

课程名称: JavaEE 技术	年级: XXXX	实验日期:
姓名: XXX	学号: XXXXX	班级: XXXX
实验名称: Struts2 与 Hibernate 整合编程	实验序号: 实验六	成员人数: 1

一、实验目的及要求

实验目的:

- 1、MVC 的分层原理;
- 2、控制层(动作层)的方法调用、数据传递、页面转发;
- 3、模型层中 Hibernate 对数据的增删改查、hql 语句的利用;
- 4、熟悉 Struts 标签。

实验原理:

- 1、Hibernate 对 jdbc 进行了轻量级的封装。
- 2、利用 Hibernate 框架完成数据库中数据的增删改查, 以及业务类对方法的调用。
- 3、利用 Struts2 框架完成对模型层方法的调用, 在 struts.xml 配置文档中, 进行页面跳转的相关配置。

二、实验环境(仪器与材料)

Windows 7/10, MySQL/MariaDB, Eclipse, Tomcat

三、实验内容及完成情况

实验用数据库:

```
create database UsersManager
on
( name='UsersManager_data',
  filename='d:\UsersManager_data.mdf',
  size=10mb
)
log on
( name='UsersManager_log',
  filename='d:\UsersManager_log.ldf',
  size=10mb
)
go
use UsersManager
go
create table TUserLogin
(
  id int identity(1,1) primary key,
  username nvarchar(20) not null,
  [password] nvarchar(20),
  grade int default 1,
  email nvarchar(50)
)
```

1、新建 Web Project

建立一个新的项目, 新建 Web Project, 取名为 register, 如图 1 所示。

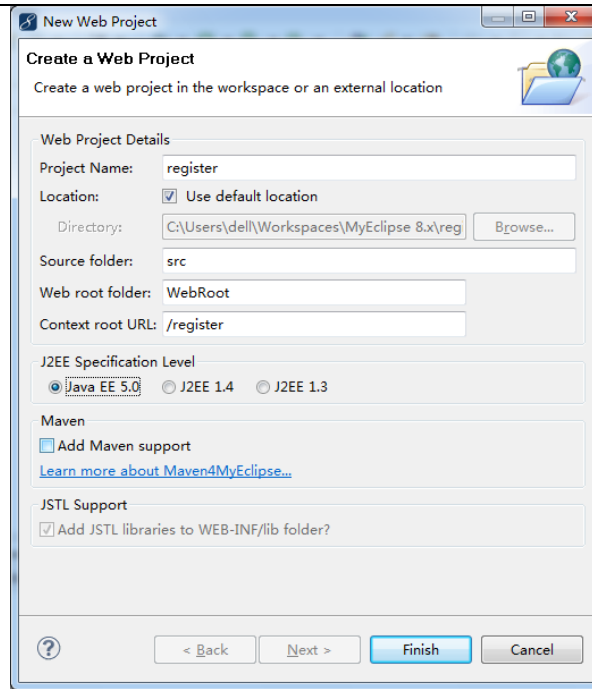


图 1 新建 Web Project

2、增加 Struts2 的能力支持

在这个项目中，使用 MyEclipse 自带的菜单选项，增加 Struts2 的能力支持。这样，系统会自动增加 jar 包和 xml 配置文件，如图 2 所示。

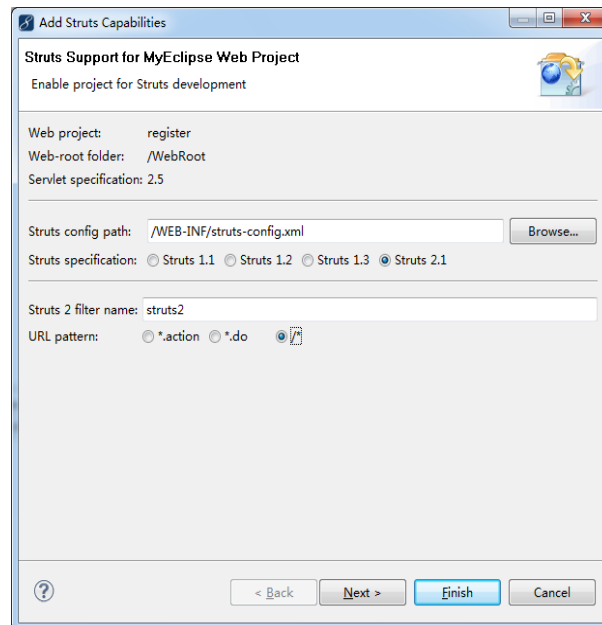


图 2 添加 Struts Capabilities

3、自动创建 struts.xml

MyEclipse 会自动在 src 下面创建一个 struts.xml，并且在 WEB-INF 文件夹下面的 web.xml 文件中自动添加下面的语句：

```
<filter>
```

```

<filter-name>struts2</filter-name>
<filter-class>org.apache.struts2.dispatcher.ng.filter.
StrutsPrepareAndExecuteFilter
</filter-class>
</filter>
<filter-mapping>
<filter-name>struts2</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>

```

4、增加 Hibernate 的能力支持

在这个项目中，使用 MyEclipse 自带的菜单选项，增加 Hibernate 的能力支持。这样，系统会自动增加 jar 包和 xml 配置文件，如图 3 所示。

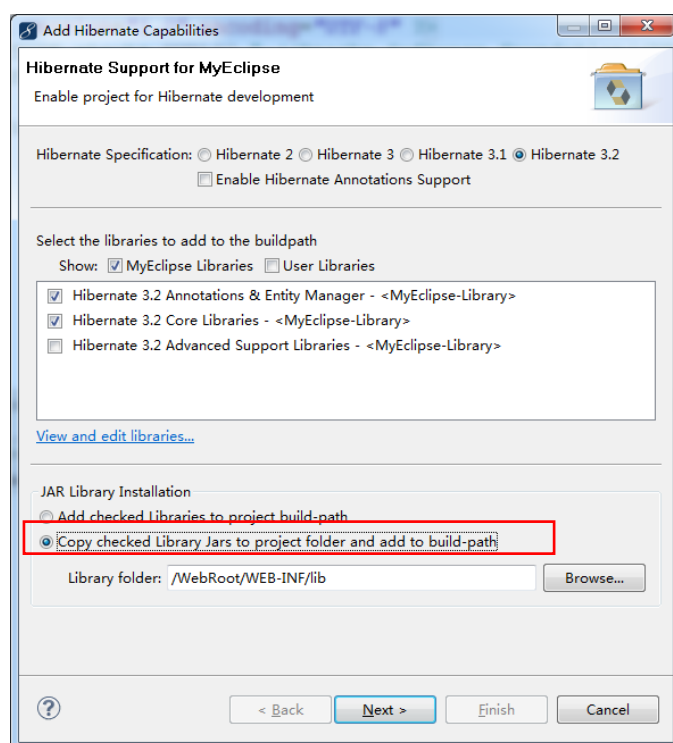


图 3 添加 Hibernate Capabilities

注意：需要选择第二项，将 jar 包拷贝到项目的 lib 文件夹下，这样，TomCat 启动时会从 lib 中找到需要加载的 jar 包，就不会报错。而选择第一项，编译时是没有问题的，但是 TomCat 启动会报：找不到 jar 包的错误。

5、新建 hibernate.cfg.xml 文件

在图 3 中，点击 next，进入下一步，出现图 4 所示的配置选项，选择新建 hibernate.cfg.xml 文件。

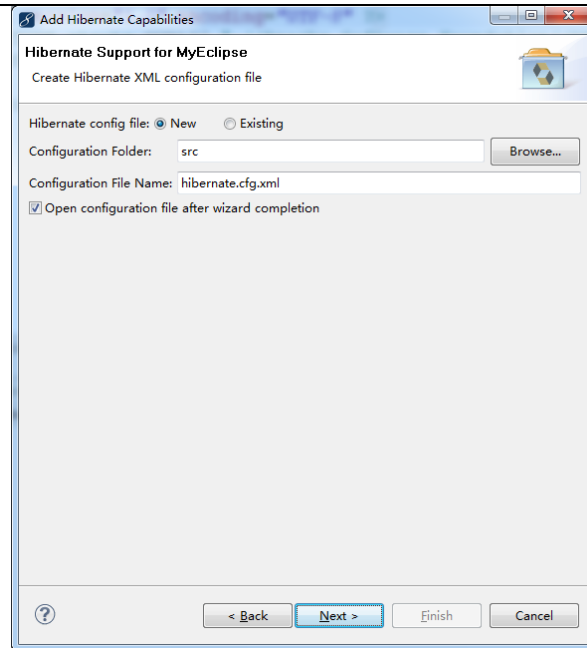


图 4 选择新建 hibernate.cfg.xml 文件

6、手工配置数据库连接细节

在图 4 中，点击 next，进入下一步，出现图 5 所示的配置细节，把选项 “Specify Hibernate database connection details?”选项设置为不选中（默认选中），不指定数据库连接的详情。完全利用手工进行配置即可。

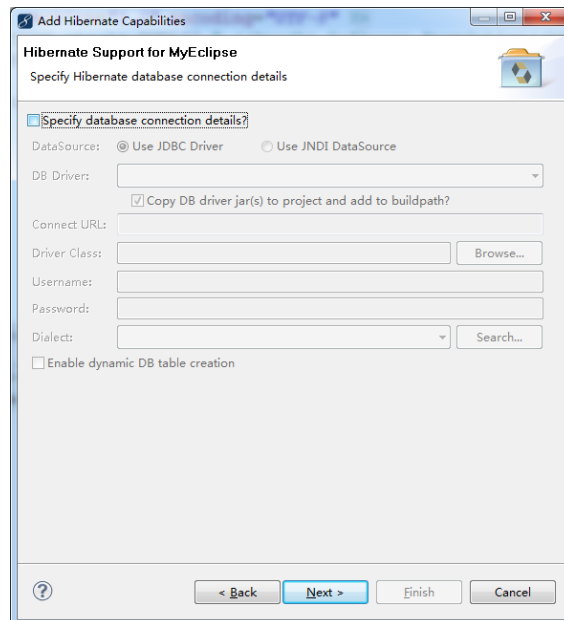


图 5 不指定数据库连接的详情

7、不创建 SessionFactory

在图 5 中，点击 next，进入下一步，出现图 6 所示的配置细节，询问是否需要创建 SessionFactory，选择不需要。最后点击 “Finish”即可。

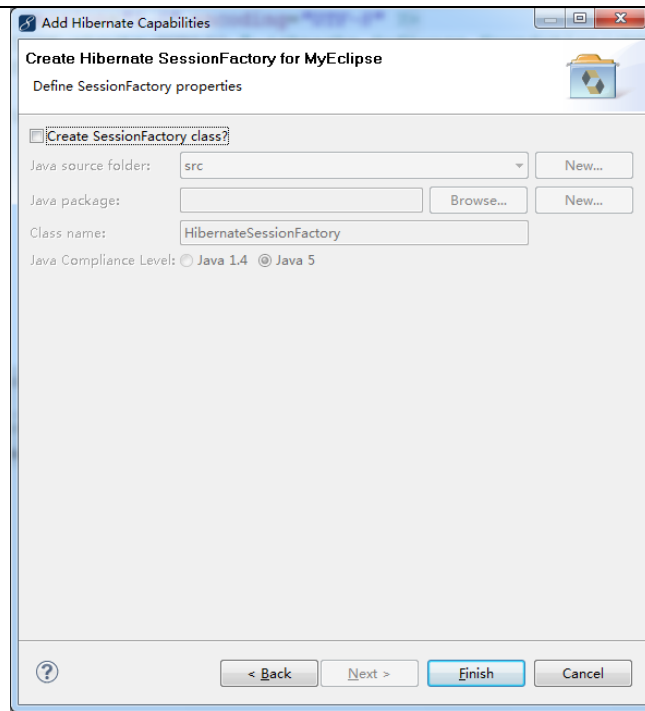


图 6 不创建 SessionFactory

8、显示 Hibernate.cfg.xml 配置信息

自动出现 Hibernate.cfg.xml 的图形化配置信息，切换至“Source”，出现

```
<?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>
    </session-factory>
</hibernate-configuration>
```

9、加载 Sql Server 的驱动

加载 Sql Server 的驱动,将 sqljdbc4.jar 放置到 lib 文件夹下。

10、详细配置 Hibernate.cfg.xml 配置文件

```
<hibernate-configuration>
    <session-factory>
    <property name="connection.url">jdbc:sqlserver://localhost:1433;
DatabaseName=UsersManager</property>
    <property name="connection.username">sa</property>
    <property name="connection.password">123456</property>
    <property
```

```

name="connection.driver_class">com.microsoft.sqlserver.jdbc.
SQLServerDriver</property>
<property name="dialect">org.hibernate.dialect.
SQLServerDialect</property>
<property name="show_sql">true</property>
</session-factory>
</hibernate-configuration>

```

11、创建 User 类

准备创建持久化的类，先创建包：com.bbc.model，在这个包下面创建 User 类

```

public class User {
    //写上User的属性，要和数据库中的user表 一一对应上
    private int id;
    private String username;
    private String password;
    private int grade;
    private String email;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public int getGrade() {
        return grade;
    }
    public void setGrade(int grade) {
        this.grade = grade;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
}

```

12、创建 register.jsp 页面

在 WebRoot 下创建 register.jsp 页面。

```
<body>
  <form action="saveUser.action" method="post">
    用户名: <input type="text" name="username"/>
    密 码: <input type="password" name="password"/>
    电子邮件: <input type="text" name="email"/>
    <input type="submit" value="注册"/>
  </form>
</body>
```

注：等级自动注册为1，用户编号自动分配，无需用户输入

13、创建 UserAction 类

创建 com.bbc.action 包，在这个包下面创建 UserAction 类，继承自 ActionSupport。

```
public class UserAction extends ActionSupport {
    private String username;
    private String password;
    private String email;
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String saveUser() throws Exception{
        User user=new User();
        user.setUsername(username);
        user.setPassword(password);
        user.setEmail(email);
        user.setGrade(1); //注册，用户等级默认为1
        //还有部分代码暂时无法书写
    }
}
```

14、配置 struts.xml 文件

```
<struts>
```

```

    <package name="register" extends="struts-default">
    <action name="saveUser" class="com.bbc.action.UserAction"
method="saveUser">
    </action>
    </package>
</struts>

```

15、创建 UserDao 接口

创建 com.bbc.dao 包，在这个包下面创建 UserDao 接口。

```

package com.bbc.dao;
import com.bbc.model.User;
public interface UserDao {
    public void saveUser(User user);
}

```

16、创建 UserDao 接口的实现类

新建 com.bbc.dao.impl 包，在这个包下面创建 UserDao 接口的实现类---UserDAOImpl。

```

package com.bbc.dao.impl;
import com.bbc.dao.UserDao;
import com.bbc.model.User;
public class UserDaoImpl implements UserDao {
    public void saveUser(User user) {
    }
}

```

17、创建 HibernateUtil 类

新建 com.bbc.util 包，在这个包下面创建 HibernateUtil 类

```

package com.bbc.util;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class HibernateUtil {
    private static SessionFactory sessionFactory;
    static{
        try{
            sessionFactory =new Configuration().configure().
buildSessionFactory();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
    public static Session openSession(){
        Session session=sessionFactory.openSession();
        return session;
    }
    public static void close(Session session){

```



```

        if (null != session) {
            session.close();
        }
    }
}

```

18、创建 User.hbm.xml 文件

在 src 下面创建 User.hbm.xml 文件

```

<hibernate-mapping>
<class name="com.bbc.model.User" table="TUserLogin">
    <id name="id" column="id" type="int">
        <generator class="identity"></generator>
    </id>
    <property name="username" column="username"
type="string"></property>
    <property name="password" column="password"
type="string"></property>
    <property name="email" column="email"
type="string"></property>
</class>
</hibernate-mapping>

```

19、Hibernate.cfg.xml 增加映射

再次打开 Hibernate.cfg.xml 配置文件，需要加上一句映射语句

```

<mapping resource="User.hbm.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.url">jdbc:sqlserver://localhost:1433;DatabaseName=Us
ersManager</property>
        <property name="connection.username">sa</property>
        <property name="connection.password">123456</property>
        <property
name="connection.driver_class">com.microsoft.sqlserver.jdbc.SQLServer
Driver</property>
        <property
name="dialect">org.hibernate.dialect.SQLServerDialect</property>
        <property name="show_sql">>true</property>
        <mapping resource="User.hbm.xml"/>
    </session-factory>
</hibernate-configuration>

```

20、完善 UserDaoImpl 类

```
package com.bbc.dao.impl;
import org.hibernate.Session;
import org.hibernate.Transaction;
import com.bbc.dao.UserDAO;
import com.bbc.model.User;
import com.bbc.util.HibernateUtil;
public class UserDaoImpl implements UserDAO {
    public void saveUser(User user) {
        Session session=HibernateUtil.openSession();
        Transaction tx=session.beginTransaction();
        try{
            session.save(user);
            tx.commit();
        }catch(Exception ex){
            if(null!=tx){
                tx.rollback();
            }
        }finally{
            HibernateUtil.close(session);
        }
    }
}
```

21、创建 UserService 接口

新建 com.bbc.service 包，在这个包下面创建 UserService 接口

```
package com.bbc.service;
import com.bbc.model.User;
public interface UserService {
    public void saveUser(User user);
}
```

22、创建 UserServiceImpl 类

新建 com.bbc.service.impl 包，在这个包下面创建 UserServiceImpl 类

```
package com.bbc.service.impl;
import com.bbc.dao.UserDAO;
import com.bbc.dao.impl.UserDAOImpl;
import com.bbc.model.User;
import com.bbc.service.UserService;
public class UserServiceImpl implements UserService {
    public void saveUser(User user) {
        UserDAO userDAO=new UserDAOImpl();
        userDAO.saveUser(user);
    }
}
```

23、完善 UserAction 代码

切换至UserAction中，继续书写剩余的代码。

```
public String saveUser() throws Exception{
    User user=new User();
    user.setUsername(username);
    user.setPassword(password);
    user.setEmail(email);
    user.setGrade(1);//注册，用户等级默认为1
    UserService userService=new UserServiceImpl();
    userService.saveUser(user);
    return SUCCESS;}

```

在地址栏输入：http://localhost:8080/register/register.jsp，如图7所示。



图7 登陆界面

输入如下信息，如图8所示。

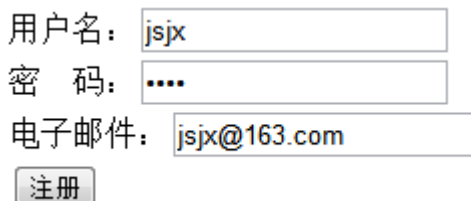


图8 录入信息

观察服务器后台信息，显示：

Hibernate: insert into TUserLogin (username, password, email) values
(?, ?, ?)

打开数据库并查看，发现信息已经录入。

四、 出现的问题及解决方案

1、HTTP Status 404 - No result defined for action action.EntryAction and result input

检查一下有没有中英文符号错误，有没有struts标签错误等

2、Exception starting filter struts2 java.lang.NullPointerException

一般是因为struts.xml文件中配置了开发模式，也有可能是commons-fileupload和commons-io-1.3.2这两个jar包没有导入，还有可能就是Tomcat安装路径里有空格。