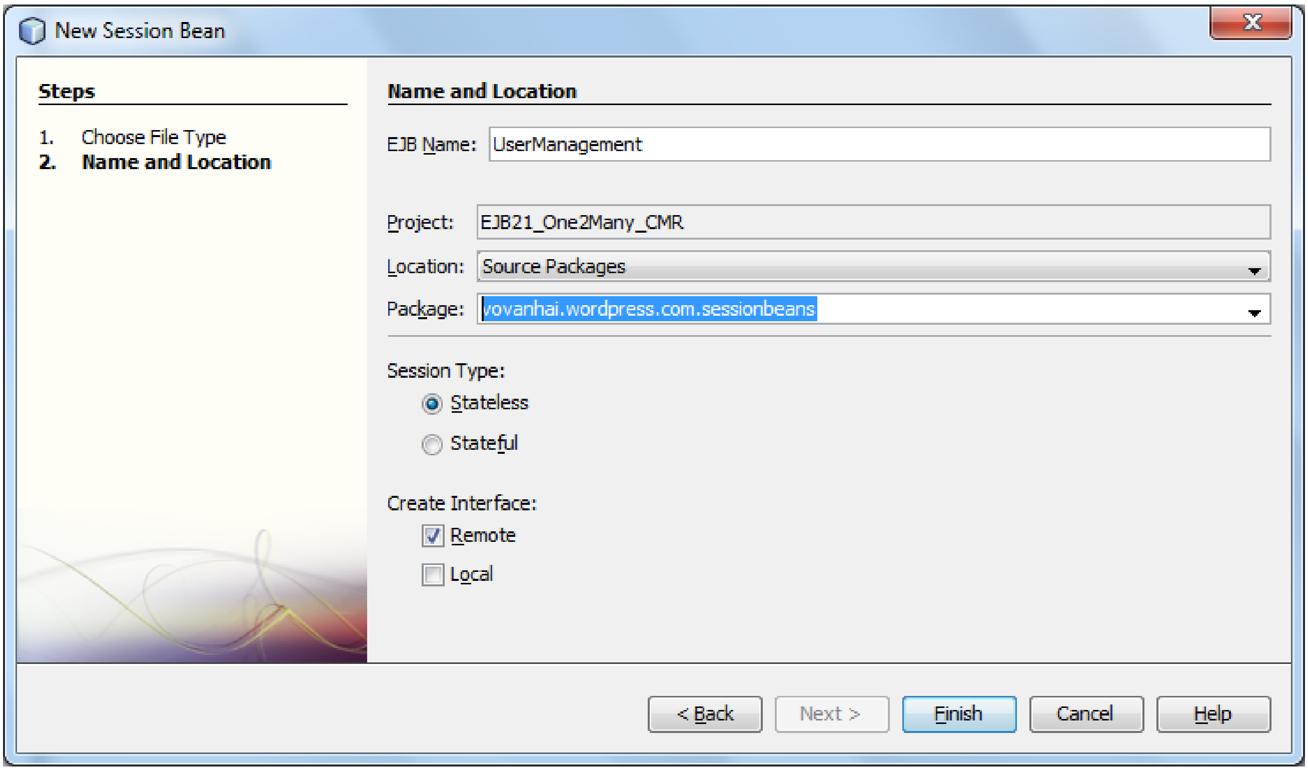
Choose File Type

Project: EJB21_One2Many_CMR

Categories:
- Java EE
- Java
- JavaFX
- JavaBeans Objects
- JUnit
- Persistence
- Groovy
- Web Services
- XML

File Types:
- Session Bean
- Entity Bean
- CMP Entity Beans from Database
- Message-Driven Bean
- Service Locator
- Caching Service Locator
- Standard Deployment Descriptor

Description:
Creates an empty Session Enterprise JavaBean (EJB) component. A session bean is typically used to encapsulate business logic or enterprise resources. This template creates the Java classes for a single session bean and registers



New Session Bean

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

EJB Name: UserManagement

Project: EJB21_One2Many_CMR

Location: Source Packages

Package: vovanhai.wordpress.com.sessionbeans

Session Type:

● Stateless

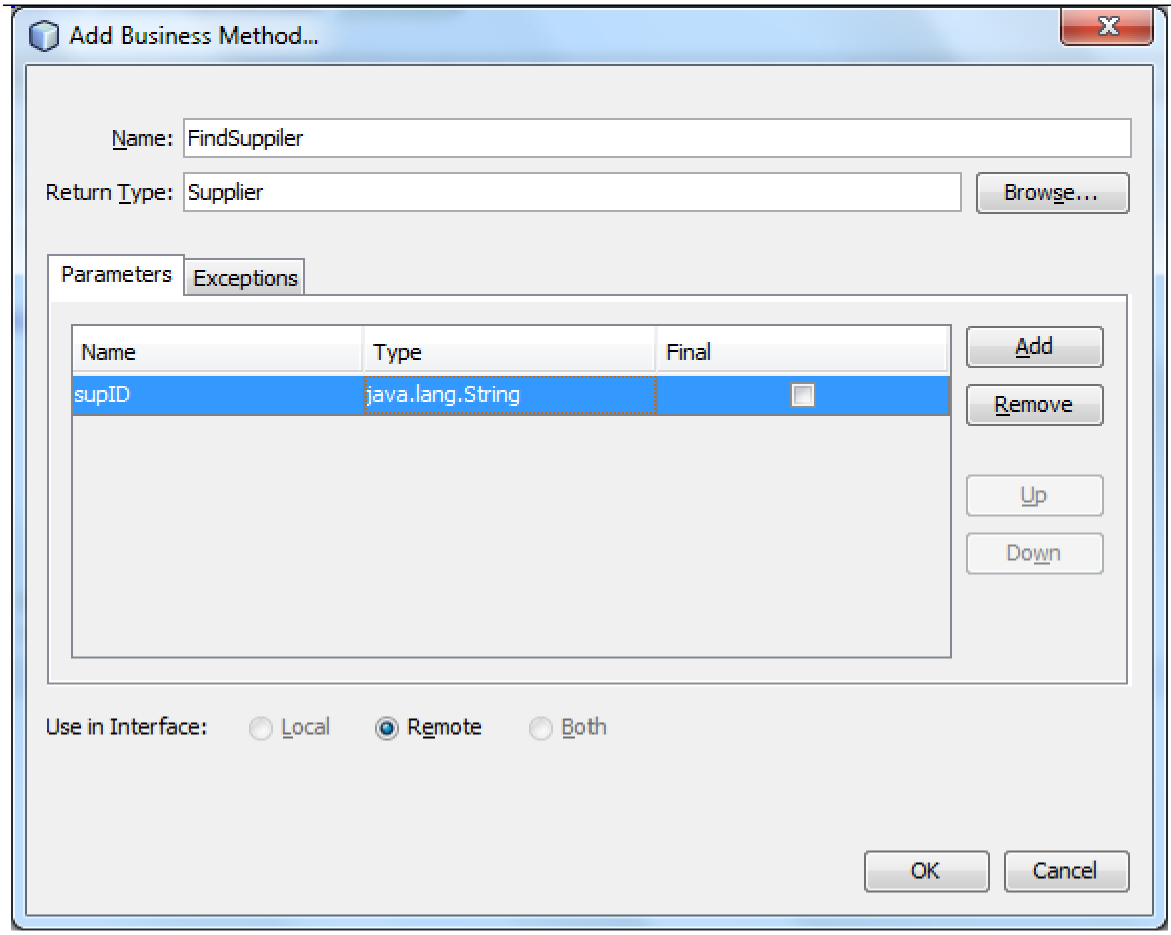
○ Stateful

Create Interface:

☑ Remote

□ Local

Thêm các Business Logic Methods



Name: FindSuppiler

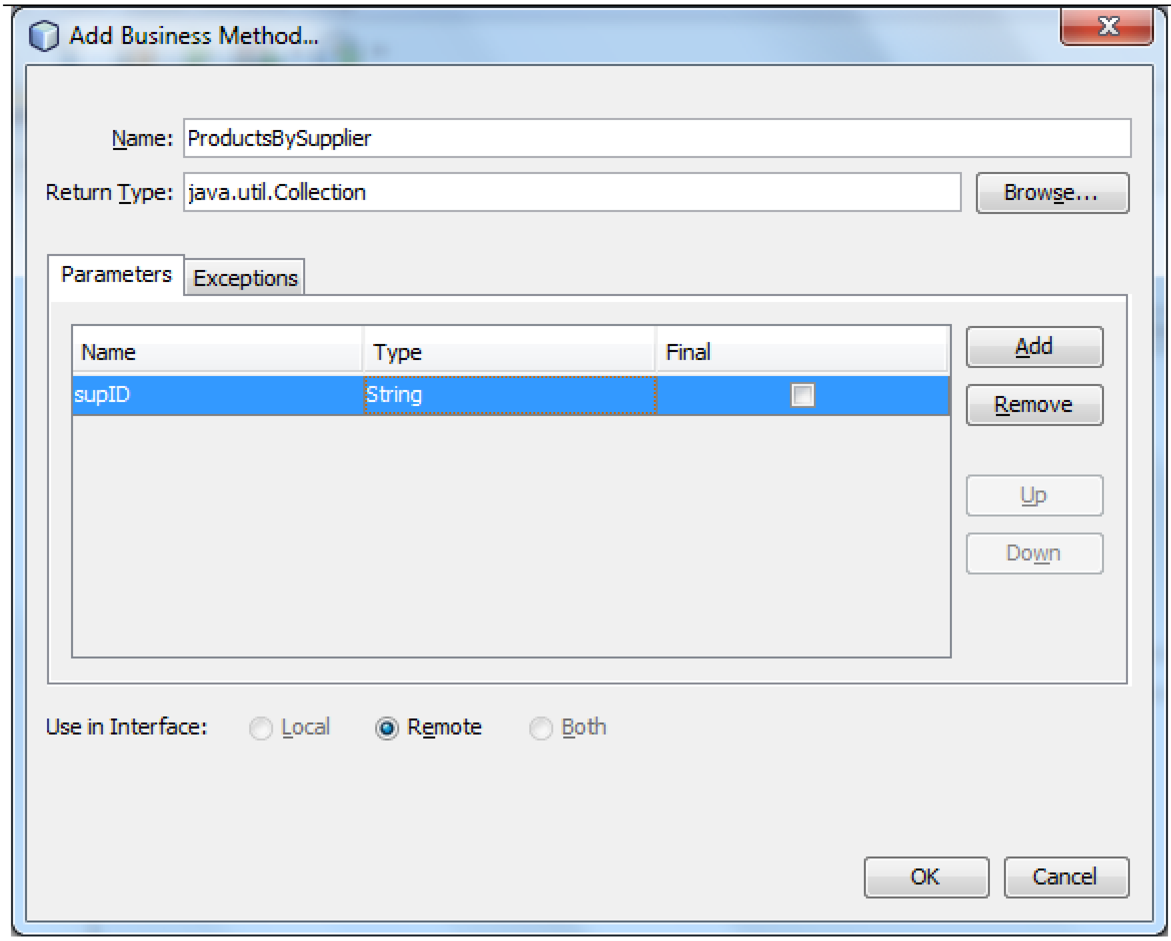
Return Type: Supplier

Name	Type	Final
supID	java.lang.String	□

Use in Interface: ○ Local ● Remote ○ Both

Để tìm 1 Supplier khi biết suppilerID. Với Suppiler là một lớp có code như sau

```
package vovanhai.wordpress.com.sessionbeans;

import java.io.Serializable;

/**
 *
 * @author VoVanHai
 */
public class Supplier implements Serializable {

    private String supID;
    private String supCompany;
    private String supContact;
    private String supPHone;

    public Supplier() {
    }

    public Supplier(String supID, String supCompany, String supContact, String supPHone) {
        this.supID = supID;
        this.supCompany = supCompany;
        this.supContact = supContact;
        this.supPHone = supPHone;
    }

    public String getSupCompany() {
        return supCompany;
    }
    public void setSupCompany(String supCompany) {
        this.supCompany = supCompany;
    }

    public String getSupContact() {
        return supContact;
    }

    public void setSupContact(String supContact) {
        this.supContact = supContact;
    }

    public String getSupID() {
        return supID;
    }

    public void setSupID(String supID) {
        this.supID = supID;
    }

    public String getSupPHone() {
        return supPHone;
    }

    public void setSupPHone(String supPHone) {
        this.supPHone = supPHone;
    }

    public boolean equals(Object obj) {
        if (obj == null) {
            return false;
        }
        if (getClass() != obj.getClass()) {
            return false;
        }
        final Supplier other = (Supplier) obj;
        if ((this.supID == null) ? (other.supID != null) : !this.supID.equals(other.supID)) {
            return false;
        }
        return true;
    }

    public int hashCode() {
        int hash = 7;
        hash = 89 * hash + (this.supID != null ? this.supID.hashCode() : 0);
        return hash;
    }

    public String toString() {
        return supCompany;
    }
}
```



Add Business Method...

Name: ProductsBySupplier

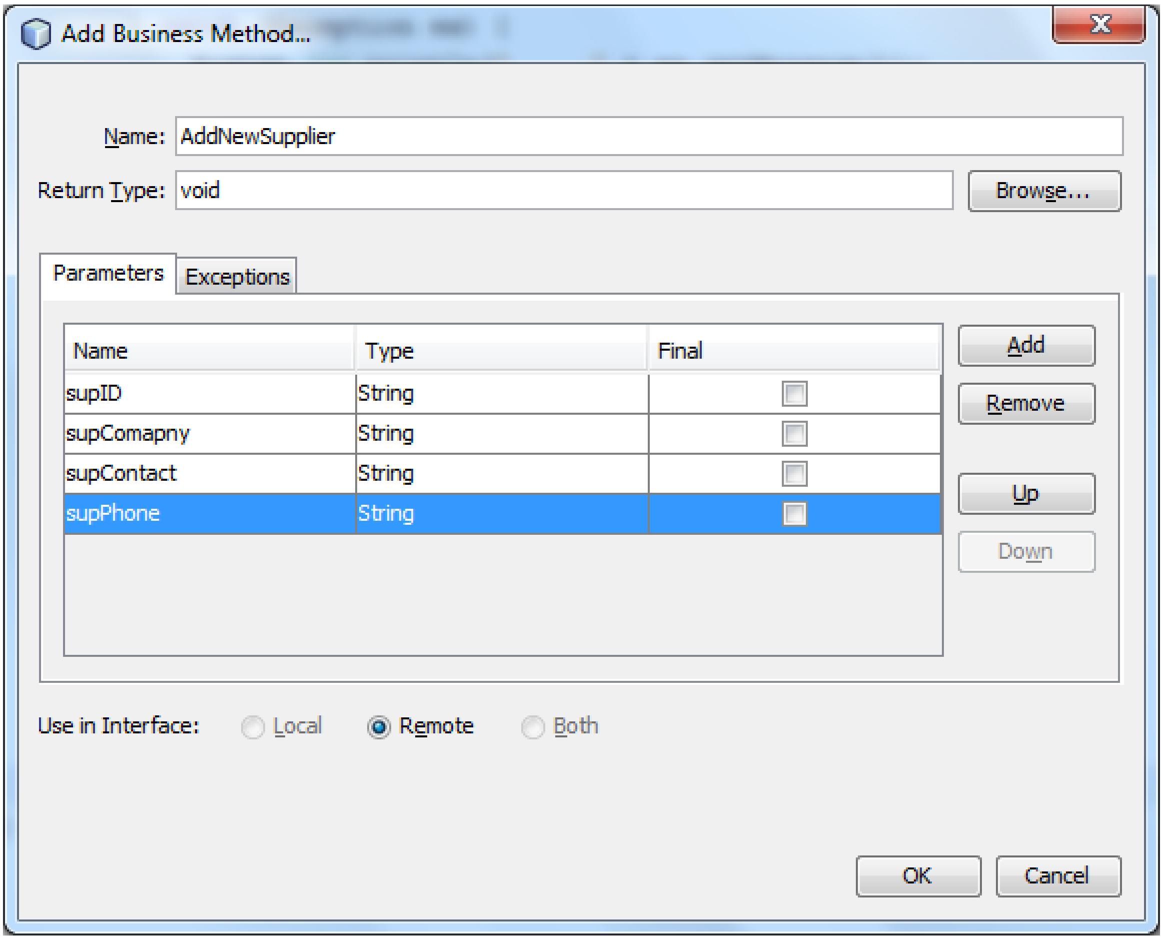
Return Type: java.util.Collection

Name	Type	Final
supID	String	☐

Use in Interface: ○ Local ● Remote ○ Both

Phương thức này lấy tất cả các mặt hàng của 1 supplier (tập các Product) trong đó Product được định nghĩa như sau:

```
package vovanhai.wordpress.com.sessionbeans;

import java.io.Serializable;

/**
 *
 * @author VoVanHai
 */
public class Product implements Serializable{

    private String productID;
    private String productName;
    private String description;
    private Float basePrice;
    private String supID;

    public Product() {
    }

    public Product(String productID, String productName,
            String description, Float basePrice, String supID) {
        this.productID = productID;
        this.productName = productName;
        this.description = description;
        this.basePrice = basePrice;
        this.supID = supID;
    }
    public Float getBasePrice() {
        return basePrice;
    }

    public void setBasePrice(Float basePrice) {
        this.basePrice = basePrice;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public String getProductID() {
        return productID;
    }

    public void setProductID(String productID) {
        this.productID = productID;
    }

    public String getProductName() {
        return productName;
    }

    public void setProductName(String productName) {
        this.productName = productName;
    }

    public String getSupID() {
        return supID;
    }

    public void setSupID(String supID) {
        this.supID = supID;
    }

    public boolean equals(Object obj) {
        if (obj == null) {
            return false;
        }
        if (getClass() != obj.getClass()) {
            return false;
        }
        final Product other = (Product) obj;
        if ((this.productID == null) ? (other.productID != null) : !this.productID.equals(other.productID)) {
            return false;
        }
        return true;
    }

    public int hashCode() {
        int hash = 3;
        hash = 97 * hash + (this.productID != null ? this.productID.hashCode() : 0);
        return hash;
    }

    public String toString() {
        return productName;
    }
}
```



Add Business Method...

Name: AddNewSupplier

Return Type: void

Name	Type	Final
supID	String	□
supComapny	String	□
supContact	String	□
supPhone	String	□

Use in Interface: ○ Local ● Remote ○ Both

Code cho các phương thức

```
…
public Supplier FindSuppiler(String supID) {
        Supplier sup = null;
        try {
            InitialContext ctx = new InitialContext();
            Object obj = ctx.lookup("SupplierBean");
            SupplierLocalHome home = (SupplierLocalHome) obj;
            SupplierLocal local = home.findByPrimaryKey(supID);
            if (local != null) {
                sup = new Supplier(local.getSupID(), local.getSupCompany(),
                        local.getSupContact(), local.getSupPHone());
            }
        } catch (Exception ex) {
            System.out.println("_____" + ex.getMessage());
        }
        return sup;
    }

    public Collection ProductsBySupplier(String supID) {
        Collection cols = null;
        Vector v=new Vector();
        try {
            InitialContext ctx = new InitialContext();
            Object obj = ctx.lookup("SupplierBean");
            SupplierLocalHome home = (SupplierLocalHome) obj;
            SupplierLocal local = home.findByPrimaryKey(supID);
            cols = local.getProductCollection();
            Iterator it=cols.iterator();

            while(it.hasNext()){
                ProductLocal pro=(ProductLocal)it.next();
                Product p=new Product(pro.getProductID(), pro.getProductName(), pro.getDescription(),
                        pro.getBasePrice(),pro.getSupID().getSupID());
                v.add(p);
            }

        } catch (Exception ex) {
            System.out.println("_____" + ex.getMessage());
        }

        return v;
    }

public void AddNewSupplier(String supID, String supComapny, String supContact, String
supPhone) {
        try {
            InitialContext ctx = new InitialContext();
            Object obj = ctx.lookup("SupplierBean");
            SupplierLocalHome home = (SupplierLocalHome) obj;
            SupplierLocal local = home.create(supID, supComapny, supContact, supPhone);
        } catch (Exception ex) {
            System.out.println("_____" + ex.getMessage());
        }
    }
…
```

Deployment descriptor(ejb-jar.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<ejb-jar version="2.1" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/ejb-
jar_2_1.xsd">
    <enterprise-beans>
        <session>
            <display-name>UserManagementSB</display-name>
            <ejb-name>UserManagementBean</ejb-name>
            <home>vovanhai.wordpress.com.sessionbeans.UserManagementRemoteHome</home>
            <remote>vovanhai.wordpress.com.sessionbeans.UserManagementRemote</remote>
            <ejb-class>vovanhai.wordpress.com.sessionbeans.UserManagementBean</ejb-class>
            <session-type>Stateless</session-type>
            <transaction-type>Container</transaction-type>
        </session>
        <entity>
            <description>jdbc:sqlserver://localhost:1433;databaseName=CMR_EJB_Demos [sa on
dbo]</description>
```

```
<display-name>SupplierEB</display-name>
<ejb-name>SupplierBean</ejb-name>
<local-home>vovanhai.wordpress.com.entities.SupplierLocalHome</local-home>
<local>vovanhai.wordpress.com.entities.SupplierLocal</local>
<ejb-class>vovanhai.wordpress.com.entities.SupplierBean</ejb-class>
<persistence-type>Container</persistence-type>
<prim-key-class>java.lang.String</prim-key-class>
<reentrant>false</reentrant>
<abstract-schema-name>Supplier</abstract-schema-name>
<cmp-field>
   <field-name>supID</field-name>
</cmp-field>
<cmp-field>
   <field-name>supCompany</field-name>
</cmp-field>
<cmp-field>
   <field-name>supContact</field-name>
</cmp-field>
<cmp-field>
   <field-name>supPHone</field-name>
</cmp-field>
<primkey-field>supID</primkey-field>
<query>
   <query-method>
      <method-name>findBySupID</method-name>
      <method-params>
         <method-param>java.lang.String</method-param>
      </method-params>
   </query-method>
   <ejb-ql>SELECT OBJECT(s) FROM Supplier AS s WHERE s.supID = ?1</ejb-ql>
</query>
<query>
   <query-method>
      <method-name>findBySupCompany</method-name>
      <method-params>
         <method-param>java.lang.String</method-param>
      </method-params>
   </query-method>
   <ejb-ql>SELECT OBJECT(s) FROM Supplier AS s WHERE s.supCompany = ?1</ejb-ql>
</query>
<query>
   <query-method>
      <method-name>findBySupContact</method-name>
      <method-params>
         <method-param>java.lang.String</method-param>
      </method-params>
   </query-method>
   <ejb-ql>SELECT OBJECT(s) FROM Supplier AS s WHERE s.supContact = ?1</ejb-ql>
</query>
<query>
   <query-method>
      <method-name>findBySupPHone</method-name>
      <method-params>
         <method-param>java.lang.String</method-param>
      </method-params>
   </query-method>
   <ejb-ql>SELECT OBJECT(s) FROM Supplier AS s WHERE s.supPHone = ?1</ejb-ql>
</query>
```

```
      <query>
        <description/>
        <query-method>
          <method-name>findAll</method-name>
          <method-params/>
        </query-method>
        <ejb-ql>select Object(o) from Supplier o</ejb-ql>
      </query>
    </entity>
    <entity>
      <description>jdbc:sqlserver://localhost:1433;databaseName=CMR_EJB_Demos [sa on
dbo]</description>
      <display-name>ProductEB</display-name>
      <ejb-name>ProductBean</ejb-name>
      <local-home>vovanhai.wordpress.com.entities.ProductLocalHome</local-home>
      <local>vovanhai.wordpress.com.entities.ProductLocal</local>
      <ejb-class>vovanhai.wordpress.com.entities.ProductBean</ejb-class>
      <persistence-type>Container</persistence-type>
      <prim-key-class>java.lang.String</prim-key-class>
      <reentrant>false</reentrant>
      <abstract-schema-name>Product</abstract-schema-name>
      <cmp-field>
        <field-name>productID</field-name>
      </cmp-field>
      <cmp-field>
        <field-name>productName</field-name>
      </cmp-field>
      <cmp-field>
        <field-name>description</field-name>
      </cmp-field>
      <cmp-field>
        <field-name>basePrice</field-name>
      </cmp-field>
      <primkey-field>productID</primkey-field>
      <query>
        <query-method>
          <method-name>findByProductID</method-name>
          <method-params>
            <method-param>java.lang.String</method-param>
          </method-params>
        </query-method>
        <ejb-ql>SELECT OBJECT(p) FROM Product AS p WHERE p.productID = ?1</ejb-ql>
      </query>
      <query>
        <query-method>
          <method-name>findByProductName</method-name>
          <method-params>
            <method-param>java.lang.String</method-param>
          </method-params>
        </query-method>
        <ejb-ql>SELECT OBJECT(p) FROM Product AS p WHERE p.productName = ?1</ejb-ql>
      </query>
      <query>
        <query-method>
          <method-name>findByDescription</method-name>
          <method-params>
            <method-param>java.lang.String</method-param>
          </method-params>
        </query-method>
        <ejb-ql>SELECT OBJECT(p) FROM Product AS p WHERE p.description = ?1</ejb-ql>
      </query>
      <query>
        <query-method>
          <method-name>findByBasePrice</method-name>
          <method-params>
            <method-param>java.lang.Float</method-param>
          </method-params>
        </query-method>
        <ejb-ql>SELECT OBJECT(p) FROM Product AS p WHERE p.basePrice = ?1</ejb-ql>
      </query>
      <query>
        <description/>
        <query-method>
          <method-name>findAll</method-name>
          <method-params/>
        </query-method>
        <ejb-ql>select Object(o) from Product o</ejb-ql>
      </query>
    </entity>
  </enterprise-beans>
  <relationships>
    <ejb-relation>
      <ejb-relation-name>Product-Supplier</ejb-relation-name>
      <ejb-relationship-role>
        <ejb-relationship-role-name>Supplier</ejb-relationship-role-name>
        <multiplicity>One</multiplicity>
        <!--<cascade-delete/>-->
        <relationship-role-source>
          <ejb-name>SupplierBean</ejb-name>
        </relationship-role-source>
        <cmr-field>
          <cmr-field-name>productCollection</cmr-field-name>
          <cmr-field-type>java.util.Collection</cmr-field-type>
        </cmr-field>
      </ejb-relationship-role>
      <ejb-relationship-role>
        <ejb-relationship-role-name>Product</ejb-relationship-role-name>
        <multiplicity>Many</multiplicity>
        <relationship-role-source>
          <ejb-name>ProductBean</ejb-name>
        </relationship-role-source>
        <cmr-field>
          <cmr-field-name>supID</cmr-field-name>
        </cmr-field>
      </ejb-relationship-role>

    </ejb-relation>
  </relationships>
  <assembly-descriptor>
    <container-transaction>
      <method>
        <ejb-name>SupplierBean</ejb-name>
        <method-name>*</method-name>
      </method>
      <trans-attribute>Required</trans-attribute>
    </container-transaction>
    <container-transaction>
      <method>
        <ejb-name>ProductBean</ejb-name>
        <method-name>*</method-name>
      </method>
      <trans-attribute>Required</trans-attribute>
    </container-transaction>
    <container-transaction>
      <method>
        <ejb-name>UserManagementBean</ejb-name>
        <method-name>*</method-name>
      </method>
      <trans-attribute>Required</trans-attribute>
    </container-transaction>
    </assembly-descriptor>
</ejb-jar>
```

File jboss.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jboss PUBLIC "-/JBoss//DTD JBOSS 4.0//EN"
"http://www.jboss.org/j2ee/dtd/jboss_4_0.dtd">
<jboss>
  <enterprise-beans>
    <entity>
      <ejb-name>SupplierBean</ejb-name>
      <local-jndi-name>SupplierBean</local-jndi-name>
    </entity>
    <entity>
      <ejb-name>ProductBean</ejb-name>
      <local-jndi-name>ProductBean</local-jndi-name>
    </entity>
  </enterprise-beans>

</jboss>
```

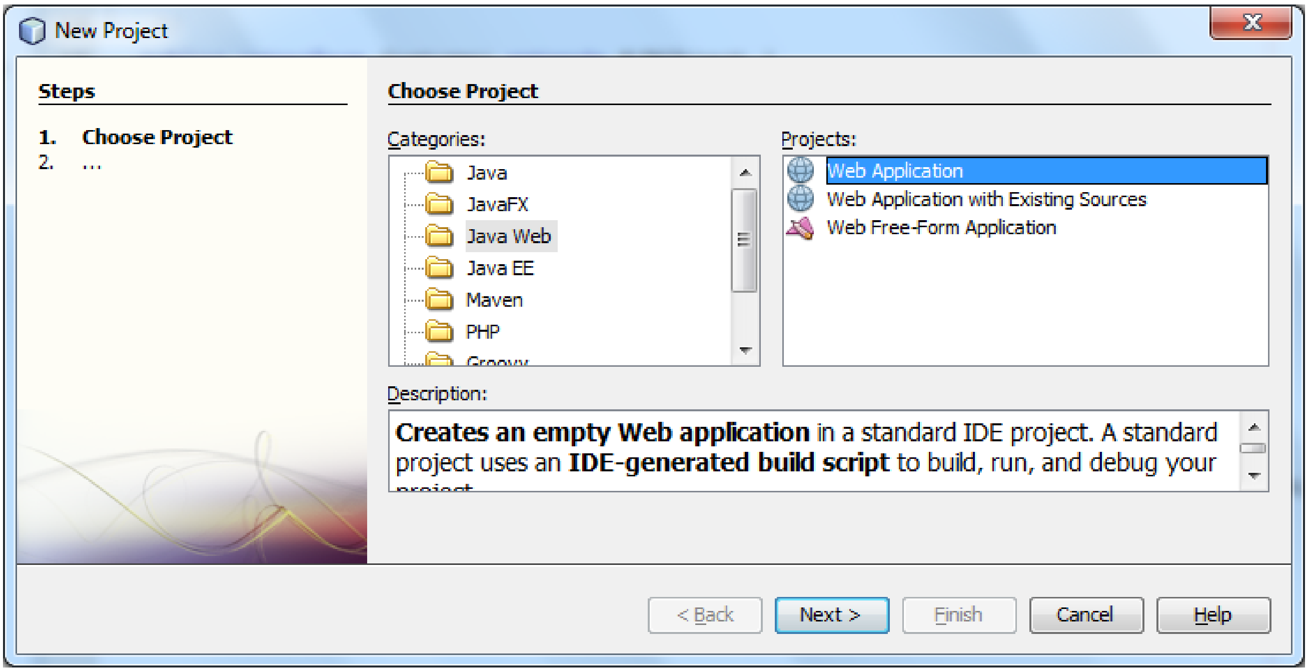
File jbosscmp-jdbc.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jbosscmp-jdbc PUBLIC "-//JBoss//DTD JBOSSCMP-JDBC 4.0//EN"
"http://www.jboss.org/j2ee/dtd/jbosscmp-jdbc_4_0.dtd">

<jbosscmp-jdbc>
  <defaults>
    <datasource>java:/CMR_EJB_Demos</datasource>
    <datasource-mapping>MS SQLSERVER</datasource-mapping>
    <create-table>false</create-table>
    <remove-table>false</remove-table>
  </defaults>
  <enterprise-beans>
    <entity>
      <ejb-name>SupplierBean</ejb-name>
      <datasource>java:/CMR_EJB_Demos</datasource>
      <datasource-mapping>MS SQLSERVER</datasource-mapping>
      <create-table>false</create-table>
      <table-name>Supplier</table-name>
      <cmp-field>
        <field-name>supID</field-name>
        <column-name>supID</column-name>
      </cmp-field>
      <cmp-field>
        <field-name>supCompany</field-name>
        <column-name>supCompany</column-name>
      </cmp-field>
      <cmp-field>
        <field-name>supContact</field-name>
        <column-name>supContact</column-name>
      </cmp-field>
      <cmp-field>
        <field-name>supPHone</field-name>
        <column-name>supPHone</column-name>
      </cmp-field>

    </entity>
    <entity>
      <ejb-name>ProductBean</ejb-name>
      <datasource>java:/CMR_EJB_Demos</datasource>
      <datasource-mapping>MS SQLSERVER</datasource-mapping>
      <create-table>false</create-table>
      <table-name>Product</table-name>
      <cmp-field>
        <field-name>productID</field-name>
        <column-name>productID</column-name>
      </cmp-field>
      <cmp-field>
        <field-name>productName</field-name>
        <column-name>productName</column-name>
      </cmp-field>
      <cmp-field>
        <field-name>description</field-name>
        <column-name>description</column-name>
      </cmp-field>
      <cmp-field>
        <field-name>basePrice</field-name>
        <column-name>basePrice</column-name>
      </cmp-field>

    </entity>
  </enterprise-beans>
  <relationships>
    <ejb-relation>
     <ejb-relation-name>Product-Supplier</ejb-relation-name>
      <ejb-relationship-role>
        <ejb-relationship-role-name>Supplier</ejb-relationship-role-name>
        <key-fields>
          <key-field>
            <field-name>supID</field-name>
            <column-name>supID</column-name>
          </key-field>
        </key-fields>
      </ejb-relationship-role>
      <ejb-relationship-role>
        <ejb-relationship-role-name>Product</ejb-relationship-role-name>
        <key-fields/>
```

```
            </ejb-relationship-role>
        </ejb-relation>
    </relationships>
</jbosscmp-jdbc>
```

Tạo client

Tạo 1 web application có tên **EJB21_CMR_One2Many_client**



New Project

Steps

1. **Choose Project**
2. ...

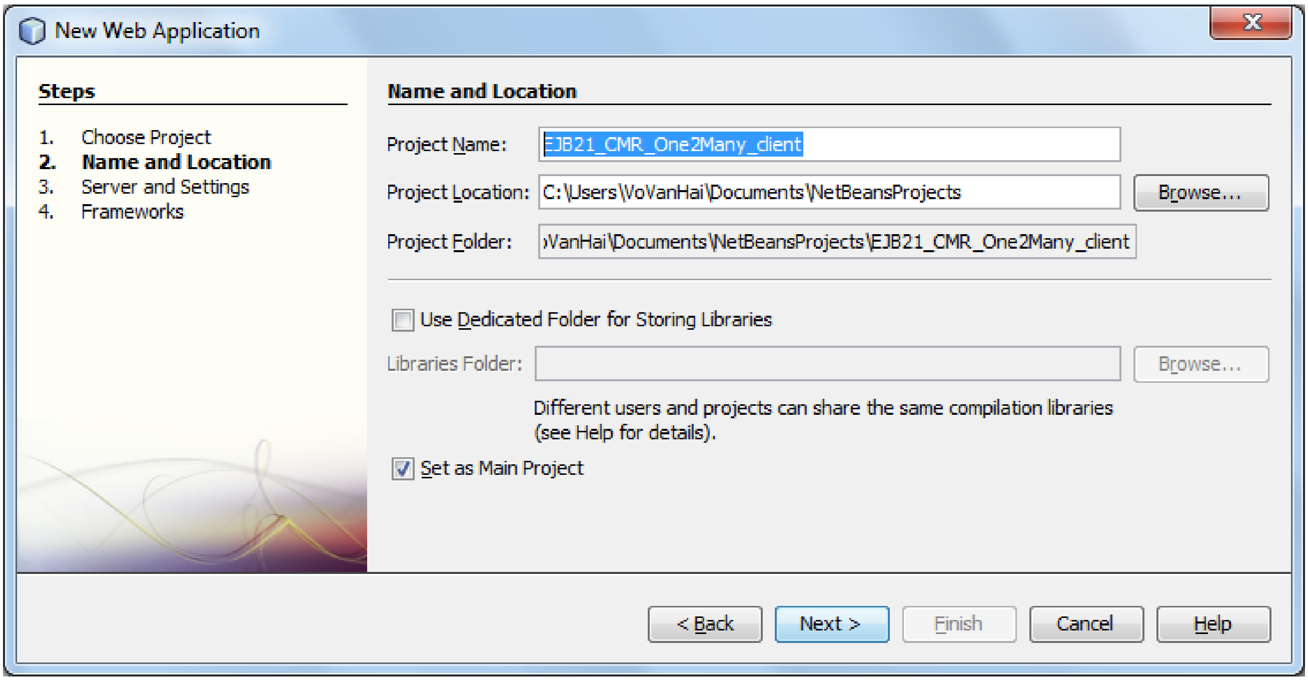
Choose Project

Categories: Java, JavaFX, Java Web, Java EE, Maven, PHP, Groovy

Projects: Web Application; Web Application with Existing Sources; Web Free-Form Application

Description:

Creates an empty Web application in a standard IDE project. A standard project uses an **IDE-generated build script** to build, run, and debug your project.



New Web Application

Steps

1. Choose Project
2. **Name and Location**
3. Server and Settings
4. Frameworks

Name and Location

Project Name: EJB21_CMR_One2Many_client

Project Location: C:\Users\VoVanHai\Documents\NetBeansProjects

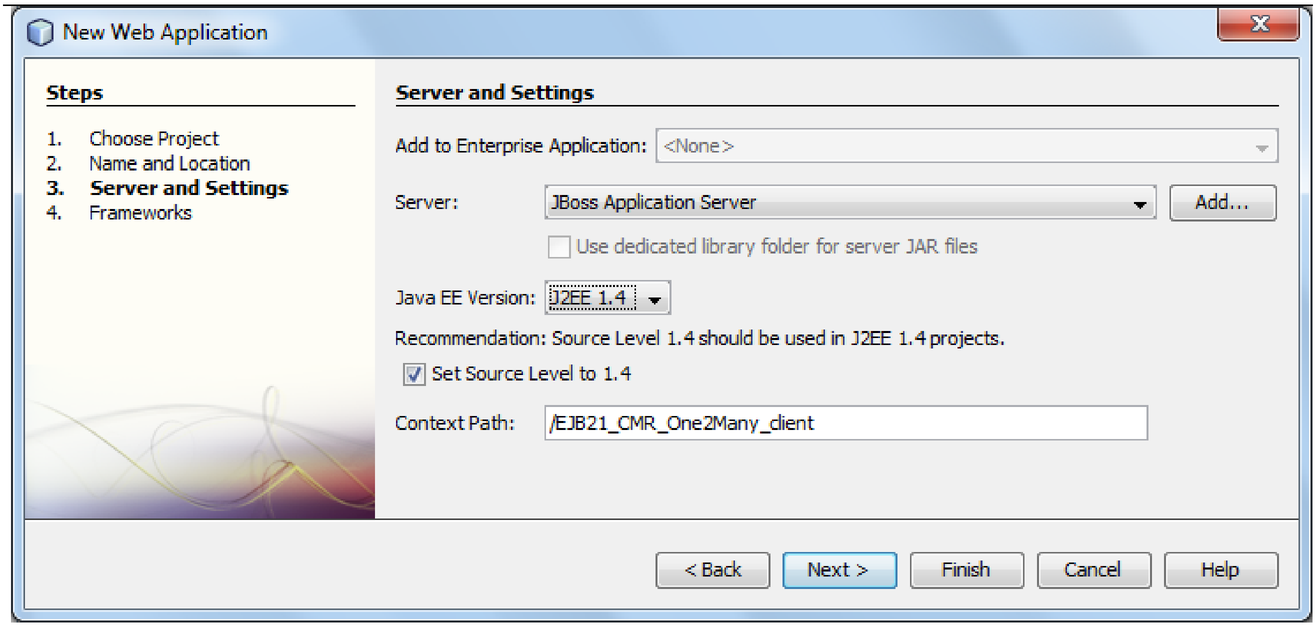
Project Folder: VanHai\Documents\NetBeansProjects\EJB21_CMR_One2Many_client

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☑ Set as Main Project



New Web Application

Steps

1. Choose Project
2. Name and Location
3. **Server and Settings**
4. Frameworks

Server and Settings

Add to Enterprise Application: <None>

Server: JBoss Application Server

□ Use dedicated library folder for server JAR files

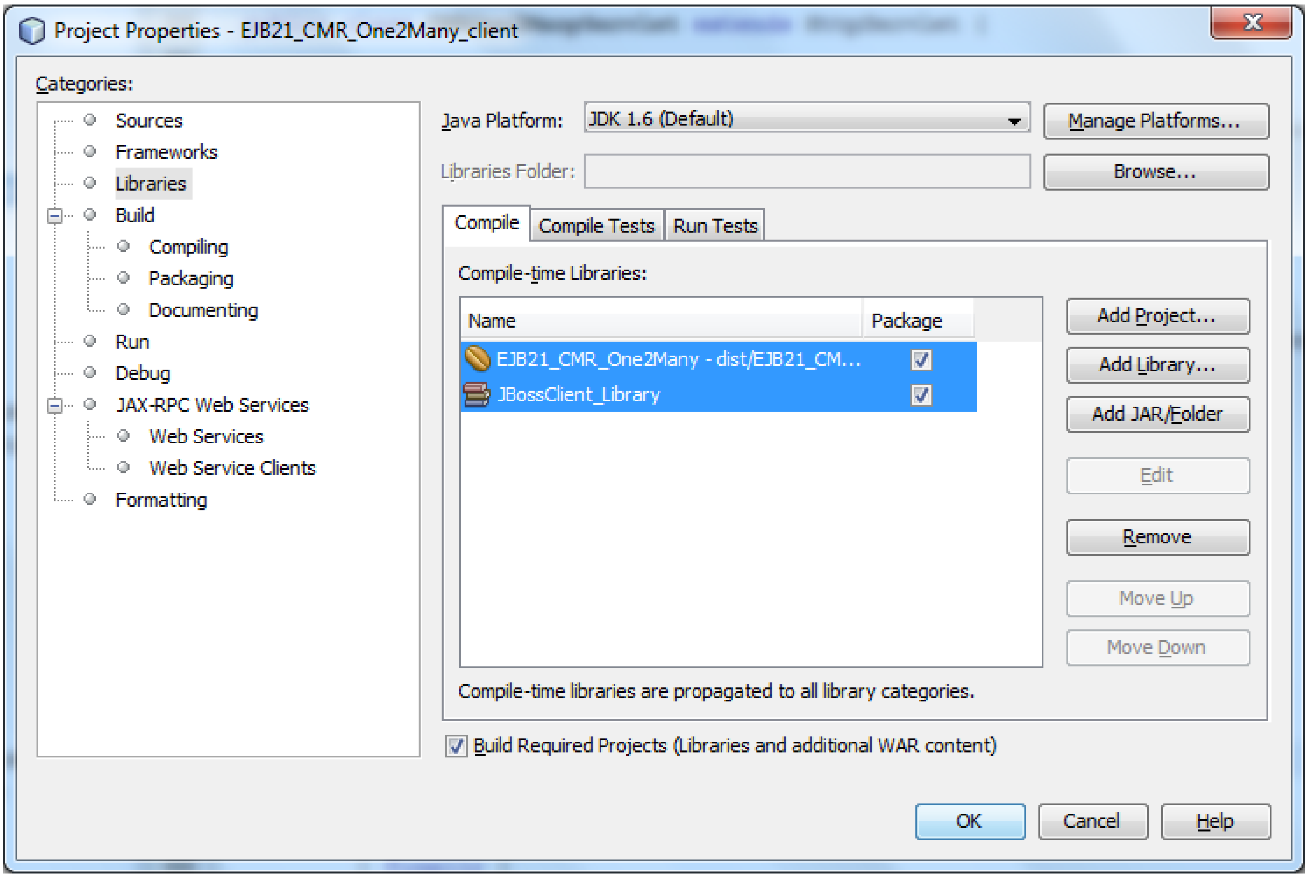
Java EE Version: J2EE 1.4

Recommendation: Source Level 1.4 should be used in J2EE 1.4 projects.

☑ Set Source Level to 1.4

Context Path: /EJB21_CMR_One2Many_client

Thêm tham chiếu cho project



Project Properties - EJB21_CMR_One2Many_client

Categories:

- Sources
- Frameworks
- Libraries
- Build
  - Compiling
  - Packaging
  - Documenting
- Run
- Debug
- JAX-RPC Web Services
  - Web Services
  - Web Service Clients
- Formatting

Java Platform: JDK 1.6 (Default)

Libraries Folder:

Compile | Compile Tests | Run Tests

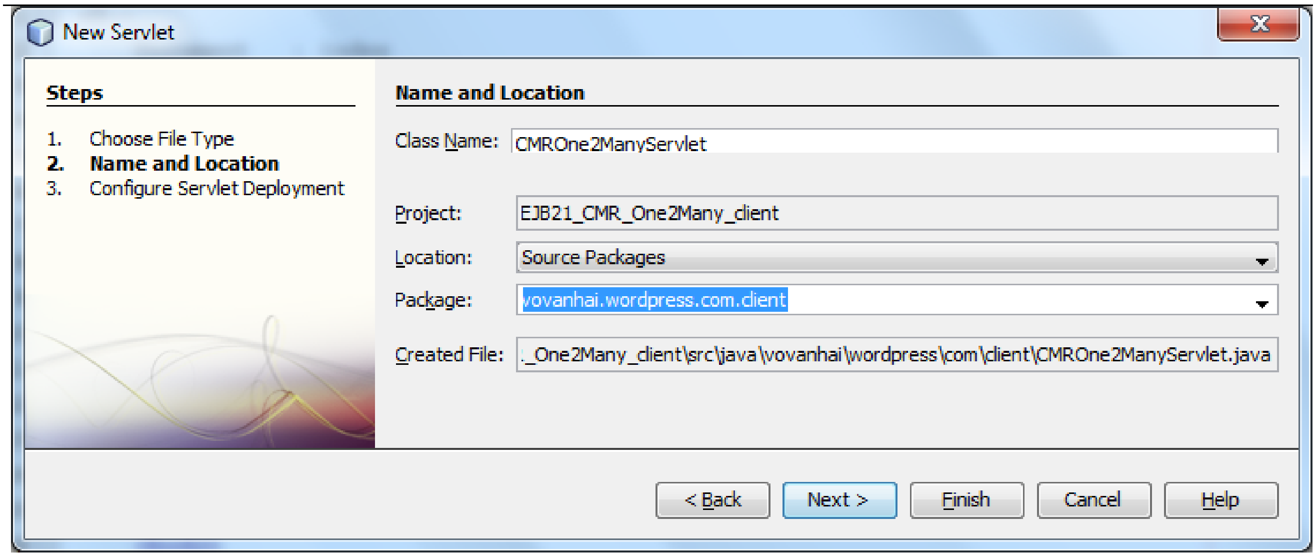
Compile-time Libraries:

Name	Package
EJB21_CMR_One2Many - dist/EJB21_CM...	☑
JBossClient_Library	☑

Compile-time libraries are propagated to all library categories.

☑ Build Required Projects (Libraries and additional WAR content)

Tạo servlet có tên CMROne2ManyServlet



New Servlet

Steps

1. Choose File Type
2. **Name and Location**
3. Configure Servlet Deployment

Name and Location

Class Name: CMROne2ManyServlet

Project: EJB21_CMR_One2Many_client

Location: Source Packages

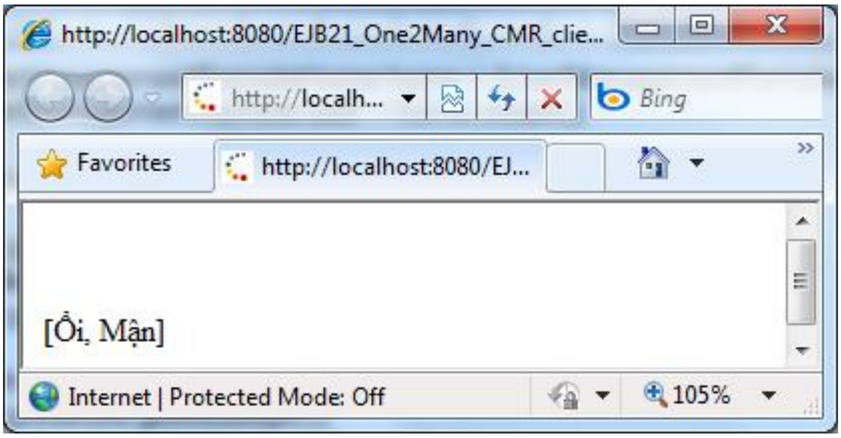
Package: vovanhai.wordpress.com.client

Created File: ..._One2Many_client\src\java\vovanhai\wordpress\com\client\CMROne2ManyServlet.java

Code cho phần processRequest như sau

```
protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            System.setProperty(Context.INITIAL_CONTEXT_FACTORY,
                    "org.jnp.interfaces.NamingContextFactory");
            System.setProperty(Context.PROVIDER_URL, "localhost:1099");
            InitialContext ctx = new InitialContext();
            Object obj1 = ctx.lookup("UserManagementBean");
            UserManagementRemoteHome home = (UserManagementRemoteHome)
obj1;
            UserManagementRemote user = home.create();

            Supplier sup = user.FindSuppiler("sup05");
            out.print("<br/>");
            Collection pros=user.ProductsBySupplier("sup01");
            out.print("<br/>"+pros);
        } catch (Exception ex) {
            out.println(ex.getMessage());
        } finally {
            out.close();
        }
    }
```

Kết quả:



http://localhost:8080/EJB21_One2Many_CMR_clie...
http://localh...
Bing
Favorites
http://localhost:8080/EJ...

[Ổi, Mận]

Internet | Protected Mode: Off
105%