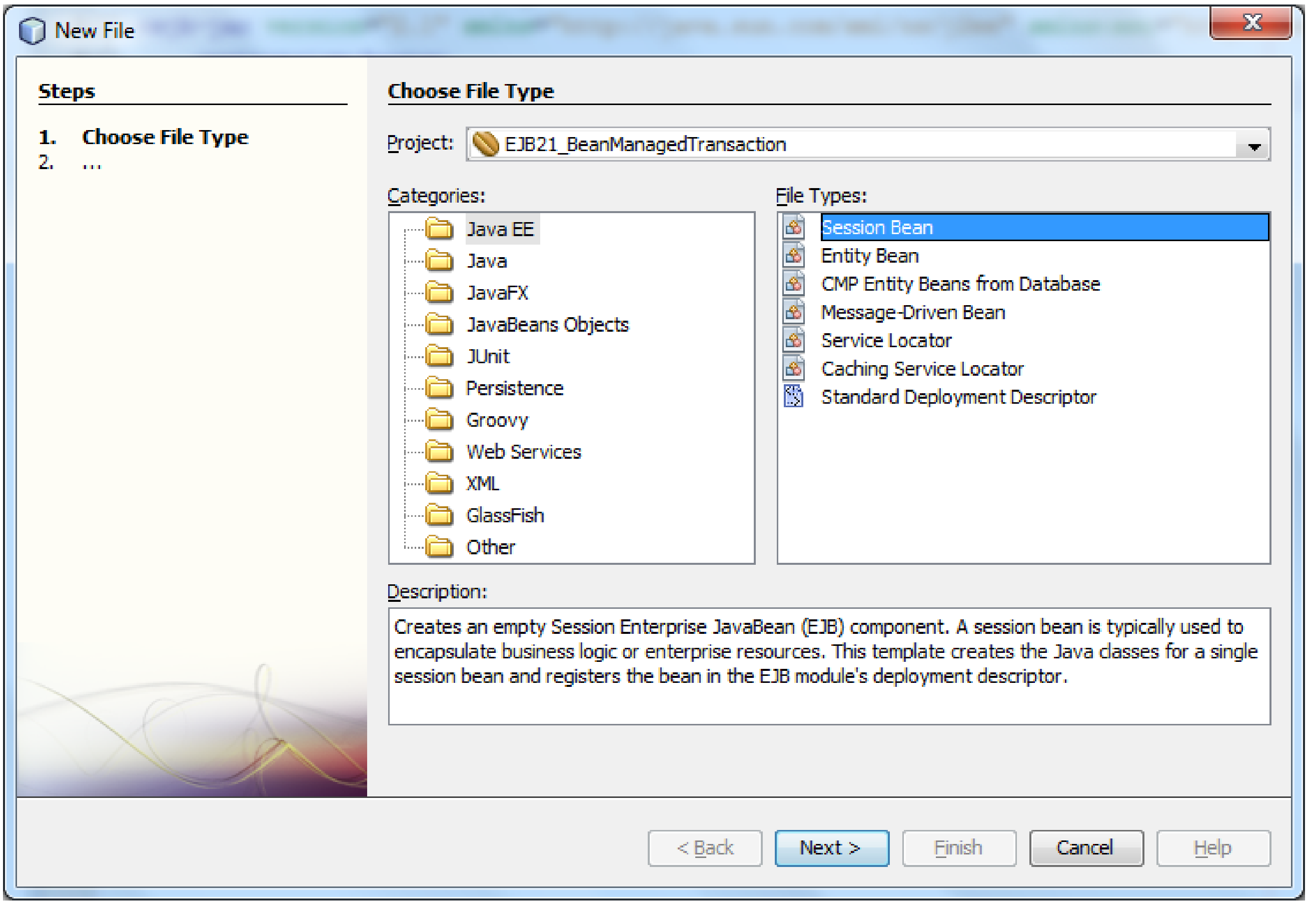
Bean Managed Transaction

Tạo Java EE Project có tên EJB21_BeanManagedTransaction.



New File

Steps

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

Choose File Type

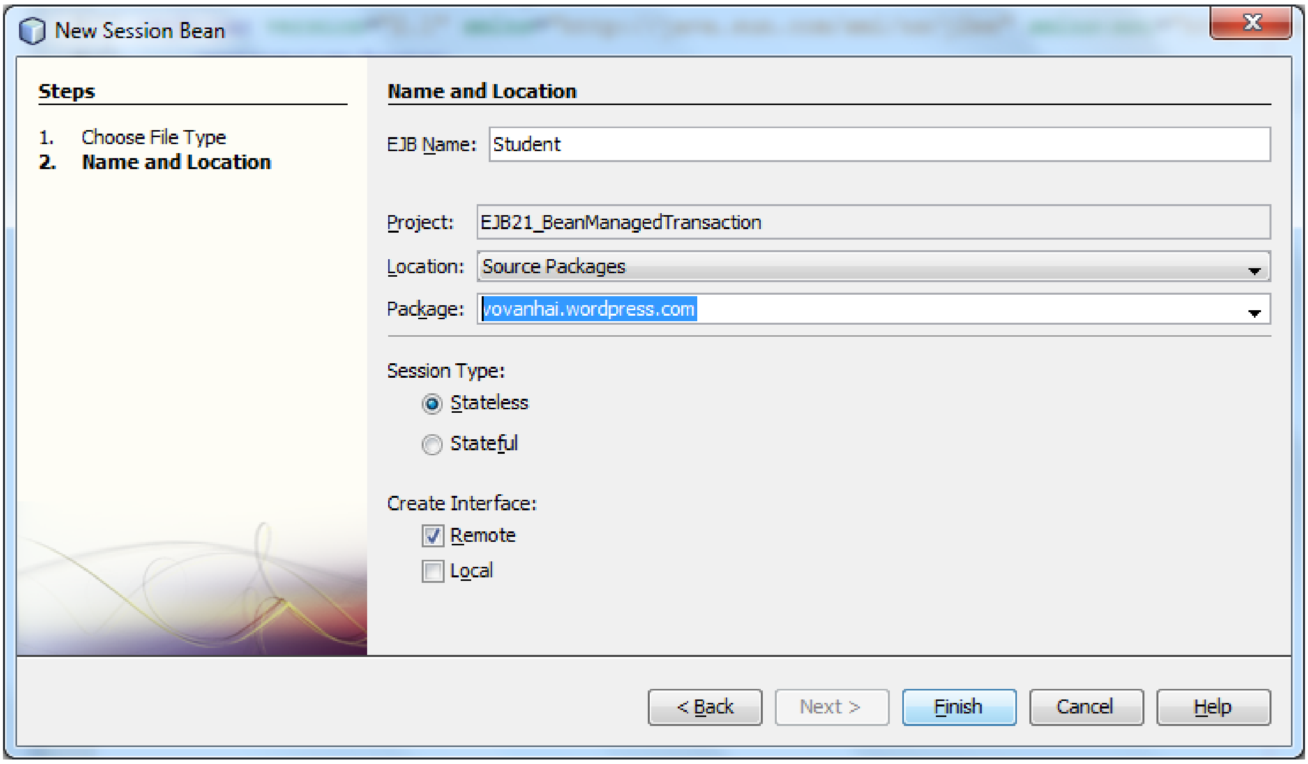
Project: EJB21_BeanManagedTransaction

Categories: Java EE, Java, JavaFX, JavaBeans Objects, JUnit, Persistence, Groovy, Web Services, XML, GlassFish, Other

File Types: Session Bean (selected), 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 the bean in the EJB module's deployment descriptor.



New Session Bean

Steps

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

Name and Location

EJB Name: Student

Project: EJB21_BeanManagedTransaction

Location: Source Packages

Package: vovanhai.wordpress.com

Session Type:

● Stateless

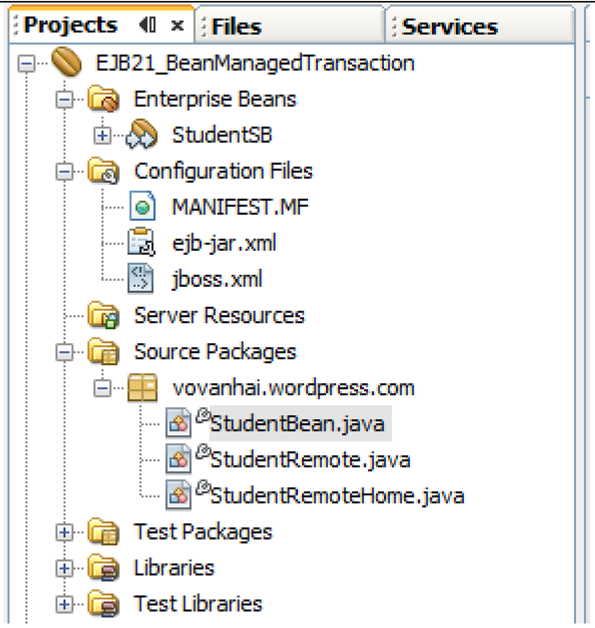
○ Stateful

Create Interface:

☑ Remote

☐ Local

Ta được:



Code cho StudentRemote.java

```
package vovanhai.wordpress.com;

import java.rmi.RemoteException;
import java.sql.SQLException;
import javax.ejb.EJBObject;

/**
 *
 * @author VoVanHai
 */
public interface StudentRemote extends EJBObject {

    public void setupDB() throws RemoteException;

    public StudentInfo getStudent(String id)
            throws RemoteException, SQLException;

    public void addStudent(StudentInfo Student)
            throws RemoteException, SQLException;

    public void updateStudent(StudentInfo Student)
            throws RemoteException, SQLException;

    public void deleteStudent(String id)
            throws RemoteException, SQLException;
}
```

Code cho StudentRemoteHome.java

```
package vovanhai.wordpress.com;

import java.rmi.RemoteException;
import javax.ejb.CreateException;
import javax.ejb.EJBHome;

/**
```

```
 *
 * @author VoVanHai
 */
public interface StudentRemoteHome extends EJBHome {

    StudentRemote create()  throws CreateException, RemoteException;

}
```

Code cho StudentBean.java

```
                    package vovanhai.wordpress.com;

import java.rmi.RemoteException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.ejb.CreateException;
import javax.ejb.SessionBean;
import javax.ejb.SessionContext;
import javax.naming.InitialContext;
import javax.sql.DataSource;
import javax.transaction.SystemException;
import javax.transaction.UserTransaction;

/**
 *
 * @author VoVanHai
 */
public class StudentBean implements SessionBean {

    SessionContext ctx;
    DataSource ds = null;
    String id = null;
    String fname = null;
    String lname = null;
    String address = null;

    public void setupDB()
            throws RemoteException {
        System.out.println("Settingup Database...");
        try {
            InitialContext ctx = new InitialContext();
            //tim trong ejb-jar.xml tag <resource-env-ref-name>, sau do lookup tiep trong jboss.xml de lay duoc <jndi-name>
            ds = (DataSource) ctx.lookup("java:comp/env/jdbc/studentsDB");
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    public StudentInfo getStudent(String id)
            throws RemoteException, SQLException {
        try {
            String sql = "SELECT * " +
                    "FROM STUDENTS " +
                    "WHERE STUDENT_ID ='" + id + "'";
            Connection conn = ds.getConnection();
            System.out.println("Conntected to DB...");
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql);
            System.out.println("Getting Result...");
            while (rs.next()) {
                id = rs.getString(1);
                fname = rs.getString(2);
                lname = rs.getString(3);
                address = rs.getString(4);
            }
            conn.close();
            System.out.println("Returning Student Info...");
            return (new StudentInfo(id, fname, lname, address));

        } catch (java.sql.SQLException e) {
            throw new RemoteException("SQL failed: " + e.getMessage());
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        return null;
    }

    public void addStudent(StudentInfo st)
            throws RemoteException, SQLException {
        Connection conn = null;
        UserTransaction ut = ctx.getUserTransaction();
        try {
            System.out.println("addStudent: Starting new transaction...");
            ut.begin();//bat dau giao dich

            String sql = "INSERT INTO STUDENTS VALUES('" + st.id +
                    "','" + st.fname +
                    "', '" + st.lname +
                    "', '" + st.address + "')";
            //System.out.println("xxxxxxx toi day roi");
            conn = ds.getConnection();

            Statement stmt = conn.createStatement();

            stmt.executeUpdate(sql);//neu cau nay khong loi thi->

            ut.commit();//->khop giao dich
            stmt.close();
            conn.close();
            System.out.println("addStudent: transaction committed...");
        } catch (java.sql.SQLException e) {
            try {
                System.out.println("addStudent: transaction rolled back...");
                ut.rollback();//khoi phuc trang thai truoc khi giao dich
                conn.close();
            } catch (SystemException ex) {
                throw new RemoteException("Rollback failed: " + ex.getMessage());
            }

        } catch (Exception e) {
            try {
```

```
            ut.rollback();
        } catch (SystemException ex) {
            throw new RemoteException("Rollback failed: " + ex.getMessage());
        }
    }
}

public void updateStudent(StudentInfo st)
        throws RemoteException, SQLException {
    Connection conn = null;
    UserTransaction ut = ctx.getUserTransaction();
    try {
        System.out.println("updateStudent: Starting new transaction...");
        ut.begin();

        String sql = "UPDATE STUDENTS SET FIRST_NAME  ='" + st.fname +
                "', LAST_NAME ='" + st.lname +
                "', ADDRESS ='" + st.address +
                "' WHERE STUDENT_ID='" + st.id + "'";
        conn = ds.getConnection();
        Statement stmt = conn.createStatement();
        stmt.executeUpdate(sql);
        ut.commit();
        stmt.close();
        conn.close();
        System.out.println("updateStudent: transaction committed...");
    } catch (java.sql.SQLException e) {
        try {
            System.out.println("updateStudent: transaction rolled back...");
            e.printStackTrace();
            ut.rollback();
            conn.close();
        } catch (SystemException ex) {
            throw new RemoteException("Rollback failed: " + ex.getMessage());
        }

    } catch (Exception e) {
        try {
            ut.rollback();
        } catch (SystemException ex) {
            throw new RemoteException("Rollback failed: " + ex.getMessage());
        }
    }
}

public void deleteStudent(String id)
        throws RemoteException, SQLException {
    Connection conn = null;
    UserTransaction ut = ctx.getUserTransaction();
    try {
        System.out.println("deleteStudent: Starting new transaction...");
        ut.begin();
        String sql = "DELETE FROM STUDENTS WHERE STUDENT_ID='" + id + "'";
        conn = ds.getConnection();
        Statement stmt = conn.createStatement();
        int x = stmt.executeUpdate(sql);
        if (x > 0) {
                System.out.println("????deleteStudent: transaction committed...");
            } else {
                System.out.println("________deleteStudent: student not exist...");
            }
            ut.commit();
            stmt.close();
            conn.close();

        } catch (java.sql.SQLException e) {
            try {
                System.out.println("deleteStudent: transaction rolled back...");
                ut.rollback();
                conn.close();
            } catch (SystemException ex) {
                throw new RemoteException("Rollback failed: " + ex.getMessage());
            }
        } catch (Exception e) {
            try {
                ut.rollback();
            } catch (SystemException ex) {
                throw new RemoteException("Rollback failed: " + ex.getMessage());
            }
        }
    }

    public void ejbCreate() throws RemoteException, CreateException {
    }

    public void ejbRemove() {
    }

    public void setSessionContext(SessionContext ctx) {
        this.ctx = ctx;
    }

    public void ejbActivate() {
    }

    public void ejbPassivate() {
    }
}
```

File StudentInfo.java chứa thông tin của 1 sinh viên:

```
package vovanhai.wordpress.com;

public class StudentInfo implements java.io.Serializable {
  public String id = null;
  public String fname = null;
  public String lname = null;
  public String address = null;
  public StudentInfo (String ID, String fname, String lname, String address){
    this.fname = fname;
    this.lname = lname;
    this.id = ID;
    this.address = address;
  }
}
```

Cấu hình của ejb-jar.xml như sau

```
<?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>StudentSB</display-name>
            <ejb-name>StudentBean</ejb-name>
            <home>vovanhai.wordpress.com.StudentRemoteHome</home>
            <remote>vovanhai.wordpress.com.StudentRemote</remote>
            <ejb-class>vovanhai.wordpress.com.StudentBean</ejb-class>
            <session-type>Stateless</session-type>
            <transaction-type>Bean</transaction-type>
            <resource-env-ref>
                <resource-env-ref-name>jdbc/studentsDB</resource-env-ref-name>
                <resource-env-ref-type>javax.sql.DataSource</resource-env-ref-type>
            </resource-env-ref>
        </session>
    </enterprise-beans>
</ejb-jar>
```

Cấu hình của jboss.xml như sau

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss>
    <enterprise-beans>
        <session>
            <ejb-name>Student</ejb-name>
            <jndi-name>bmt/MyStudentBMT</jndi-name>
            <resource-env-ref>
                <resource-env-ref-name>jdbc/studentsDB</resource-env-ref-name>
                        <!--Cau hinh trong \server\default\deploy\mssql-ds.xml ten jndi nay-->
                <jndi-name>java:/Students_DSN</jndi-name>
            </resource-env-ref>
        </session>
    </enterprise-beans>
</jboss>
```

Phân đoạn mssql–ds.xml trong thư mục deploy có nội dung:

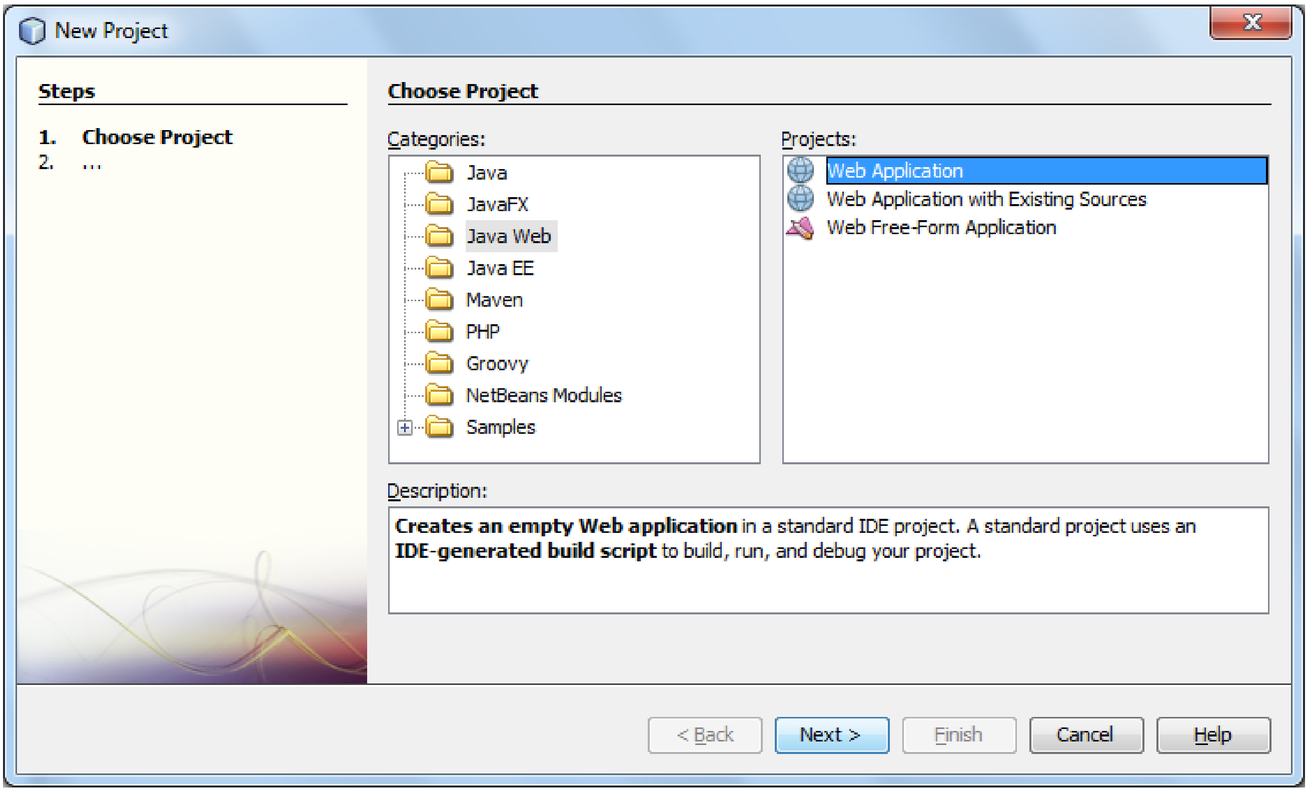
```
…
<local-tx-datasource>
    <jndi-name>Students_DSN</jndi-name>
    <connection-url>jdbc:sqlserver://localhost:1433;databaseName=StudentMS</connection-url>
    <driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
    <user-name>sa</user-name>
    <password/>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
    <idle-timeout-minutes>5</idle-timeout-minutes>
```

```
  </local-tx-datasource>
…
```

Tiền hành triển khai

Client

Tạo Web Application, đặt tên: EJB21_BeanManagedTransaction_client



New Project

Steps

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

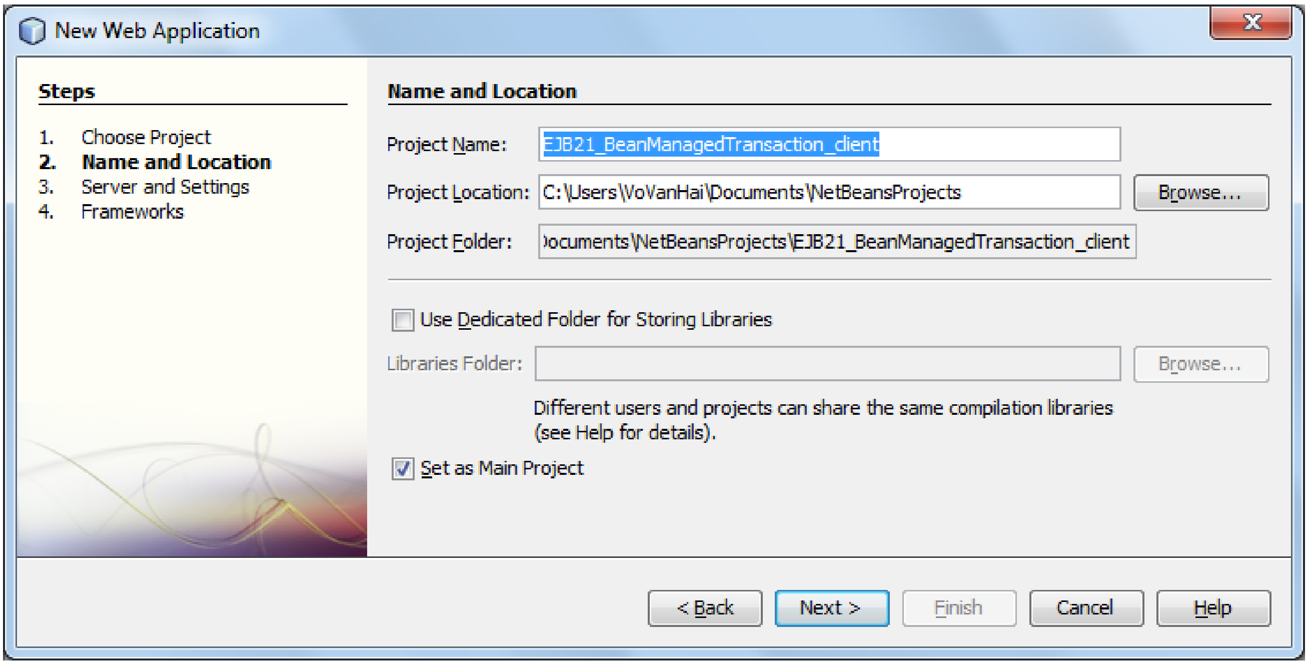
Choose Project

Categories: Java, JavaFX, Java Web, Java EE, Maven, PHP, Groovy, NetBeans Modules, Samples

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_BeanManagedTransaction_client

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

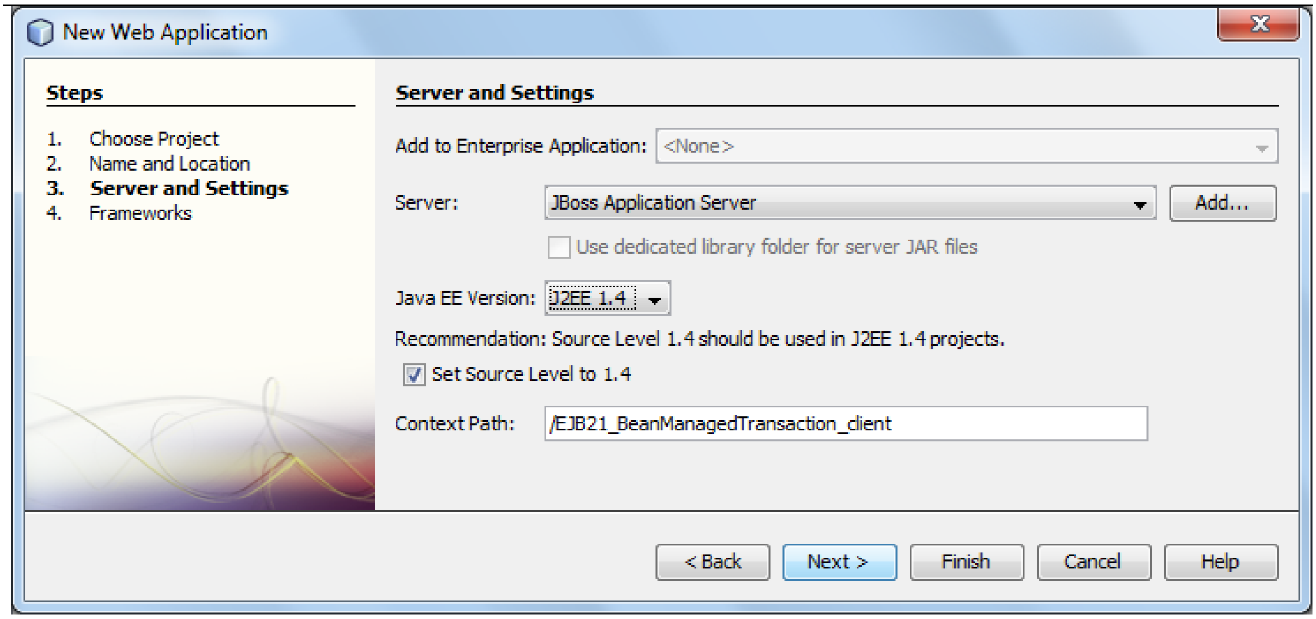
Project Folder:)ocuments\NetBeansProjects\EJB21_BeanManagedTransaction_client

□ 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

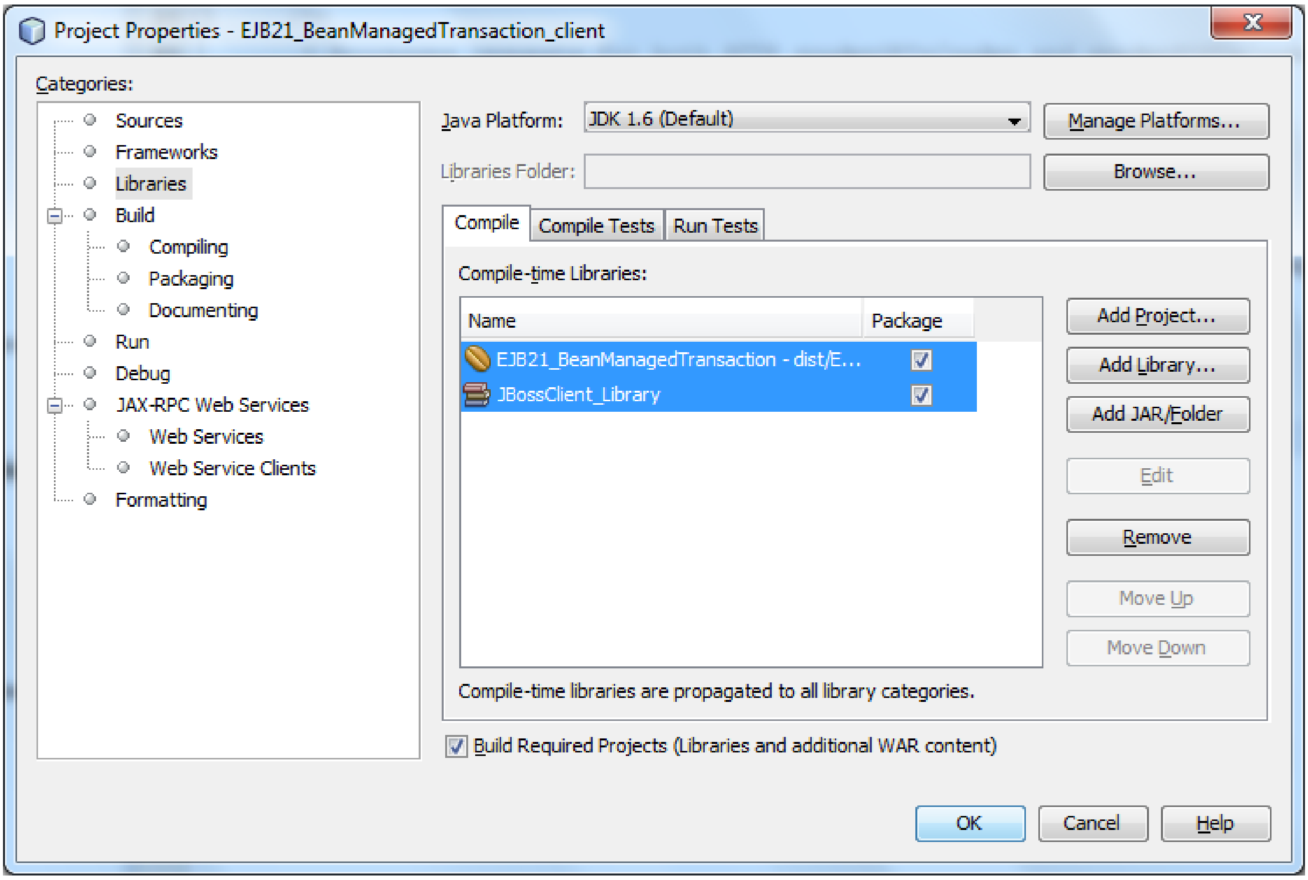
Java EE Version: J2EE 1.4

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

☑ Set Source Level to 1.4

Context Path: /EJB21_BeanManagedTransaction_client

Thêm tham chiếu đến project và libraries



Project Properties - EJB21_BeanManagedTransaction_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

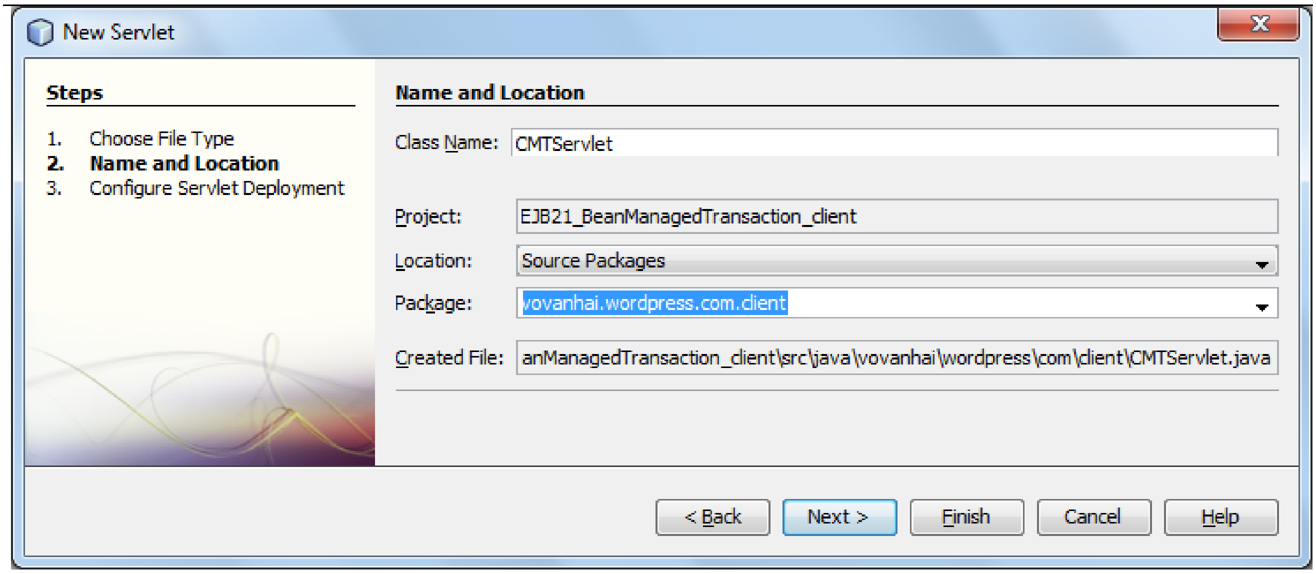
Compile-time Libraries:

Name	Package
EJB21_BeanManagedTransaction - dist/E...	☑
JBossClient_Library	☑

Compile-time libraries are propagated to all library categories.

☑ Build Required Projects (Libraries and additional WAR content)

Tạo 1 Servlet có tên CMTServlet



Class Name: CMTServlet

Project: EJB21_BeanManagedTransaction_client

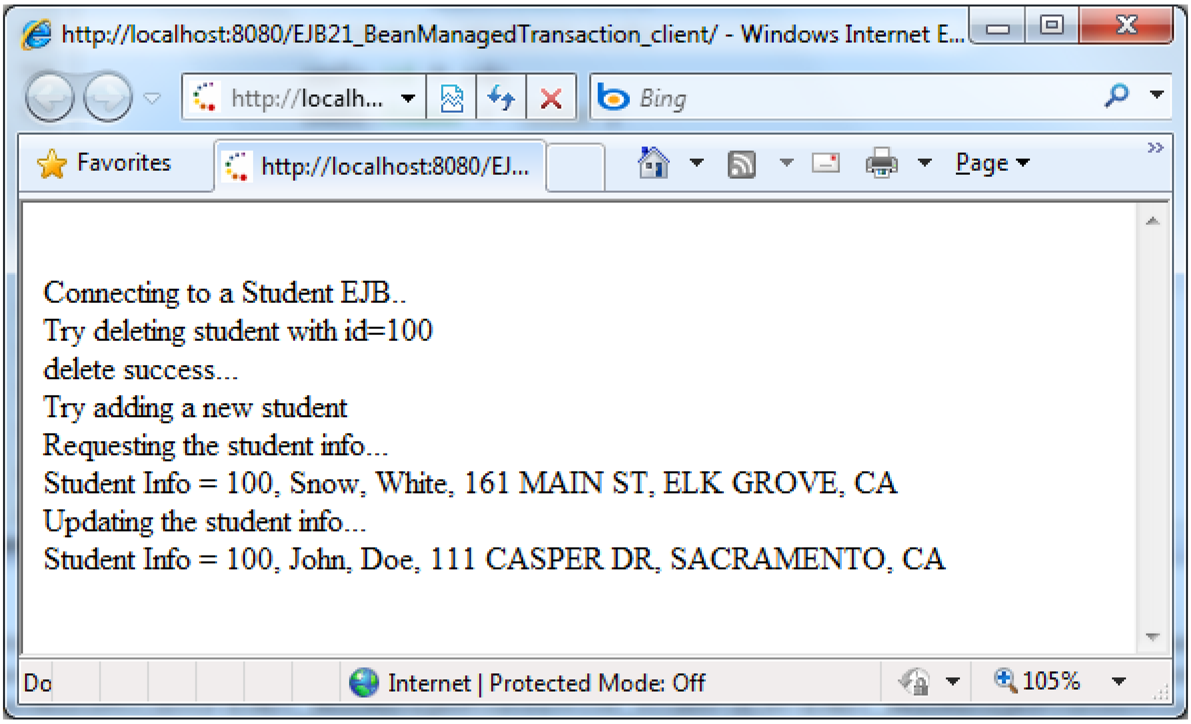
Location: Source Packages

Package: vovanhai.wordpress.com.client

Created File: anManagedTransaction_client\src\java\vovanhai\wordpress\com\client\CMTServlet.java

Code cho phương thức processRequest của servlet

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String id = "100";
        StudentInfo info = null;
        try {
            //set environment
            System.setProperty("java.naming.factory.initial",
                    "org.jnp.interfaces.NamingContextFactory");
            System.setProperty("java.naming.provider.url", "localhost:1099");
            System.setProperty("java.naming.factory.url.pkgs", "org.jboss.naming");
            // get handle to the Student object
            out.println("<br/>Connecting to a Student EJB..");
            Context ctx = new InitialContext();
            Object obj = ctx.lookup("bmt/MyStudentBMT");
            StudentRemoteHome studentHome = (StudentRemoteHome)
javax.rmi.PortableRemoteObject.narrow(obj,
                    StudentRemoteHome.class);
            StudentRemote student = (StudentRemote) studentHome.create();
            student.setupDB();
            try {
                out.println("<br/>Try deleting student with id=" + id);
                student.deleteStudent(id);
                out.println("<br/>delete success...");
            } catch (Exception e) {
                out.println("<br/>Record does not exist...");
            }
            out.println("<br/>Try adding a new student");
            info = new StudentInfo(id, "Snow", "White", "161 MAIN ST, ELK GROVE, CA");
            student.addStudent(info);
            out.println("<br/>Requesting the student info...");
            info = student.getStudent(id);
            out.println("<br/>Student Info = " +
                    info.id + ", " +
                    info.fname + ", " +
                    info.lname + ", " +
                    info.address);
            // Modify student info
            info.id = id;
            info.fname = "John";
            info.lname = "Doe";
            info.address = "111 CASPER DR, SACRAMENTO, CA";
            // Update student in the database
            out.println("<br/>Updating the student info...");
            student.updateStudent(info);
            info = student.getStudent(id);
            out.println("<br/>Student Info = " +
                    info.id + ", " +
                    info.fname + ", " +
                    info.lname + ", " +
                    info.address);
        } catch (Exception ex) {
            out.println(ex.getMessage());
        } finally {
            out.close();
        }
    }
```


http://localhost:8080/EJB21_BeanManagedTransaction_client/ - Windows Internet E...
http://localh...
Bing
Favorites
http://localhost:8080/EJ...
Page
Do
Internet | Protected Mode: Off
105%

```
Connecting to a Student EJB..
Try deleting student with id=100
delete success...
Try adding a new student
Requesting the student info...
Student Info = 100, Snow, White, 161 MAIN ST, ELK GROVE, CA
Updating the student info...
Student Info = 100, John, Doe, 111 CASPER DR, SACRAMENTO, CA
```