

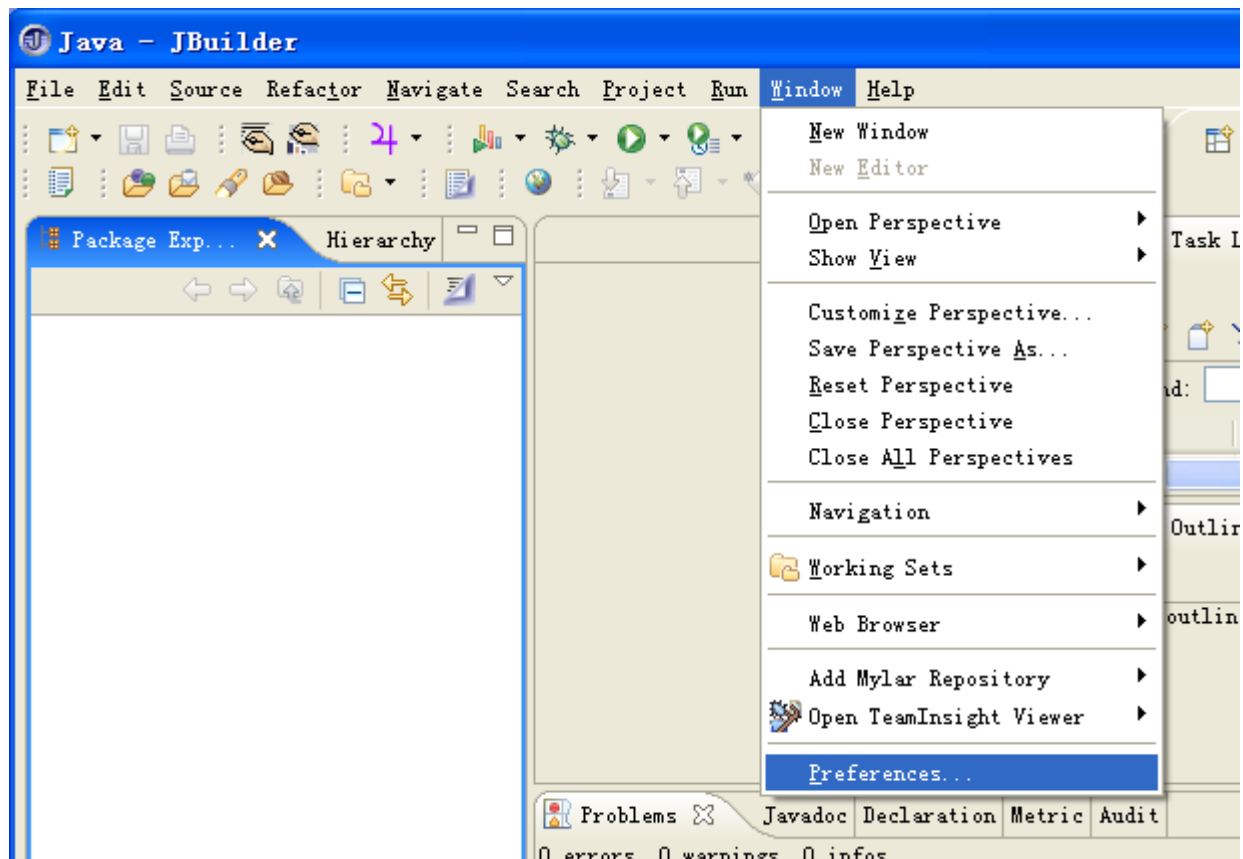
一. 准备开发环境

Jbuilder 2007, 新版 JBuilder2007 基于 Eclipse 平台, 与以往的 JBuilder 版本完全不同. 借助于 Eclipse 的开放平台以及丰富的插件, 使得 JBuilder 对当前的流行的框架支持非常好, 同时支持 EJB3.0 的可视化开发.

Jboss 4.0, Jbuilder 2007 自带, 在 JBuilder 安装目录下的 thirdparty 目录下.

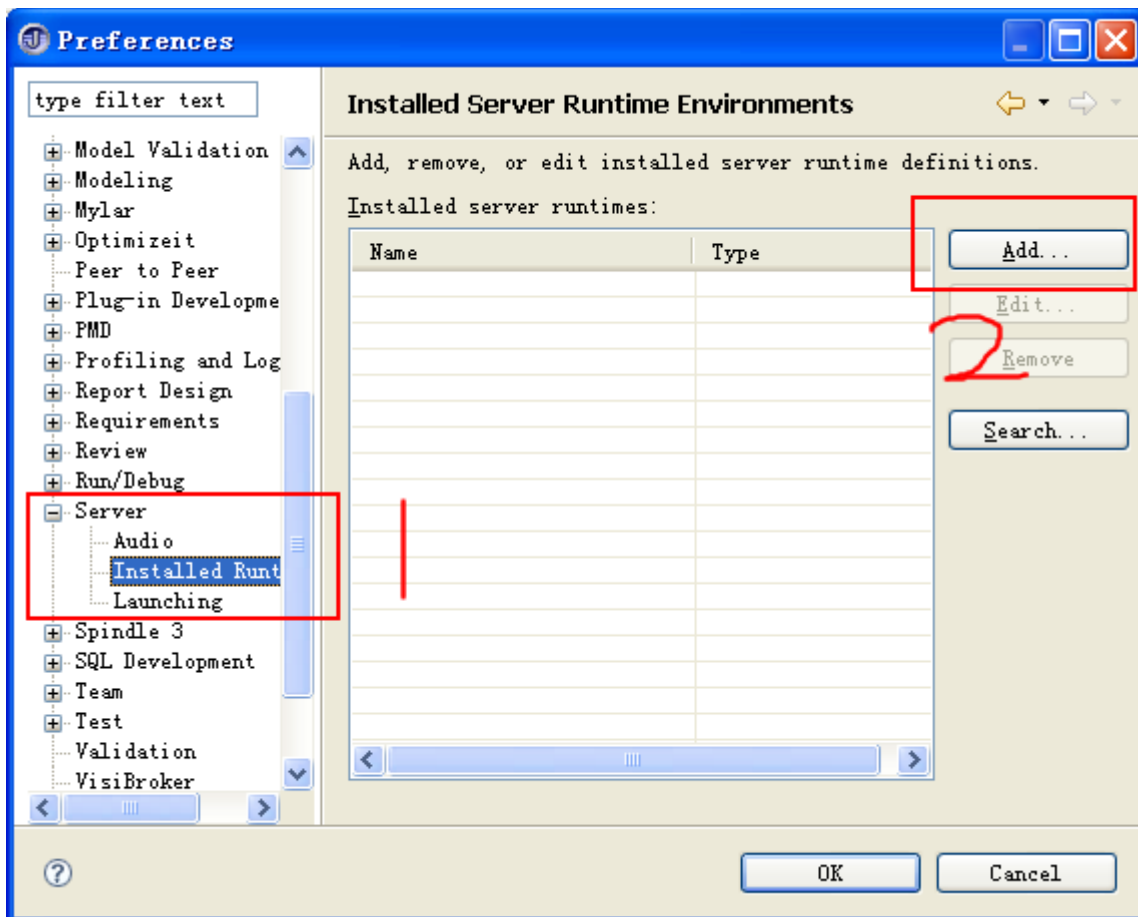
二. 在 JBuilder 2007 中配置 JBoss4.0

2.1 选择菜单 *Window / Preferences...*

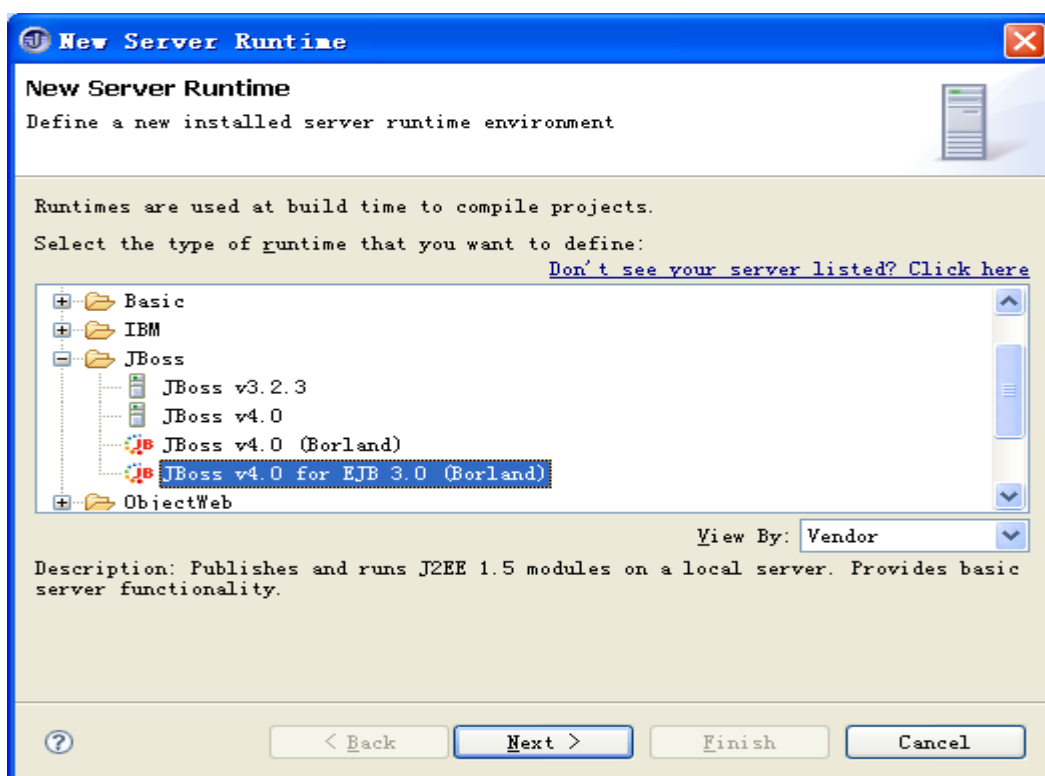


2.2 配置 Server

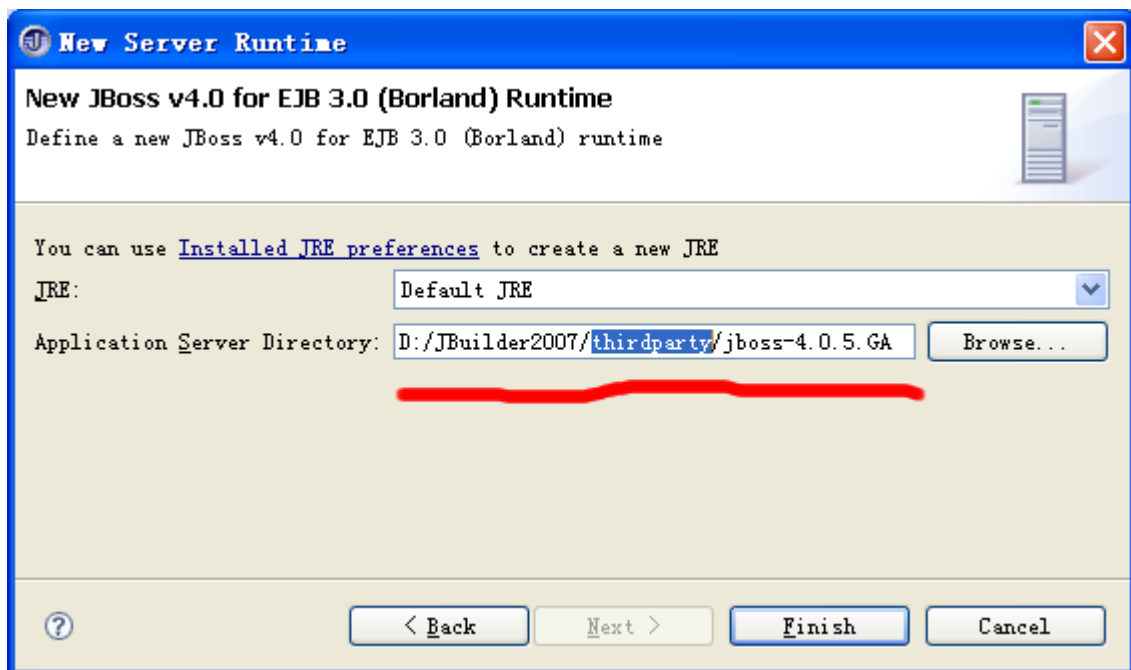
2.2.1 左边选择 servler, 单击右边的 [Add] 按钮



2.2.2 在弹出的“New Server Runtime”对话框中,选择 JBoss 4.0 for EJB3.0



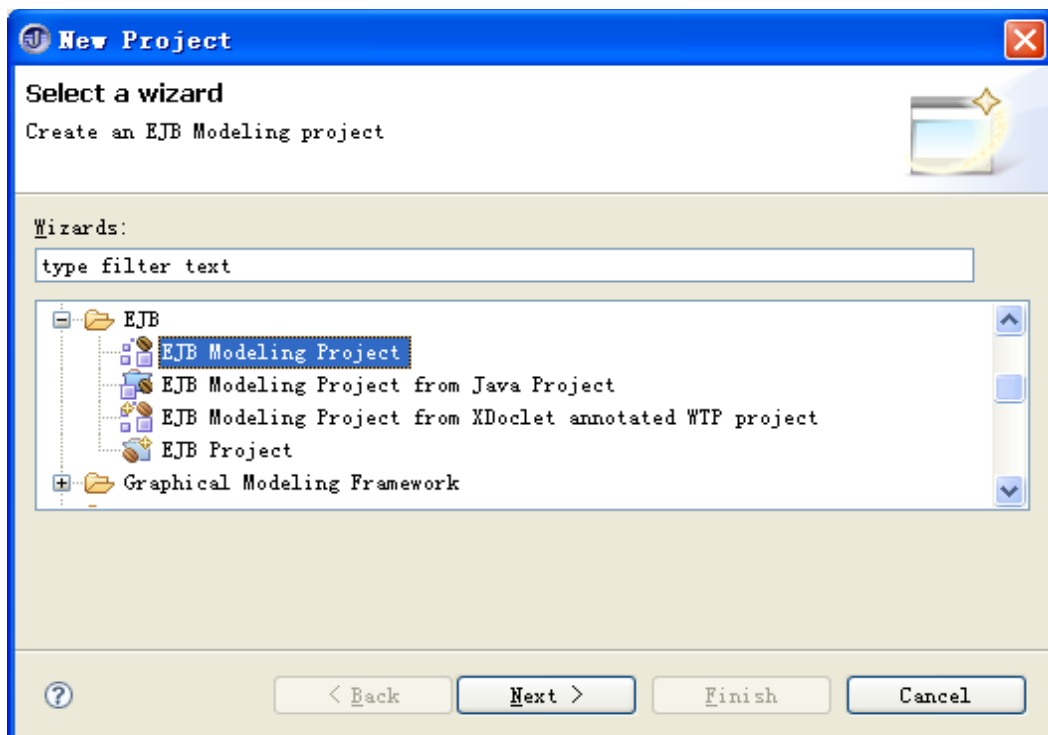
2.2.3 单击[Next]按钮, 选择 JBoss 的目录, 在 JBuilder 安装目录下的 thirdparty 目录下



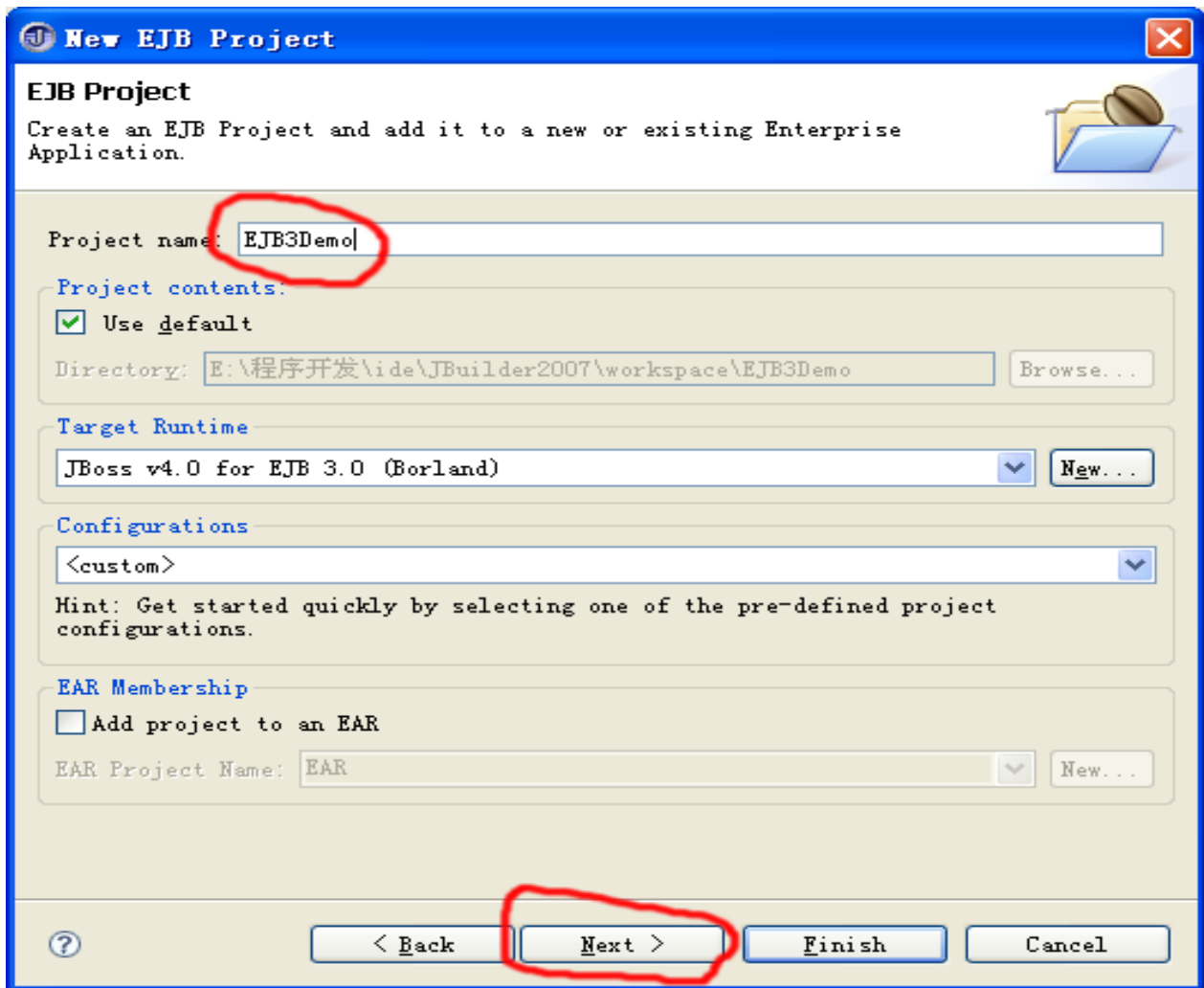
三. 建立 EJB3.0 工程

3.1 选择菜单 *File / New / Project...*

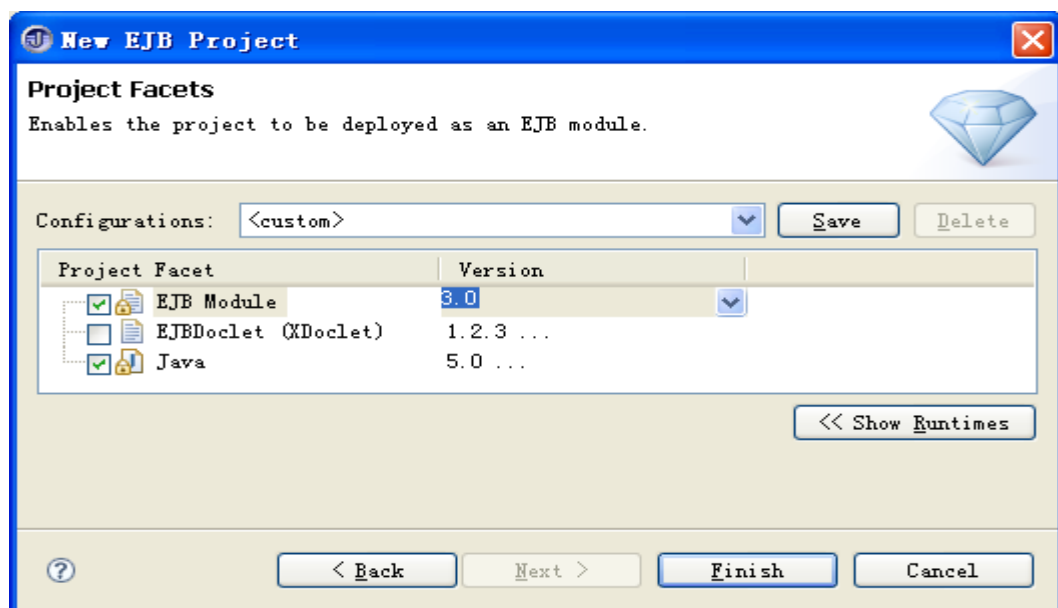
在弹出对话框中, 选择 ejb Modeling Project 项目



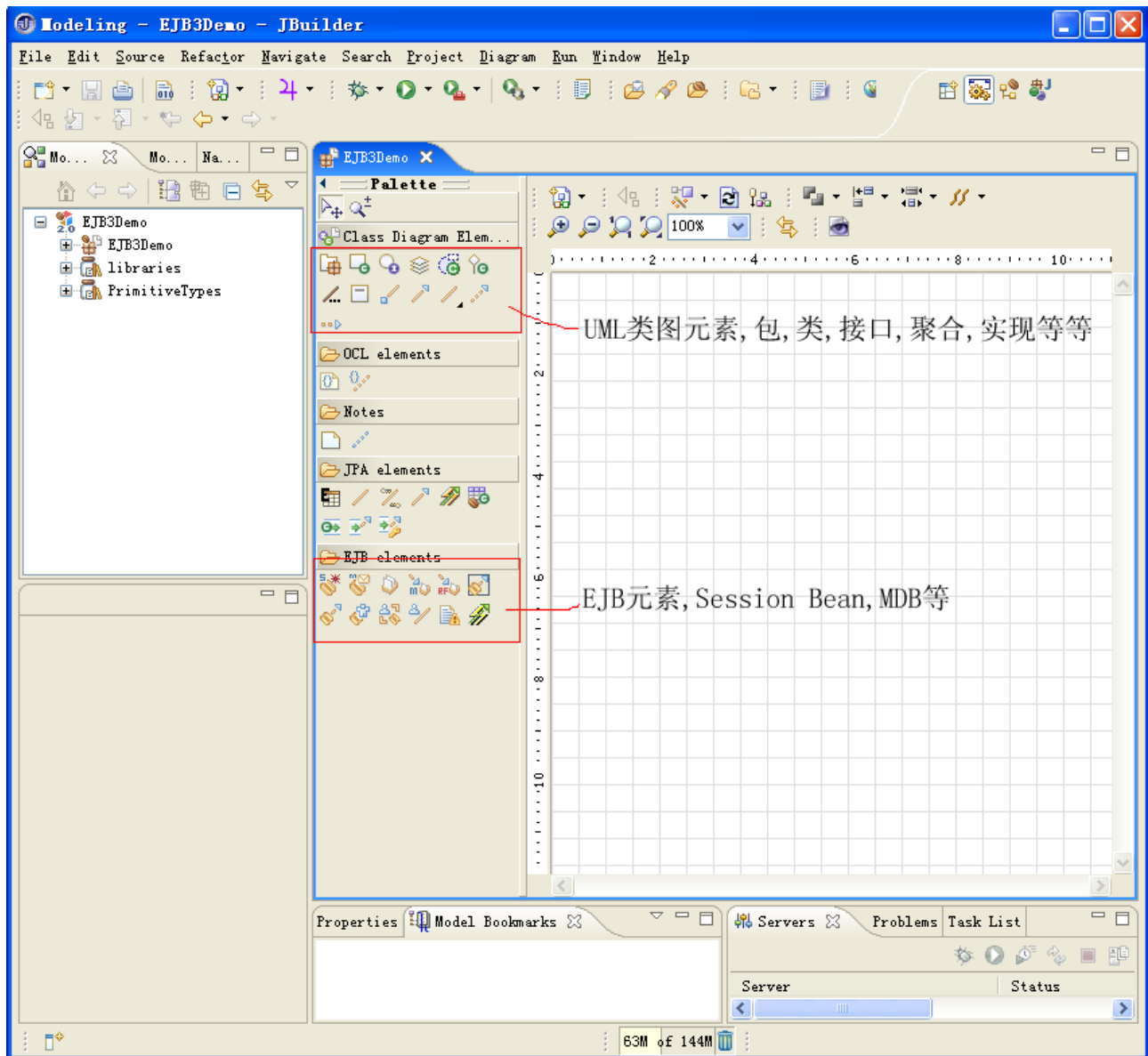
3.2 单击[Next]按钮, 输入工程名称 EJB3Demo



3.3 选择EJB版本, EJB Module 选择 3.0, java 选择 5.0



3.4 单击[Finish]按钮, 打开建模透视图



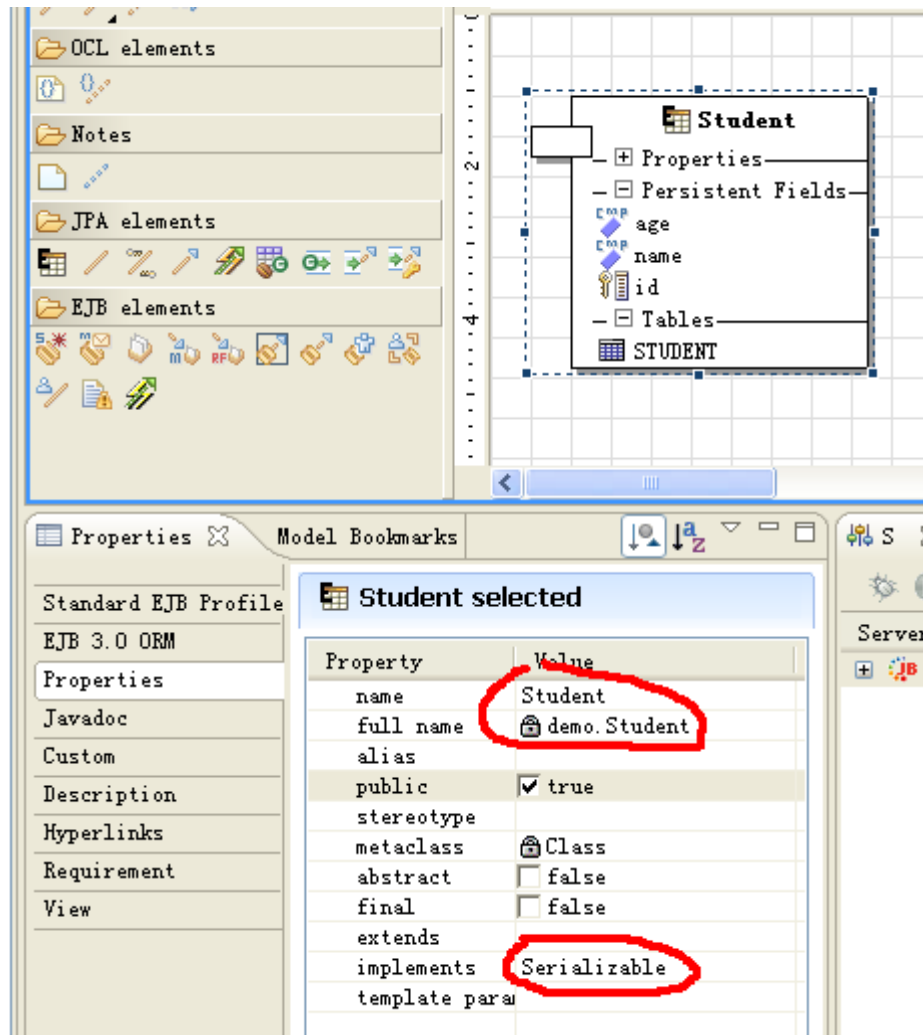
四. 开发 Entity

4.1 新建 Entity

选择 JPA 元素区的 Entity 图标, 然后在右边画图区单击鼠标,

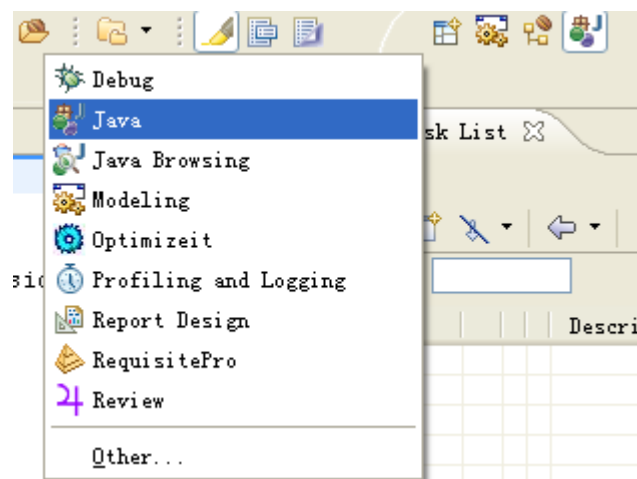


修改 Entity 的属性



4.2 新增属性

其实 ejb3.0 的 Entity 只是普通 java 类, 所以我们手工添加属性即可
打开 Java 透视图



打开 Student.java, 添加属性 age, name

Student.java 内容如下

```
package demo;

import java.io.Serializable;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity(name = "Student")
@Table(name = "STUDENT") //Entity对应的数据库表名
public class Student implements Serializable {
    private int age;

    private Long id;

    private String name;

    private static final long serialVersionUID = 1L;

    @GeneratedValue(strategy = GenerationType.AUTO) //ID生成方式
    @Id
    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

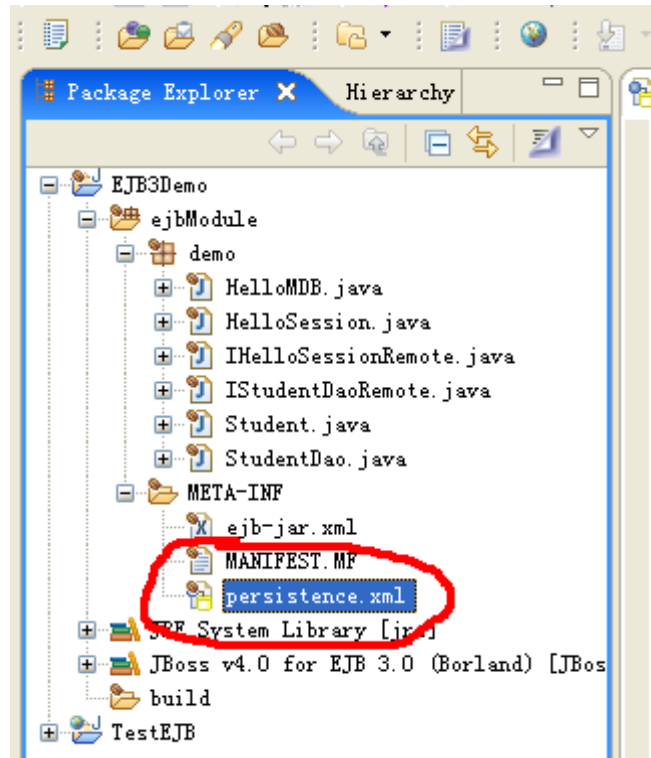
    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}
```

4.3 编辑 persistence.xml 文件

该文件在项目目录, META-INF 下



persistence.xml 内容如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0" xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence persistence_1_0.xsd">
<persistence-unit name="default">
    <jta-data-source>java:/OracleDS</jta-data-source>
    <properties>
        <property name="hibernate.hbm2ddl.auto" value="create"/>
        <property name="hibernate.show_sql" value="true"/>
    </properties>
</persistence-unit>
</persistence>
```

java:/OracleDS, 是JBoss配置的数据源名称, 下面章节会介绍如何配置

下面的两个property 属性设置, 和hibernate是相同的, 第一个表示自动创建表, 因此, 要注意, 不要自己去数据库创建表 Student!!, 第二个属性, 表示显示Hibernate生成的SQL语句.

4.3 配置 JBoss 数据源

从 D:\JBuilder2007\thirdparty\jboss-4.0.5.GA\docs\examples\jca\目录下, 拷贝 oracle-ds.xml 到 D:\JBuilder2007\thirdparty\jboss-4.0.5.GA\server\default\deploy\

目录下, 编辑 oracle-ds.xml, 修改数据库 ip 地址, 用户名, 密码等. 注意, 这里不是一定要 oracle-ds.xml, 如果你使用其它数据库, 请拷贝相应的-ds.xml 文件即可.

我这里修改好的 oracle-ds.xml 内容如下: (我把<!-- -->之间的注释掉的内容删除了)

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>OracleDS</jndi-name>
    <connection-url>jdbc:oracle:thin:@10.6.30.17:1521:ltaisln</connection-url>
    <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
    <user-name>ctaisd</user-name>
    <password>oracle</password>
    <exception-sorter-class-name>
      org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter
    </exception-sorter-class-name>
    <metadata>
      <type-mapping>Oracle9i</type-mapping>
    </metadata>
  </local-tx-datasource>
</datasources>
```

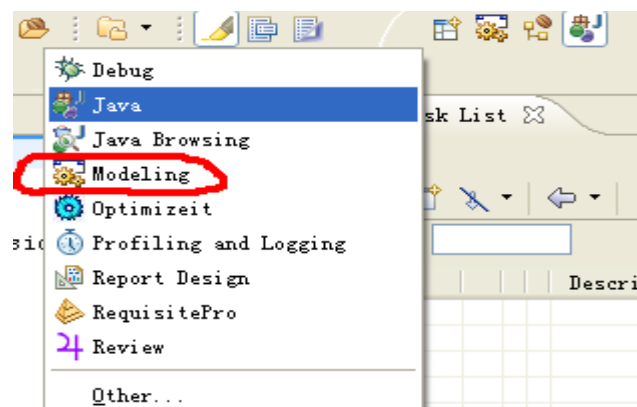
然后, 拷贝数据库的 jdbc 驱动到目录 D:\JBuilder2007\thirdparty\jboss-4.0.5.GA\server\default\lib 下. 这样, 数据源就配置好了.

五. 开发 Session Bean 作为 Dao 类

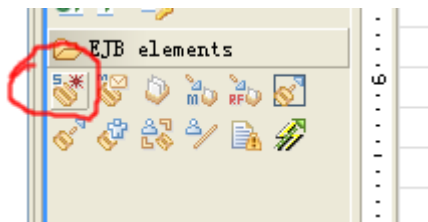
细心一点, 就可以注意到. 我们刚才的 Entity, 并不能 save, 也不能 query..., 因此, 我们还需要写一个 Session Bean 来实现 save, remove, query 等操作.

5.1 新建 Session Bean

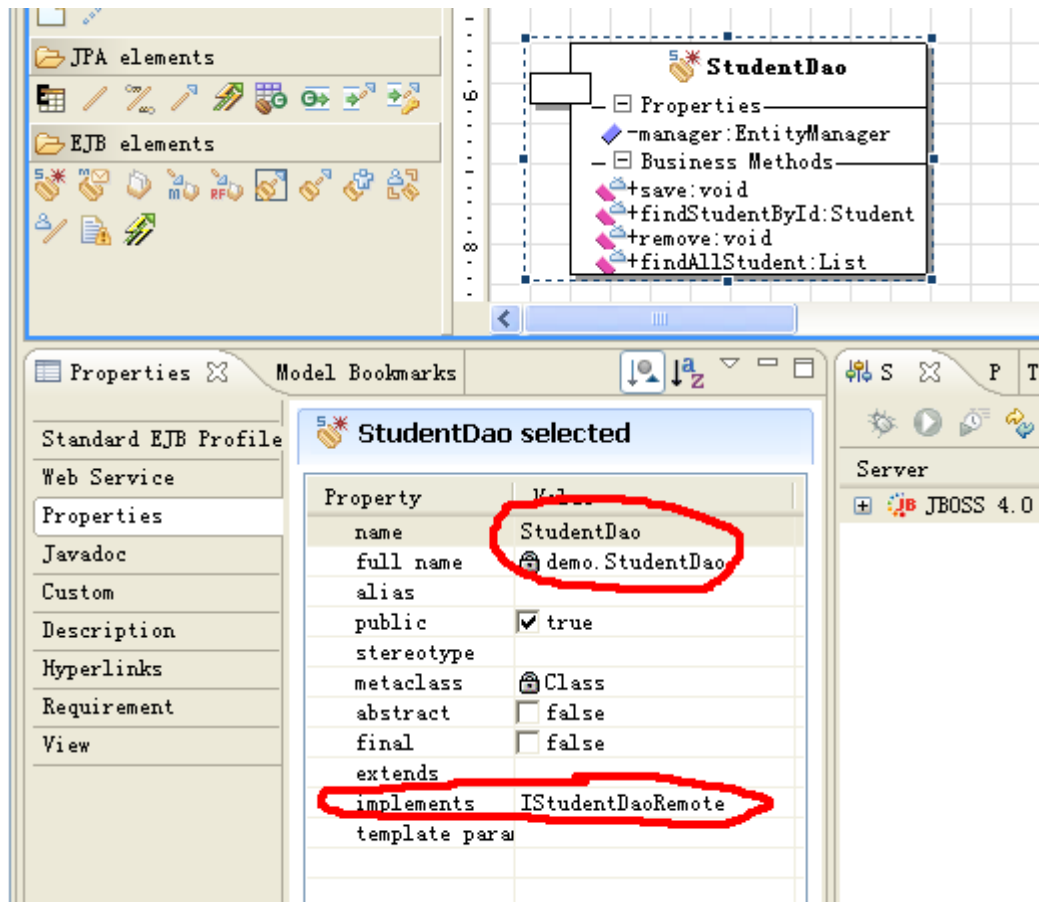
回到建模视图, 双击左边工程的包名 demo, 即可打开建模区



选择 Session Bean 图标, 然后在右边画图区单击鼠标,

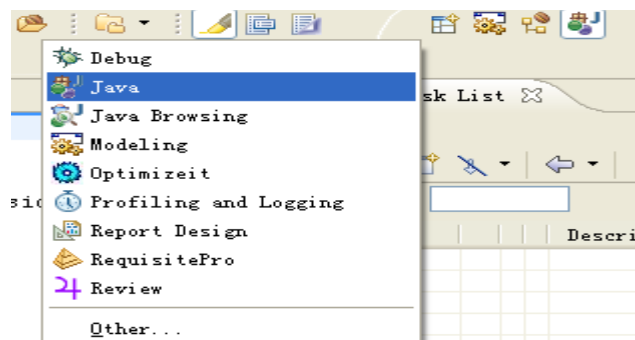


修改 Session Bean 的属性



5.2 新增业务方法

其实 ejb3.0 的 session bean 只是普通 java 类, 所以我们手工编写业务方法即可
打开 Java 透视图



打开 IStudentDaoRemote.java, 添加方法

IStudentDaoRemote.java 内容如下

```
package demo;

import java.util.List;

@javax.ejb.Remote
public interface IStudentDaoRemote {

    public void save(Student student);

    public Student findStudentById(long id);

    public void remove(Student student);

    public List findAllStudent();

}
```

打开 StudentDao.java, 实现方法

StudentDao.java 的内容如下

```
package demo;

import java.util.List;

import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

@Stateless(name = "StudentDao")
public class StudentDao implements IStudentDaoRemote {

    @PersistenceContext //这里应用EJB3的IOC, 得到EntityManager
    private EntityManager manager; //相当于Hibernate的Session

    public void save(Student student) {
        manager.persist(student);
    }

    public Student findStudentById(long id) {

        Student student = manager.find(Student.class, id);
        return student;
    }

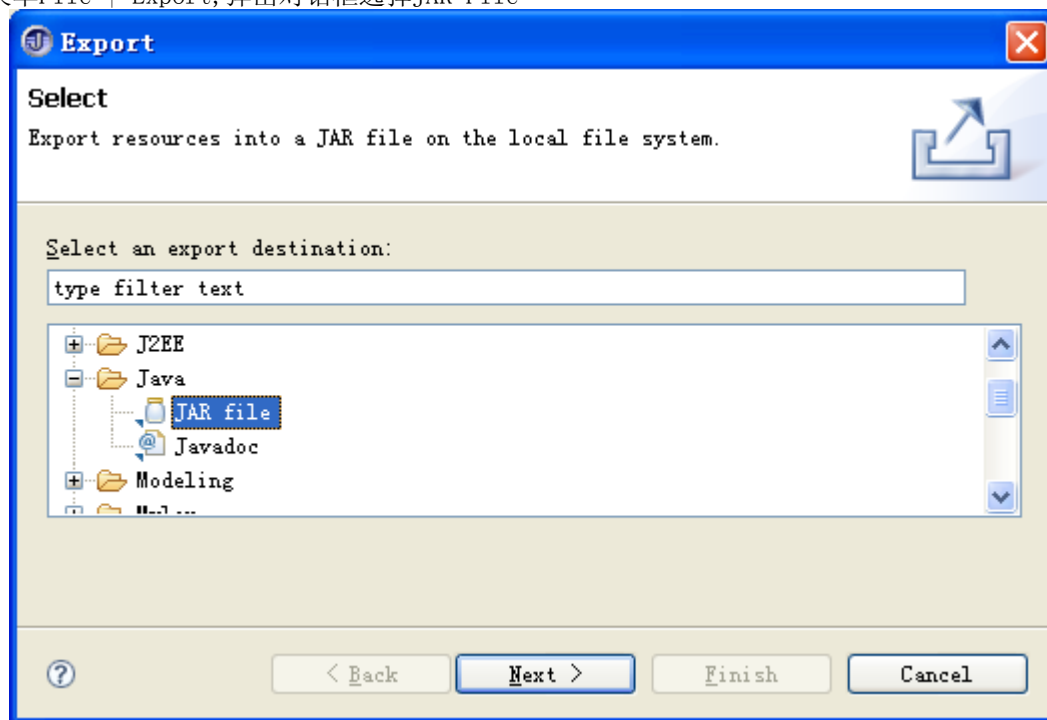
    public void remove(Student student) {
        //EJB QL, EJB 查询语言, 我还是喜欢叫它HQL
        String hql = "delete from Student s where s.id = :id";
        Query query = manager.createQuery(hql);
        query.setParameter("id", student.getId());
        query.executeUpdate();
    }

    public List findAllStudent() {
```

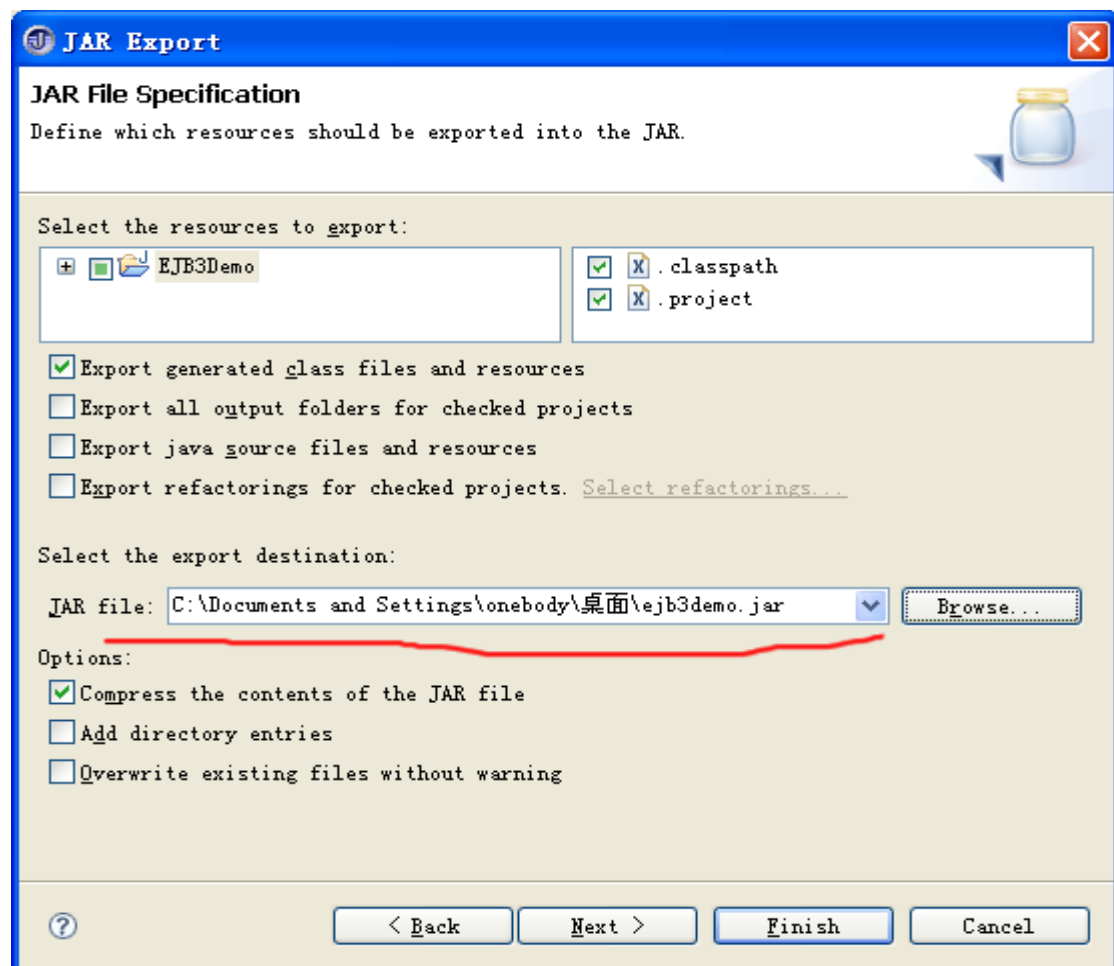
```
}  
  
    Query query = manager.createQuery("select s from Student s");  
    return query.getResultList();  
}
```

5.3 导出 jar 包

选择菜单File | Export,弹出对话框选择JAR File



单击[Next]按钮. 选择jar包导出的位置



单击[Finish]按钮, 完成导出

六. 部署及测试

6.1 部署 EJB

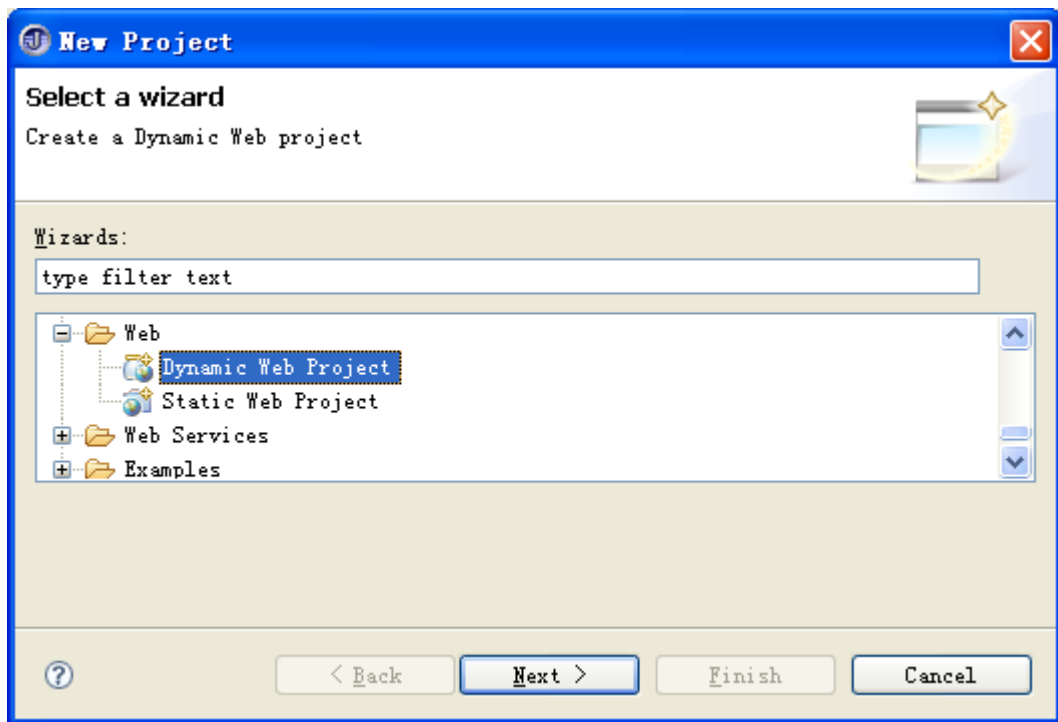
把我们刚才导出的 jar 文件. 放到 JBoss 下的 server\default\deploy 目录, 我这里是 D:\JBUILDER2007\thirdparty\jboss-4.0.5.GA\server\default\deploy 下

ok, ejb 部署好了. 下面我们写一个 servlet 测试一下

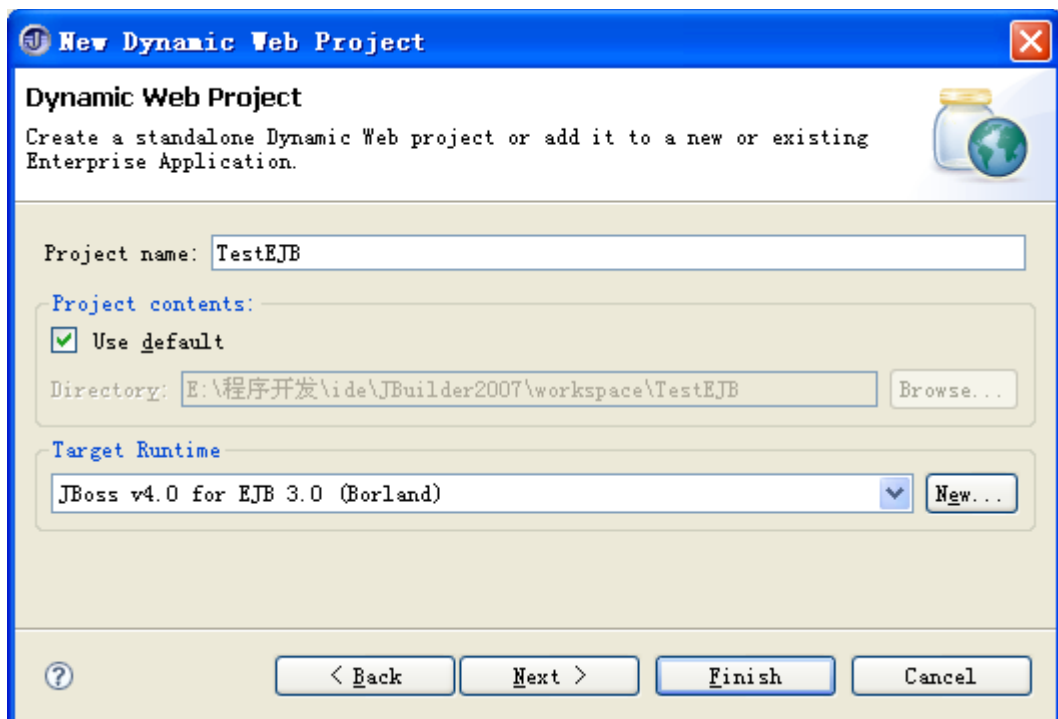
6.2 测试 EJB

5.2.1 新建动态 web 项目

选择菜单 File | new | Project , 在弹出对话框中选择 web | Dynamic Web Project



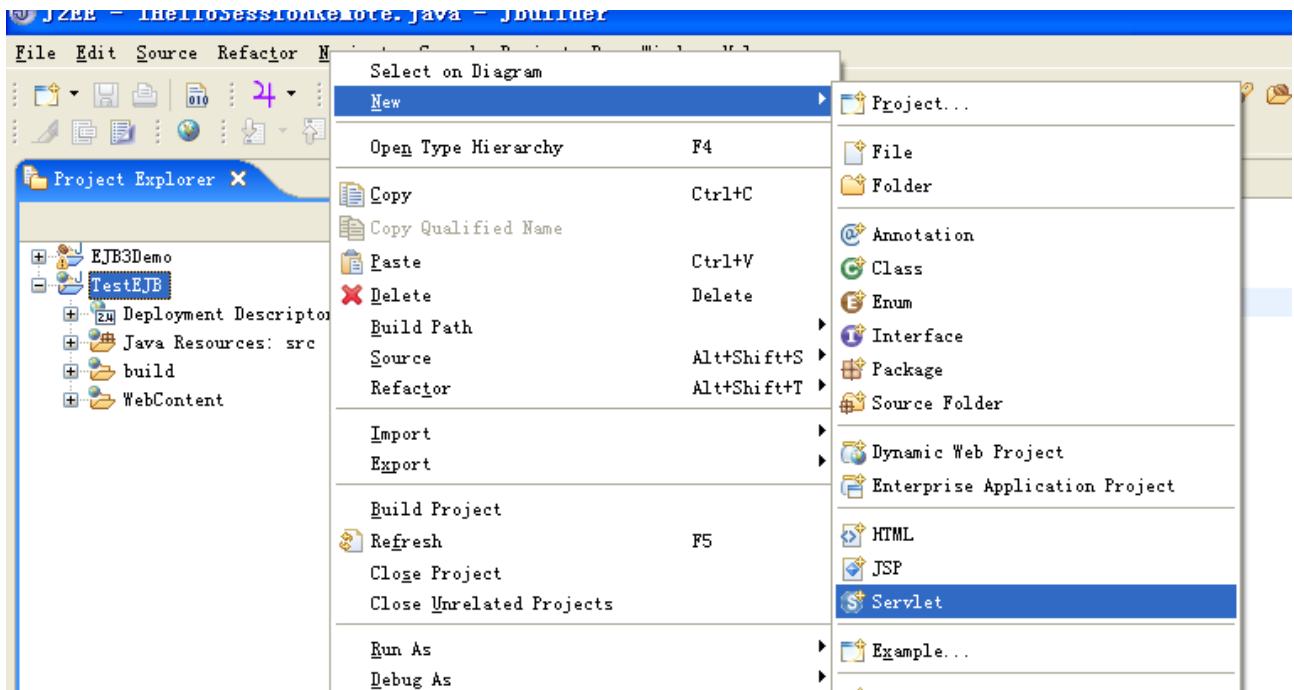
单击[Next]按钮, 填写工程名称



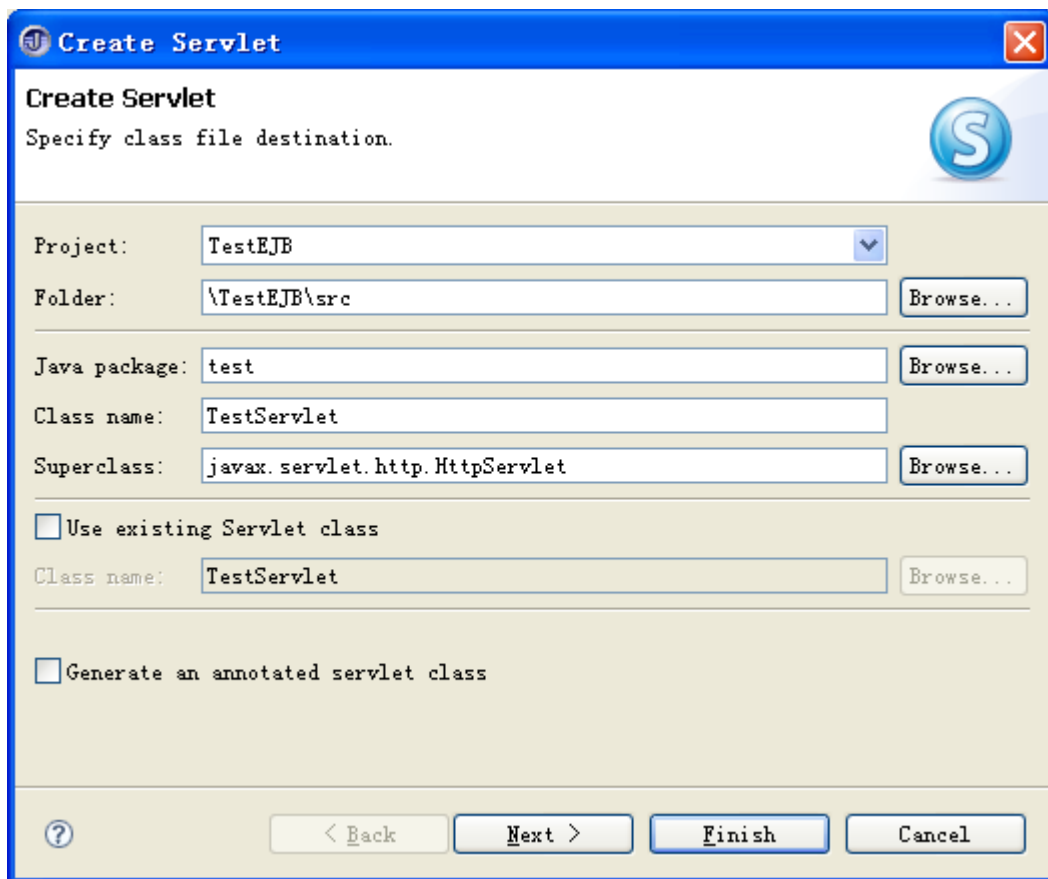
单击[Finish]按钮完成.

5.2.2 新建 Servlet

单击右键, 选择 New | Servlet



弹出对话框中, 按下图填写



单击[Finish]按钮完成

5.2.3 在 TestServlet 的 doGet 方法中, 编写测试 EJB 的代码

TestServlet.java 内容如下

```

package test;

import java.io.IOException;

import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import demo.IHelloSessionRemote;

public class TestServlet extends javax.servlet.http.HttpServlet implements
        javax.servlet.Servlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        try {
            Context ctx = new InitialContext();
            IStudentDaoRemote studentDao=(IStudentDaoRemote)ctx.lookup("StudentDao/remote");
            System.out.println("=====开始 保存Student对象=====");
            Student stus=new Student();
            stus.setName("罗代均");
            stus.setAge(24);
            studentDao.save(stus);
            System.out.println("=====结束 保存Student对象=====");
            System.out.println("=====\n");
            System.out.println("=====开始 查找Student列表=====");
            List list=studentDao.findAllStudent();
            for(Object o : list){
                Student stu=(Student)o;
                System.out.println(" name : "+stu.getName() );
                System.out.println(" age : "+stu.getAge());
            }
            System.out.println("=====结束 查找Student列表=====");

            System.out.println("=====删除开始=====");
            Student student=new Student();
            student.setId(4L);
            studentDao.remove(student);
            System.out.println("=====删除结束=====");

            System.out.println("=====开始 查找Student列表 =====");
            List list2=studentDao.findAllStudent();
            for(Object o : list2){
                Student stu=(Student)o;
                System.out.println(" name : "+stu.getName() );
                System.out.println(" age : "+stu.getAge());
            }
            System.out.println("=====结束 查找Student列表 =====");

        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}

```



```

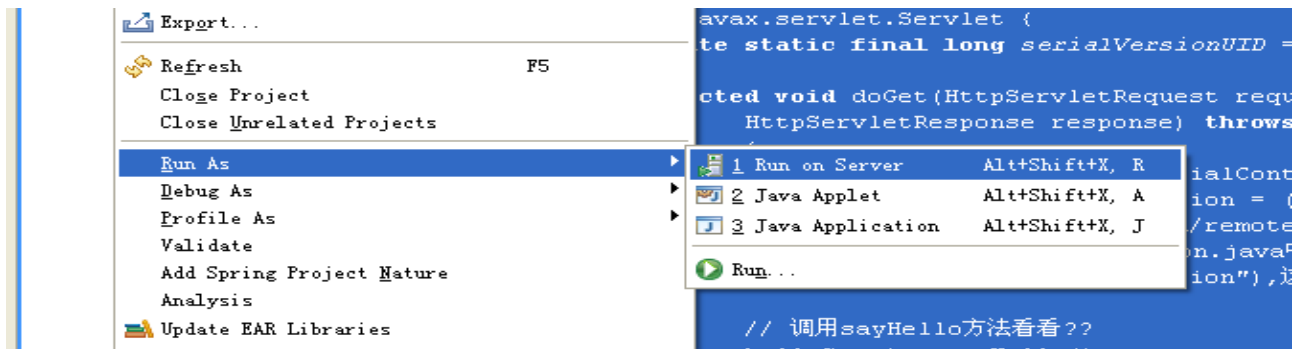
}

protected void doPost(HttpServletRequest request,
                        HttpServletResponse response)
                        throws ServletException, IOException {
}
}

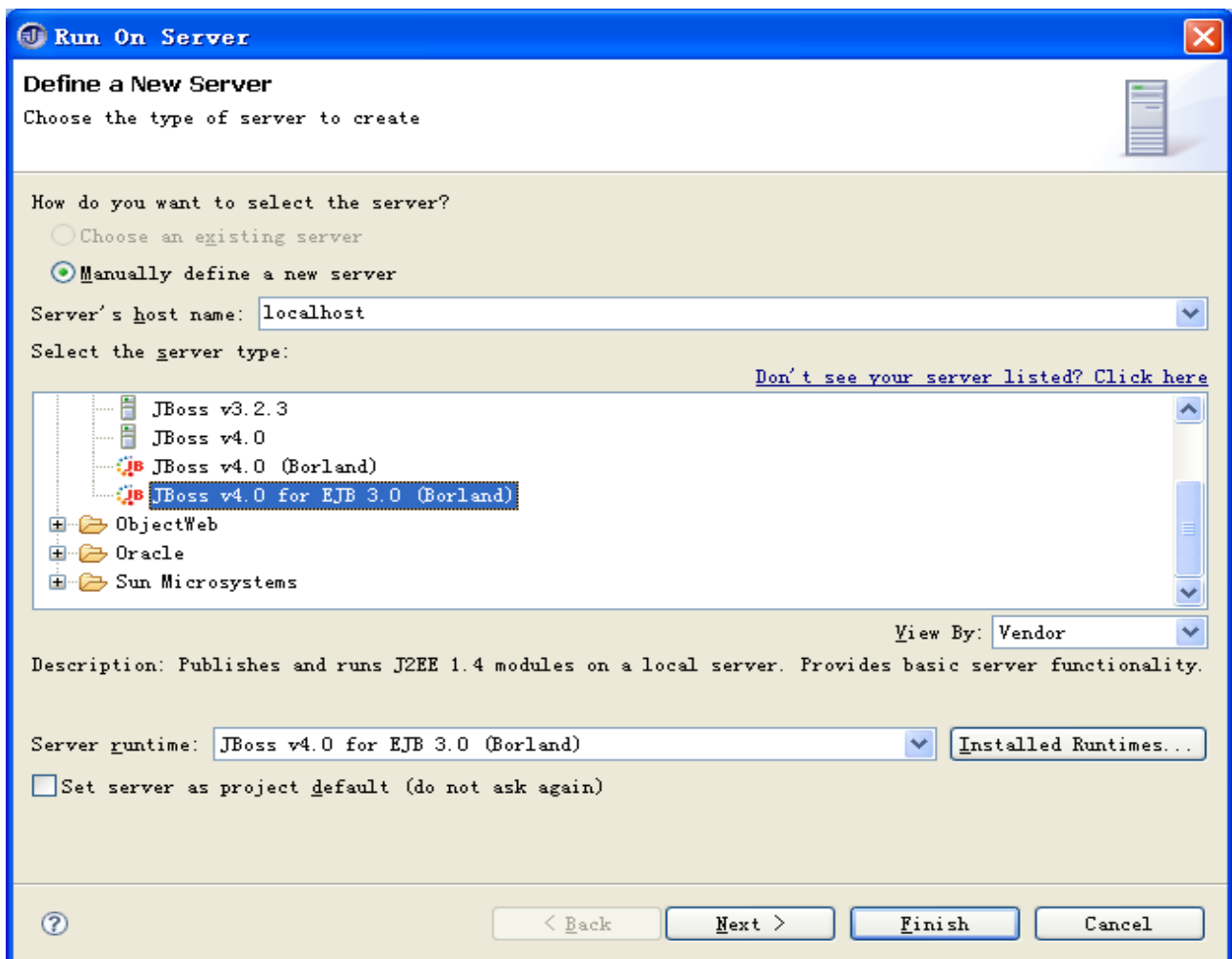
```

5.2.4 运行 TestServlet, 测试效果 ^_^

右键, 选择 Run As | Run on server



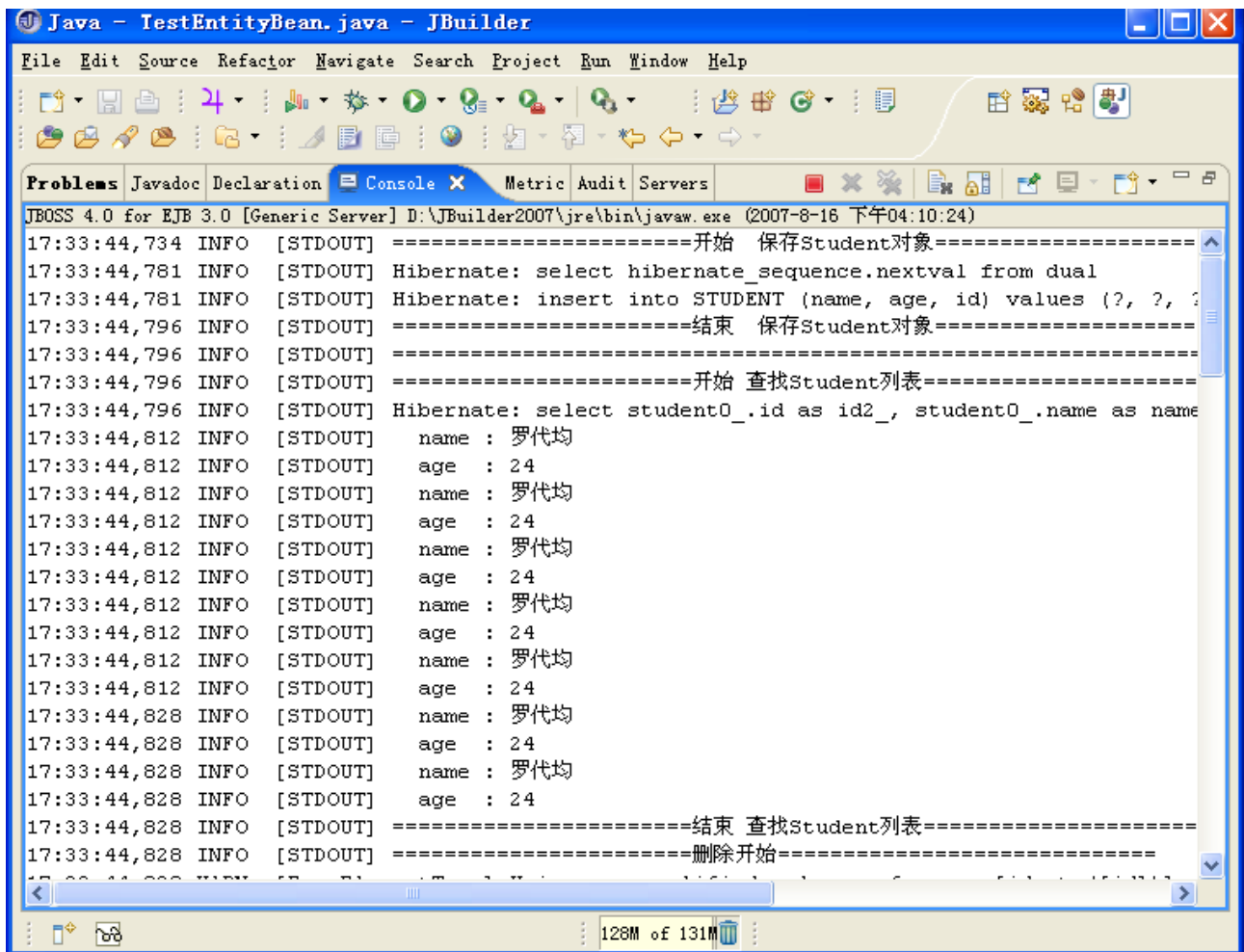
选择 JBoss 4.0 for EJB3.0



单击 [Finish] 按钮, 完成

打开 IE 浏览器, 输入 <http://localhost:8080/TestEJB/TestServlet>

嗯.. 看看 JBuilder 的控制台吧



```
Java - TestEntityBean.java - JBuilder
File Edit Source Refactor Navigate Search Project Run Window Help
JBoss 4.0 for EJB 3.0 [Generic Server] D:\JBuilder2007\jre\bin\javaw.exe (2007-8-16 下午04:10:24)
17:33:44,734 INFO [STDOUT] =====开始 保存Student对象=====
17:33:44,781 INFO [STDOUT] Hibernate: select hibernate_sequence.nextval from dual
17:33:44,781 INFO [STDOUT] Hibernate: insert into STUDENT (name, age, id) values (?, ?, ?)
17:33:44,796 INFO [STDOUT] =====结束 保存Student对象=====
17:33:44,796 INFO [STDOUT] =====开始 查找Student列表=====
17:33:44,796 INFO [STDOUT] Hibernate: select student0_.id as id2_, student0_.name as name
17:33:44,812 INFO [STDOUT] name : 罗代均
17:33:44,812 INFO [STDOUT] age : 24
17:33:44,812 INFO [STDOUT] name : 罗代均
17:33:44,812 INFO [STDOUT] age : 24
17:33:44,812 INFO [STDOUT] name : 罗代均
17:33:44,812 INFO [STDOUT] age : 24
17:33:44,812 INFO [STDOUT] name : 罗代均
17:33:44,812 INFO [STDOUT] age : 24
17:33:44,812 INFO [STDOUT] name : 罗代均
17:33:44,812 INFO [STDOUT] age : 24
17:33:44,828 INFO [STDOUT] name : 罗代均
17:33:44,828 INFO [STDOUT] age : 24
17:33:44,828 INFO [STDOUT] name : 罗代均
17:33:44,828 INFO [STDOUT] age : 24
17:33:44,828 INFO [STDOUT] =====结束 查找Student列表=====
17:33:44,828 INFO [STDOUT] =====删除开始=====
```

嗯. 结束, 打完收工!!