

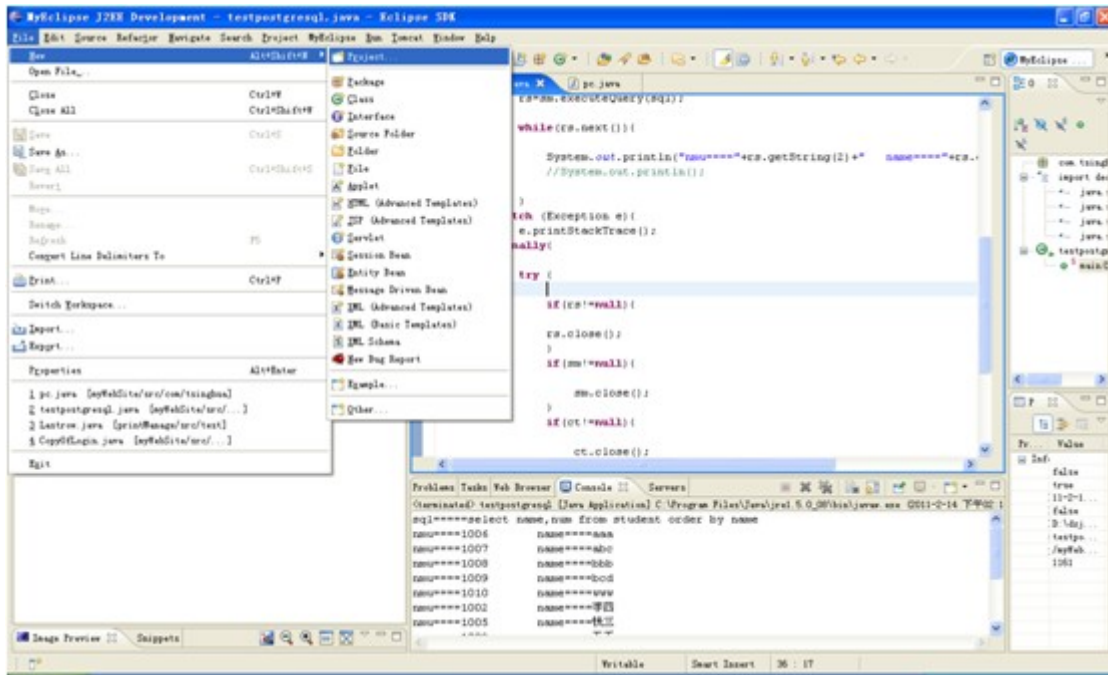
JAVA程序访问HighGo DB的环境准备

山东瀚高科技有限公司版权所有 仅允许不作任何修改的转载和转发

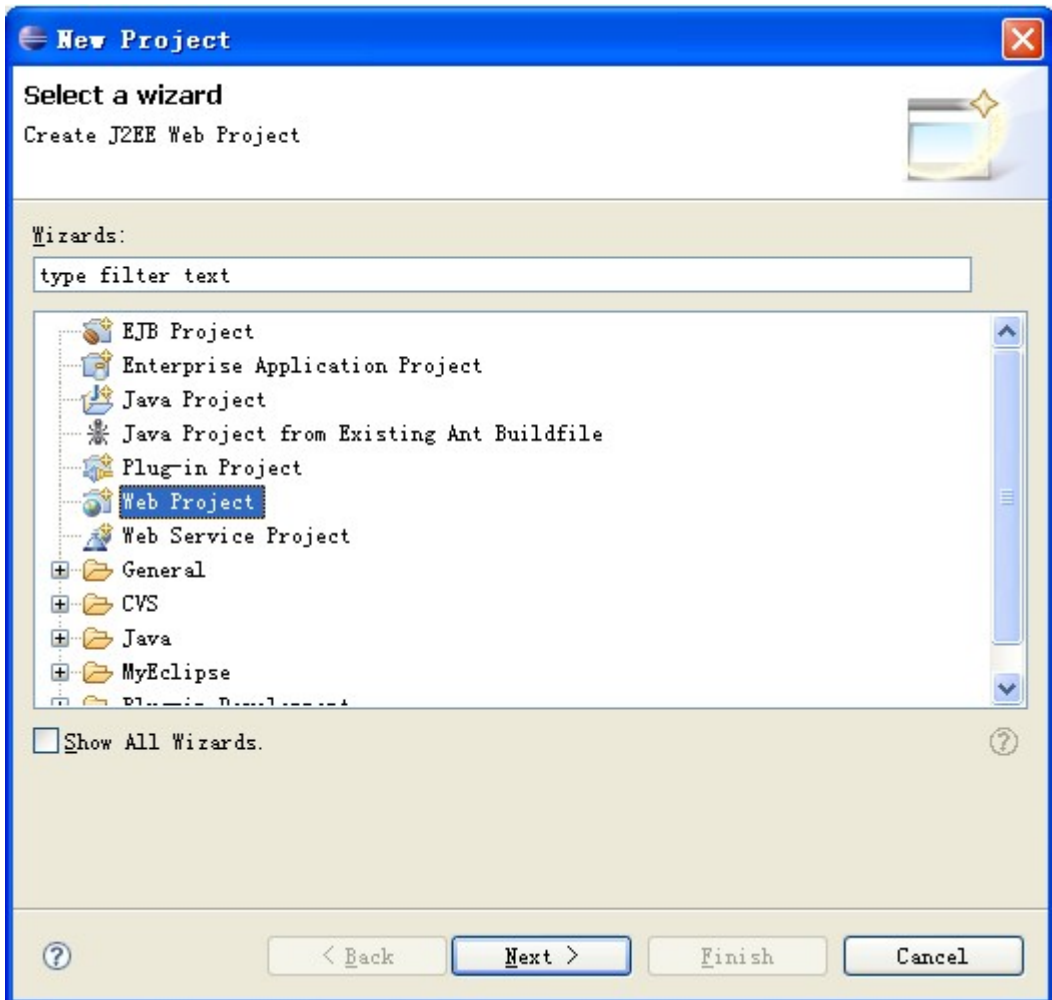
Hibernate的配置

MyEclipse中创建新项目:

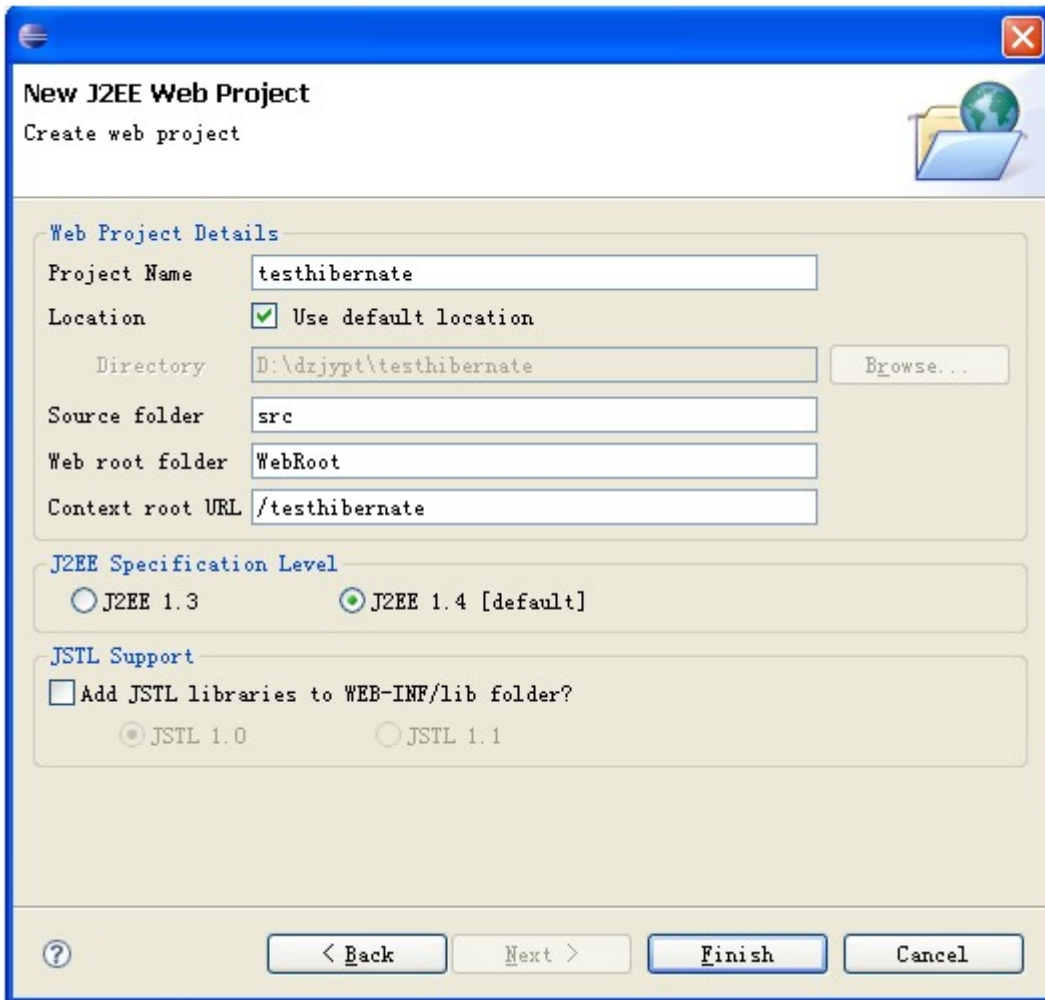
选择菜单栏file---new---project



选择web project进行下一步



输入project name 选择完成



4. 单击“添加JAR/文件夹”，会如下图出现JDBC

下载Hibernate jar包，并进行配置

1、网上下载Hibernate jar包如下：

antlr.jar

asm.jar

asm-attrs.jar

cglib-2.1.3.jar

commons-collections.jar

commons-logging.jar

dom4j-1.6.1.jar

jta.jar

log4j-1.2.11.jar

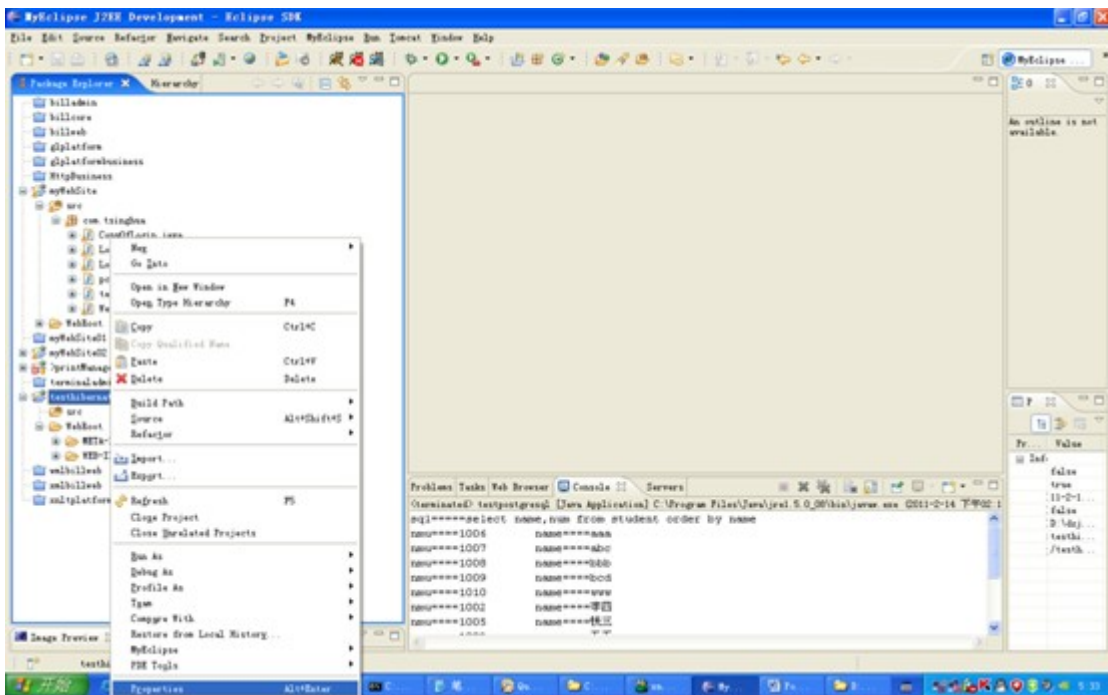
Hibernate3.jar

hgdb-1.1-jdbc4.jar (请下载相对应的版本，这里的JDK为1.6)

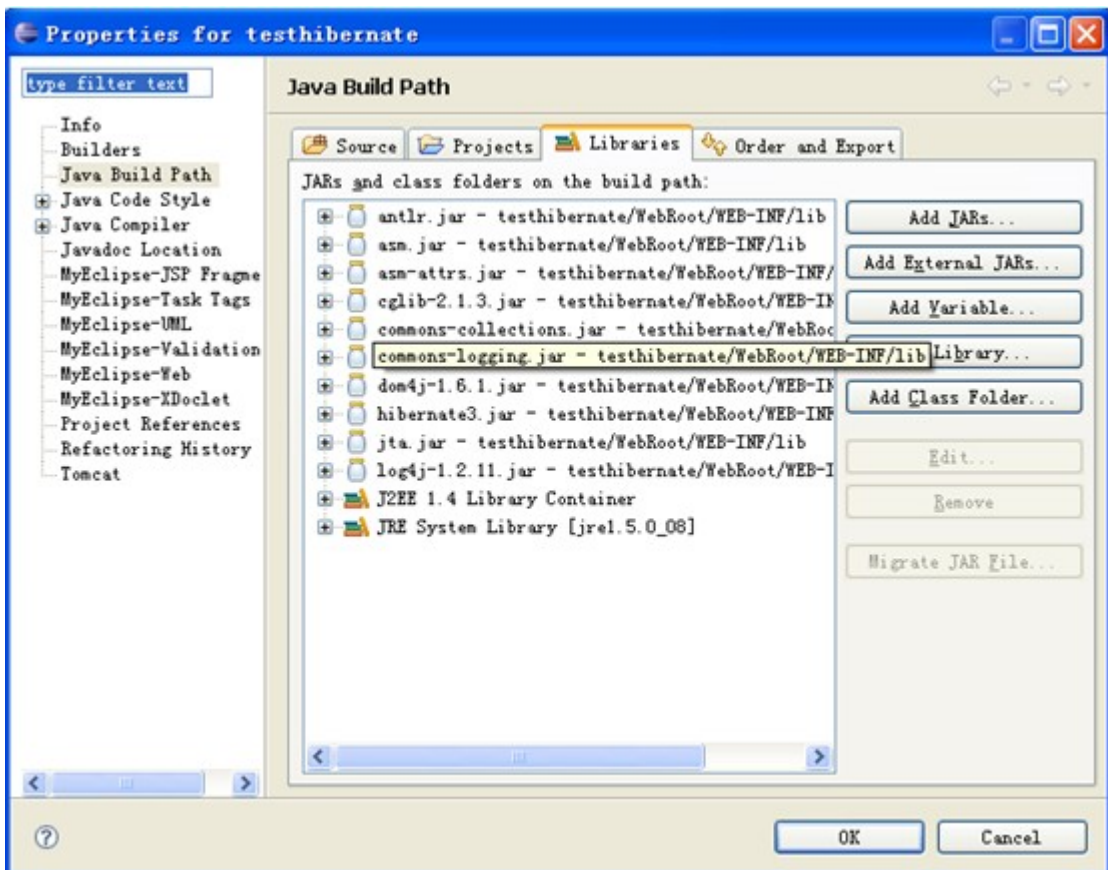
2、配置Hibernate jar包

将以上jar包拷贝到新的项目中去。比如：D:\testHibernate\WebRoot\WEB-INF\lib目录下面 查看一下testHibernate项目是否加载上：

首先刷新testHibernate项目, 然后选择testHibernate项目, 点击右键 选择properties



再选择java build path ---libraries



查看刚才拷贝的jar包是否 加载过来。

编写持久类 Student.java, 并创建Hibernate映射文件student.hbm.xml

1、编写一个学生对象类 Student.java

```

package model;

public class Student {
    private Integer num;
    private String name;
    private Integer age;
    private String addr;
    private String grade;

    //以下是get()/set()方法
    public String getAddr() {
        return addr;
    }
    public void setAddr(String addr) {
        this.addr = addr;
    }
    public Integer getAge() {
        return age;
    }
    public void setAge(Integer age) {
        this.age = age;
    }
    public String getGrade() {
        return grade;
    }
    public void setGrade(String grade) {
        this.grade = grade;
    }
    public String getName() {
        return name;
    }
}

```

2、编写Hibernate映射文件student.hbm.xml

```

<?xml version="1.0"?>
<!DOCTYPE Hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 2.0//EN"
    "http://Hibernate.sourceforge.net/Hibernate-mapping-3.0.dtd">
<Hibernate-mapping>
    <class name="model.Student" table="student">
        <id name="num" column="num" >
            <generator class="increment"/>
        </id>

        <property name="name" type="string">
            <column name="name" length="10" />
        </property>

        <property name="age" type="int">
            <column name="age" />
        </property>

        <property name="addr" type="string">
            <column name="addr" length="20" />
        </property>

        <property name="grade" type="string">
            <column name="grade" length="1" />
        </property>

    </class>
</Hibernate-mapping>

```

编写取得session的类 HibernateUtil.java

```

package persistence;

import org.Hibernate.HibernateException;
import org.Hibernate.Session;
import org.Hibernate.SessionFactory;
import org.Hibernate.cfg.Configuration;

public class HibernateUtil {
    private static final SessionFactory sessionFactory;
    static{
        try{
            sessionFactory=new
Configuration().configure().buildSessionFactory();
        }catch(HibernateException ex){
            throw new RuntimeException(ex.getMessage());
        }
    }

    public static Session currntSession() throws HibernateException{
        Session s=sessionFactory.openSession();
        return s;
    }
    public static void closeSession(Session s){
        if(s!=null){
            s.close();
        }
    }
}

```

编写操作数据库的Java文件 DAOImp. java

```

package persistence;

import java.util.List;

import model.Student;

import org.Hibernate.HibernateException;
import org.Hibernate.Query;
import org.Hibernate.Session;
import org.Hibernate.Transaction;

public class DAOImp {
    static Session session=null;
    //创建新的学生信息
    public static void createStu(Student stu){
        try{
            session=HibernateUtil.currntSession();
            Transaction tx=session.beginTransaction();
            session.save(stu);
            tx.commit();
        }catch(HibernateException ex){
            ex.printStackTrace();
        }finally{
            HibernateUtil.closeSession(session);
        }
    }
    //删除学生信息
    public static void delStu(Integer num){
        try{
            session=HibernateUtil.currntSession();

```

配置Hibernate的描述文件 Hibernate.cfg.xml及log4j初始化文件

1、配置Hibernate.cfg.xml

```

<?xml version="1.0" encoding='UTF-8'?>
<!DOCTYPE Hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://Hibernate.sourceforge.net/Hibernate-configuration-
3.0.dtd">
    <!-- Generated by MyEclipse Hibernate Tools. -->
<Hibernate-configuration>

<session-factory>
    <property name="connection.username">Administrator</property>
    <property name="connection.url">
        jdbc:highgo://localhost:5866/MyTest
    </property>
    <property name="connection.driver_class">
        com.highgo.jdbc.Driver
    </property>
    <mapping resource="model/Student.hbm.xml" />

</session-factory>

</Hibernate-configuration>

```

2、配置log4j.properties

```

#log4j.rootLogger=ERROR, INFO, A1 , R
#log4j.appender.ERROR=org.apache.log4j.ConsoleAppender
#log4j.appender.ERROR.layout=org.apache.log4j.PatternLayout
#log4j.appender.ERROR.layout.ConversionPattern=%-d{yyyy-MM-dd HH:mm:ss} [%c]-[%p] %m%n%l
#log4j.appender.INFO=org.apache.log4j.ConsoleAppender
#log4j.appender.INFO.layout=org.apache.log4j.PatternLayout
#log4j.appender.INFO.layout.ConversionPattern=%-d{yyyy-MM-dd HH:mm:ss} [%c]-[%p] %m%n%l
#log4j.appender.A1=org.apache.log4j.ConsoleAppender
#log4j.appender.A1.layout=org.apache.log4j.PatternLayout
#log4j.appender.A1.layout.ConversionPattern=%-d{yyyy-MM-dd HH:mm:ss} [%c]-[%p] %m%n%l
#log4j.appender.R=org.apache.log4j.RollingFileAppender
#log4j.appender.R.File=c:/Tomcat/logs/log4j.log
#log4j.appender.R.MaxFileSize=100KB
#log4j.appender.R.MaxBackupIndex=100
#log4j.appender.R.layout=org.apache.log4j.PatternLayout
#log4j.appender.R.layout.ConversionPattern=%-d{yyyy-MM-dd HH:mm:ss} [%c]-[%p] %m%n%l

log4j.rootLogger=info, CONSOLE
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.Target=System.out
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern= %4p [%t] (%F:%L) - %m%n

```

编写业务逻辑处理文件 BM.java

```
package BusinessManager;

import java.util.List;

import persistence.DAOImp;
import model.Student;

public class BM {
    public static void main(String[] args) {
        //创建新的学生对象

        Student stu=new Student();
        stu.setName("test1");
        stu.setAge(10);
        stu.setGrade("5");
        DAOImp.createStu(stu);

        //修改学生信息
        /*List list=DAOImp.getAllStu();
        Student stu1=(Student)list.get(0);
        stu1.setName("newname");
        stu1.setAge(11);
        DAOImp.mdStu(stu1);*/

        //删除学生信息
        /*List list1=DAOImp.getAllStu();
        Student stu2=(Student)list1.get(0);
        DAOImp.delStu(stu2.getNum());*/

    }
}
```

测试运行BM.java 并查看HighGo DB数据库中的结果

1、增加学生信息

运行BM.java文件

注意：BM.java代码

//创建新的学生对象

Student stu=new Student();

stu.setName("test1");

stu.setAge(10);

stu.setGrade("5");

DAOImp.createStu(stu);

查看HighGo DB数据库信息如下：

```
aaa=# select * from student;
 num | name | age | addr | grade
-----+-----+-----+-----+-----
  1  | test1 | 10  |      | 5
<1 行记录>
```

2、修改学生信息

运行BM.java文件

注意：BM.java代码

//修改学生信息

```
List list=DAOImp.getAllStu();
```

```
Student stu1=(Student)list.get(0);
```

```
stu1.setName("newname");
```

```
stu1.setAge(11);
```

```
DAOImp.mdStu(stu1);
```

查看HighGo DB数据库信息如下:

```
aaa=# select * from student;
 num | name  | age | addr | grade
-----+-----+-----+-----+-----
  1  | newname | 11  |      | 5
(1 行记录)
```

3、删除学生信息

运行BM.java文件

注意: BM.java代码

//删除学生信息

```
List list1=DAOImp.getAllStu();
```

```
Student stu2=(Student)list1.get(0);
```

```
DAOImp.delStu(stu2.getNum());
```

查看HighGo DB数据库信息如下:

```
aaa=# select * from student;
 num | name | age | addr | grade
-----+-----+-----+-----+-----
(0 行记录)
```