

# MyEclipse整合Struts2+Spring+Hibernate

虽然已经有很多文章介绍整合 Struts2、Spring2.5、Hibernate3.1 来开发基于 Web Project 项目的过程，但是里边有很多细节描述不是很清楚，对初学者造成了很大的困扰，特别是测试运行后出现很多不知所然的错误，往往并不是程序本身错误造成的，而是配置细节出现的问题，这里将每个过程及配置中需要注意的细节一一说明，希望大家在初学的道路上走的更加平坦。

## （一）创建Web Project

这个过程很简单，通过 MyEclipse8.5 的创建向导逐步完成就可以了，建议 JAVA 支持方面选择 JAVA EE5.0 这个版本。

在创建 Web Project 之前最好规划好项目的目录结构，这里有两个地方需要考虑目录结构，一个是 WebRoot 下，这里存放的都是项目表示层的程序及文档，即 JSP、javascript、CSS、图片及浏览器可访问的内容，可以按照项目功能模块分配目录；另一个是 src 下，这里存放的是 struts2、spring、hibernate 的配置文件，JAVA 程序等，JAVA 程序按照包分类管理，最好 struts2 的 action 程序放一个包，hibernate 的 DAO 放一个包，Hibernate Factory 放一个包，每个包下也可以按照模块再分包管理。

## （二）建立MySql数据库

通过 MySql 的管理工具创建一个数据库，并且记住数据库的访问 URL，用户名，密码，确定数据库各个表的名字，字段等，这个设计最好在系统详细设计阶段，因为在项目开发过程中修改数据库结构会给开发进程造成很大影响。

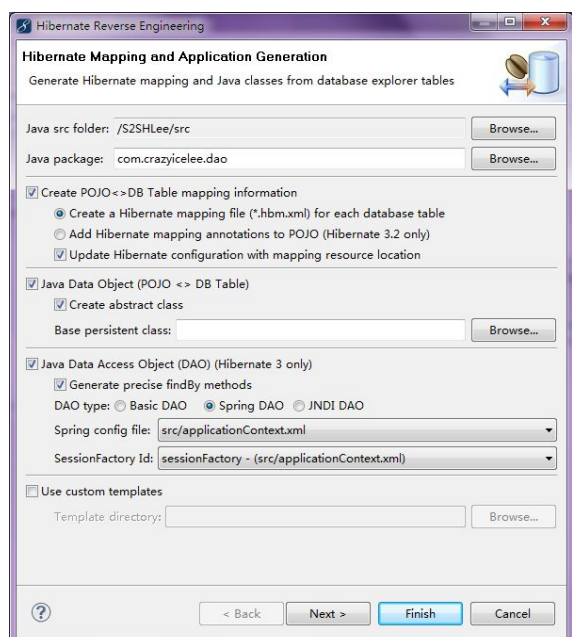
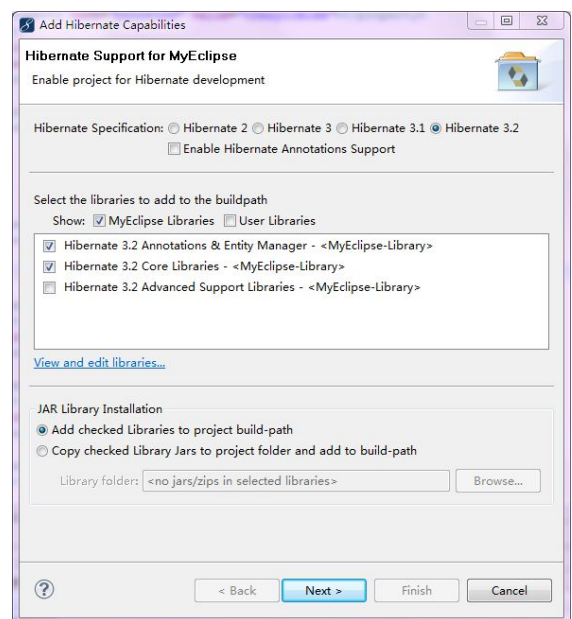
利用 MyEclipse 的 Db Browser 可以连接到该项目对应的数据库，随时查看数据库的结构及记录，具体操作见 MyEclipse8.5 IDE，这里不再赘述。

### (三) 引入Spring2.5

- 将鼠标定位到 Web Project 项目名称上，单击右键，依次选择菜单 MyEclipse->Add Spring Capabilities。
- 选择“Spring 2.5 AOP”，“Spring 2.5 Core”，“Spring 2.5 Persistence Core”，“Spring 2.5 Persistence JDBC”，“Spring 2.5 Web”等库；
- 配置 Spring 配置文件存放路径及名称，都采用缺省配置即可，配置文件存放在 src 目录下，配置文件名称为 applicationContext.xml；

### (四) 引入Hibernate3

- 将鼠标定位到 Web Project 项目名称上，单击右键，依次选择菜单 MyEclipse->Add Hibernate Capabilities。
- 选择 Hibernate3.1，其他项缺省即可。
- 点击“Next”按钮，设置配置文件存放位置及名称，选择新建一个配置文件，都用系统给出的缺省值即可。
- 点击“Next”按钮，进入数据库连接配置界面，选择一个已经配置好的DBDriver，其他项都缺省选择。
- 点击“Next”按钮，进入“Hibernate SessionFactory”创建窗口，这里设置一下“Java Package”即是前面规划好的目录名称，点击“new”创建包，建议取名为 xxx.xxxx.hibernate。
- 创建 POJO 映射
  - 打开 DB Brower，选择目标数据库的某个表 xxx；



- 单击右键，选择菜单“Hibernate Reverse engineering”，按照上图所示选择各项配置；
- 单击“Next”按钮，选择数据库的主键生成器，这个是必选项；
- 一路点击“Next”按钮下去，系统会自动生成 xxxDAO.java（对应数据库表的 JAVA 文件），xxx.java（对应数据记录的 JAVA 文件），还有其他抽象类 JAVA 文件

## （五）测试Spring和Hibernate的整合

经过上面两个引入环节，Spring 已经和 Hibernate 整合在了一起，下面就使用 MyEclipse 的 File->New->Class 编写一个 Java 测试代码来测试整合操作是否正确，实例代码如下：

```
package com.crazyicelee.test;

import org.springframework.beans.BeansException;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.FileSystemXmlApplicationContext;

import com.crazyicelee.dao.User;
import com.crazyicelee.dao.UserDAO;

/**
 * @author crazyicelee
 *
 */

public class test {

    /**
     * @param args
     */

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        try {
```

```

//下面两句获取上下文及 hibernate 的数据模型对象 DAO
ApplicationContext ctx = new
FileSystemXmlApplicationContext("G:\\myself\\MyEclipse\\S2SHTest\\src\\applicati
onContext.xml");

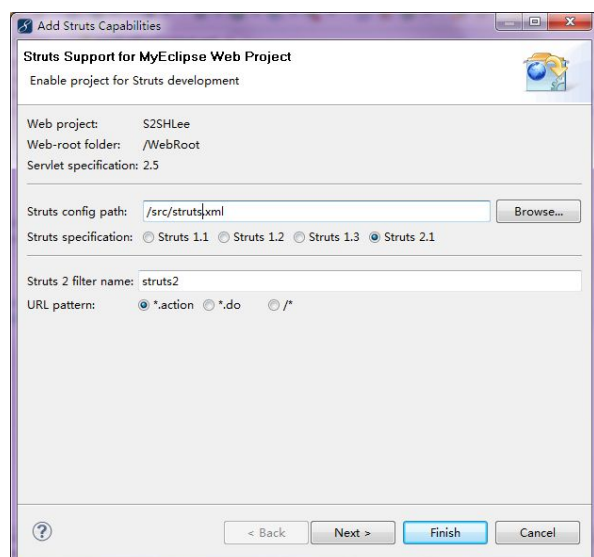
UserDAO userDAO = (UserDAO) ctx.getBean("UserDAO");
User u=new User();
u.setName("111");
u.setCity("w");
u.setCountry("sss");
u.setCreditLimit(new Integer(10));
userDAO.save(u);
} catch (BeansException e) {
    System.out.println(e.getMessage());
}
catch(Exception ee){
    System.out.println(ee.getMessage());
}
}
}

```

配置 Java Console 运行环境，并运行，调试上面代码，如果报错，可以检查错误信息，一般是某个 Spring 库没有包含进来，可以通过“Properties”的“Java Build Path”的“Libraries”添加，如果运行不再报错，则查看数据库应该已经成功写入数据，说明 Spring 和 Hibernate 整合成功。

## （六）整合Struts2

- 将鼠标定位到 Web Project 项目名



称上，单击右键，依次选择菜单 MyEclipse->Add Struts Capabilities。

- 修改配置文件存放路径及名称，通常存放在 src 目录下，配置文件名称为 struts.xml，并选择 Struts 版本为 Struts2.1，选择 Struts2 的 URL 模板为/\*使得所有请求都通过 Struts2 管理，当然后续会把一部分请求托管给 Spring 容器。

- 选择 Struts2 支持库，至少包含 “Struts 2 Core”，“Struts 2 Spring”；

- 利用 MyEclipse 的 File->New->Class 创建 Action 代码，选择基类为 ActionSupport，并设定包名称，系统会生成一个简单的程序框架，在程序编辑区单击鼠标右键，选择 Source->Override/Implement Methods，选择要重置的方法，至少要实现 execute ()，在这个方法中编写逻辑处理代码。具体代码编写规则及方法参考 Action 程序编写的有关教程。这里重点强调整合 Struts2 和 Spring 时要添加的代码：

- 将 xxxDAO 作为 Spring 要注入的属性添加到 Action 里边，本实例中将 UserDao 作为 Action 的私有属性添加进来，并配置 Bean 的 getter/setter 方法；

- 配置其他 Action 属性及 getter/setter 方法；

- 实例代码如下：

```
package com.crazyicelee.action;

import java.util.List;
import com.crazyicelee.dao.User;
import com.crazyicelee.dao.UserDAO;
import com.opensymphony.xwork2.ActionSupport;

/**
 * @author crazyicelee
 *
 */
public class ListUser extends ActionSupport {

    private UserDAO userDAO;//Spring 注入的 DAO Bean 对象
    private List<User> users;

    /* (non-Javadoc)
```

```

    * @see com.opensymphony.xwork2.ActionSupport#execute()
    */
    @SuppressWarnings("unchecked")
    @Override
    public String execute() throws Exception {
        // TODO Auto-generated method stub
        users=userDAO.findAll();
        return super.execute();
    }

    /**
     * @param userDAO the userDAO to set
     */
    public void setUserDAO(UserDAO userDAO) {
        this.userDAO = userDAO;
    }

    /**
     * @return the userDAO
     */
    public UserDAO getUserDAO() {
        return userDAO;
    }

    /**
     * @param users the users to set
     */
    public void setUsers(List<User> users) {
        this.users = users;
    }

```

```

/**
 * @return the users
 */
public List<User> getUsers() {
    return users;
}
}

```

- 配置 struts.xml，配置方法参考相关文章

## (七) 整合Struts2 和Spring

这一步整合是最为麻烦的，因为到目前为止 MyEclipse8.5 M1 的版本还不支持可视化自动配置，而是需要手动修改配置文件完成配置，这里有一个基本原则，就是 Spring 将接管一部分 Struts2 的 Action，即凡是需要通过 Hibernate 访问数据的 Action 都需要 Spring 接管，其他不需要通过 DAO 存取数据的就不需要托管给 Spring。

- Spring 配置文件 applicationContext.xml 的修改，这里需要告诉 Spring 要托管的 Action 及属性，每个 Action 就是一个 Bean，也就是向配置文件中添加一个 Bean 声明，代码如下：

```

<bean id="ListUser" class="com.crazyicelee.action.ListUser"
scope="prototype">
    <property name="userDAO">
        <ref bean="UserDAO" />
    </property>
</bean>

```

其中：

id 的值是对应到 struts2 配置文件 struts.xml 中的 Action 标签中的 class 属性；  
class 的值是实现这个 Action 的具体类名称。

上面配置文件说明了将一个 DAO 注入到一个 Action 体内，也就是完成了数据存取对象初始化工作，在 Action 里边就可以直接使用这个 DAO 对象进行相应的操作。

■ Struts2 配置文件 struts.xml 修改，首先添加一个全局常量，告诉 Action 由 Spring 托管，其次修改被 Spring 托管的 Action 的 class，将原来单纯 Struts2 管理的完全类名称改为 Spring 的相应 Bean 的 id 值（这点非常重要，否则运行出错），代码如下：

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts
Configuration 2.1//EN"
"http://struts.apache.org/dtds/struts-2.1.dtd">
<struts>
  <!-- struts2 委托Spring管理 -->
  <constant name="struts.objectFactory" value="spring"/>
  <package name="struts2" namespace="/mystruts"
extends="struts-default">
    <action name="ListUser" class="ListUser">
      <result name="success">/index.jsp</result>
      <result name="failue">/login.jsp</result>
    </action>
  </package>
</struts>
```

■ Web 服务器配置文件 Web.xml 修改，这里要告诉在 web 服务启动的时候启用 Spring 监听器和过滤器，从而优先在 struts2 之前接管有关 Action，配置文件如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <!-- 配置Spring监听器 -->
  <context-param>
    <param-name>contextConfigLocation</param-name>

    <param-value>/WEB-INF/classes/applicationContext.xml</param-va
lue>
  </context-param>
  <listener>

    <listener-class>org.springframework.web.context.ContextLoaderL
istener</listener-class>
  </listener>
  <!-- 配置openSessionInViewFilter，必须在Struts监听之前 -->
```



```

<filter>
    <filter-name>lazyLoadingFilter</filter-name>

    <filter-class>org.springframework.orm.hibernate3.support.OpenS
essionInViewFilter</filter-class>
</filter>

<welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
<filter>
    <filter-name>struts2</filter-name>
    <filter-class>

org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecut
eFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>lazyLoadingFilter</filter-name>
    <url-pattern>*.action</url-pattern>
</filter-mapping>
<filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
</web-app>

```

上面标红的部分是添加的，注意前后的顺序不能颠倒。

■ 添加 Struts 2 Spring 支持库，通过 Java Build Path 的 Libraries 添加这个库，如果缺少这个库，则加载 web.xml 时出错，致使服务器不能启动。

## （八）编写JSP代码

这个 JSP 代码是 Action 后调用的页面，将把 Action 获取的数据显示在浏览器端，代码如下：

```

<%@ page language="java" import="java.util.*"
pageEncoding="GB18030"%>
<%@ taglib uri="/struts-tags" prefix="s"%>
<%
String path = request.getContextPath();
String basePath =
request.getScheme()+"://"+request.getServerName()+":"+request.get

```

```

ServerPort()+path+"/";
%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <base href="<%=basePath%>">
    <title>My JSP 'index.jsp' starting page</title>
  </head>

  <body>
    <table border="1">
      <s:iterator id="listUser" value="users">
        <tr>
          <td>姓名: </td><td><s:property value="name" /></td>
        </tr>
        <tr>
          <td>国家: </td><td><s:property value="country" /></td>
        </tr>
        <tr>
          <td>城市: </td><td><s:property value="city" /></td>
        </tr>
        <tr>
          <td>信用额度: </td><td><s:property
value="creditLimit" /></td>
        </tr>
      </s:iterator>
    </table>
  </body>
</html>

```

## （九）部署运行

利用 MyEclipse8.5 M1 的部署工具将这个项目部署到 Resin 服务器上，注意在服务器启动前删除一个 Spring 和 Hibernate 冲突的 Jar 包（/WEB-INF/lib 下的 asm-2.2.3.jar），然后启动 Resin，在浏览器上打开 <http://localhost:8080/S2SHLee/mystruts/ListUser.action> 就可以看到这个 Web Project 运行的效果了，至此，所有配置编码工作顺利完成。