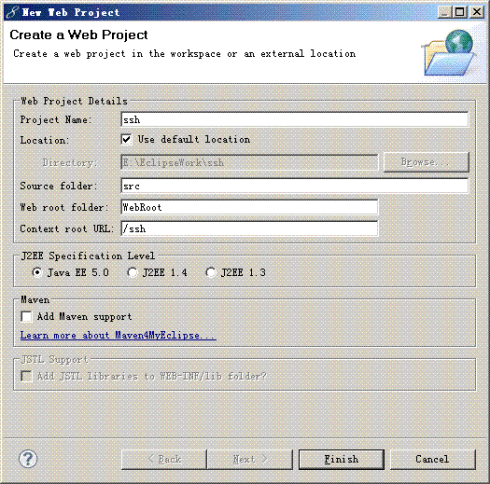
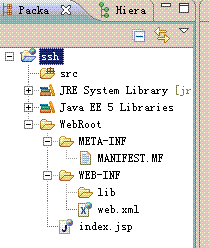
**1.创建项目**

打开Myeclipse8.6，新建web工程，命名为ssh，点击完成。



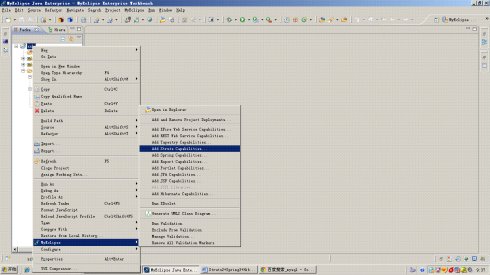


获得目录结构如下：

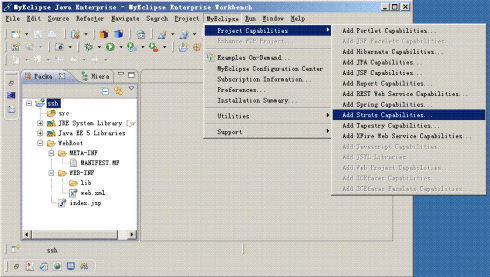


**2为项目添加Struts能力**

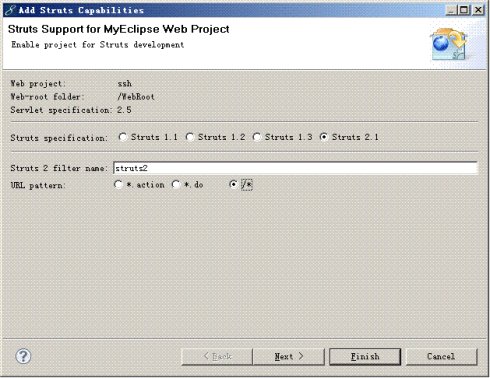
a)右键点击项目，选择MyEclipse->Add Struts Capabilities，或者点击菜单栏中MyEclipse->Project Capabilities->Add Struts Capabilities。



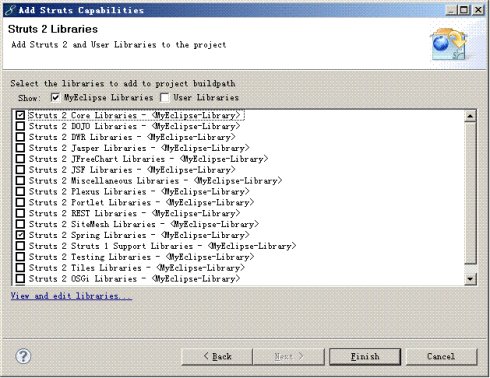
b)选择struts2.1版本



c)点击下一步，选择Struts Core(核心库)和Struts 2 Spring(包含集成Spring插件)的两个最基本库，其他根据需要可以添加。



d)点击完成，添加后的项目多了一个Struts.xml（用于配置action），web.xml中添加了核心过滤器的配置。



<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>

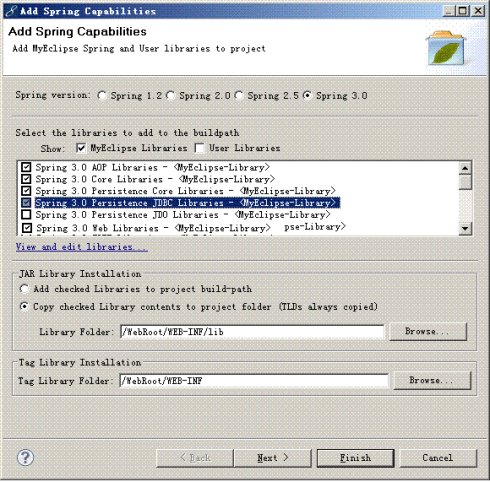
<url-pattern>/\*</url-pattern>

</filter-mapping>

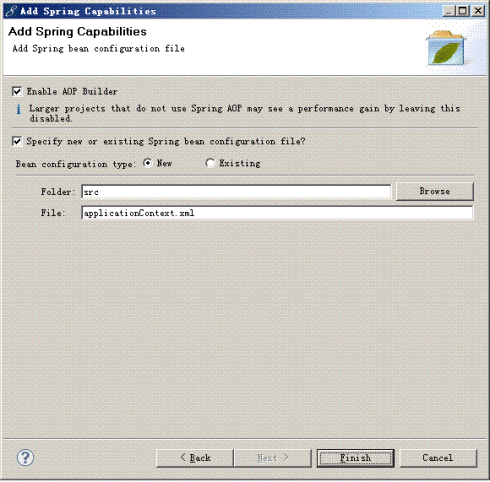
**3.为项目添加Spring能力**

a)右键点击项目，选择MyEclipse->Add Spring Capabilities，或者点击菜单栏中MyEclipse->Project Capabilities->Add Spring Capabilities。

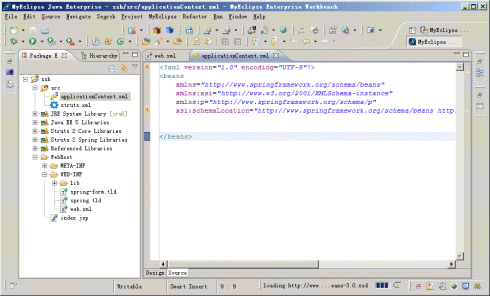
b)选择Spring3.0版本，类库选择 Spring 3.0 AOP、Spring 3.0 Core、Spring 3.0 Persistence Core、Spring 3.0 Persistence JDBC、Spring 3.0 Web 5个库，并勾选复制到本地WEB-INF/lib目录。



c)点击下一步，默认配置不需修改。



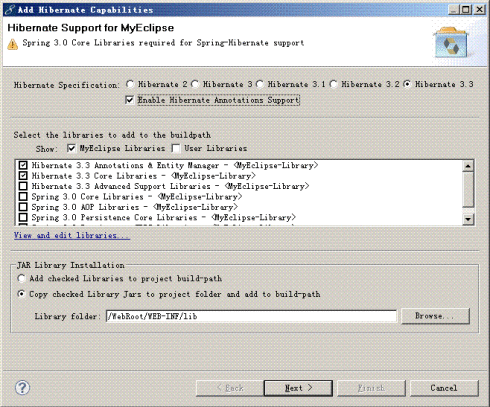
d)点击完成，查看添加Spring后的项目，增加了applicationContex.xml文件



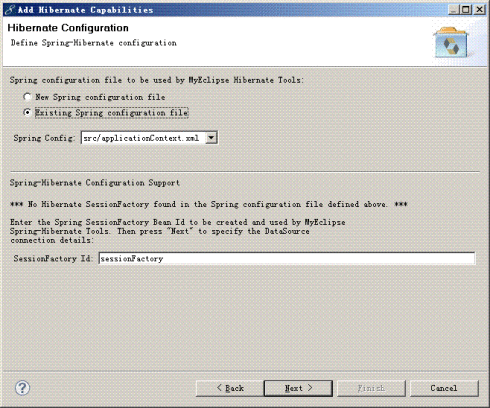
**4.为项目添加Hibernate能力**

a)右键点击项目，选择MyEclipse->Add Hibernate Capabilities，或者点击菜单栏中MyEclipse->Project Capabilities->Add Hibernate Capabilities。

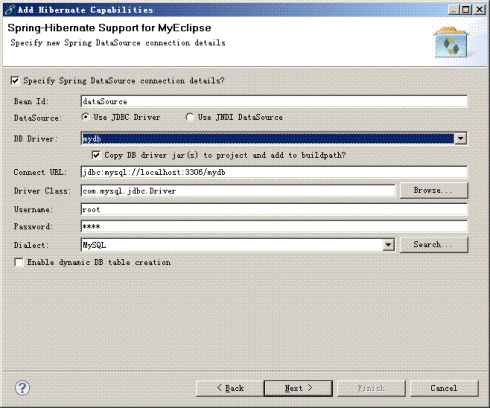
b)选择Hibernate3.3版本，如果要使用注记可选中启用annotation，但是会对后续自动生成实体类造成麻烦，不选就是用xml文件形式进行实体类映射，默认勾选Hibernate 3.3 Annotation & Entity Manager、Hibernate 3.3 Core Libraries两个类库，复制jar文件到本地WEB-INF/lib文件夹下。



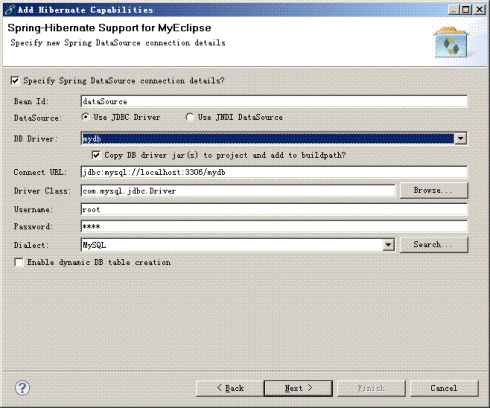
c)点击下一步，选择使用Spring的配置文件applicationContext.xml，这样就不会生成hibernate.cfg.xml，生成与否看个人习惯。



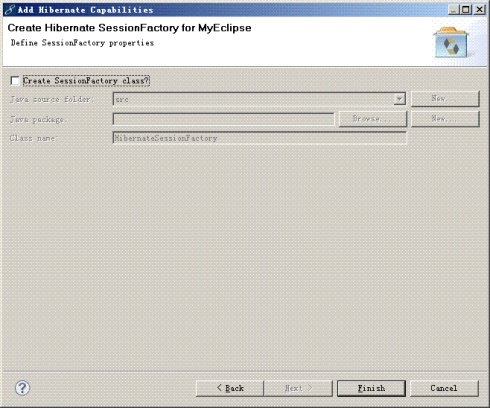
d)点击下一步，选择已经存在的applicationContext.xml文件的位置以及创建sessionFactory的名字



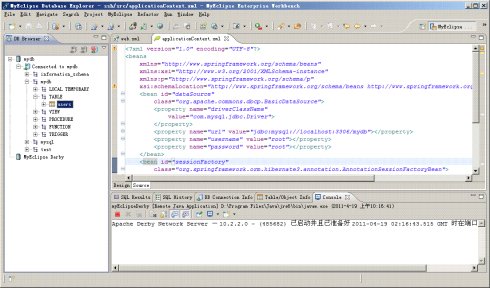
e)下一步，选择已经建立好的数据库连接mydb



f)点击下一步，取消勾选，不生成HibernateSessionFactory类

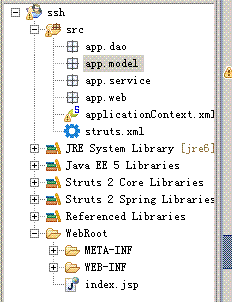


g)点击完成，查看工程，applicationContext.xml文件中增加了数据库连接的配置，包括datasource和sessionFactory



**7.搭建项目结构（分包）**

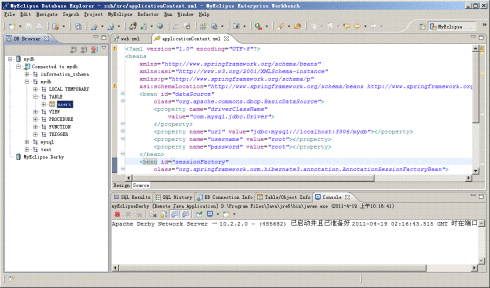
a)创建包结构：model（实体类）、dao（数据库访问类）、service（业务逻辑类）、web（控制类）



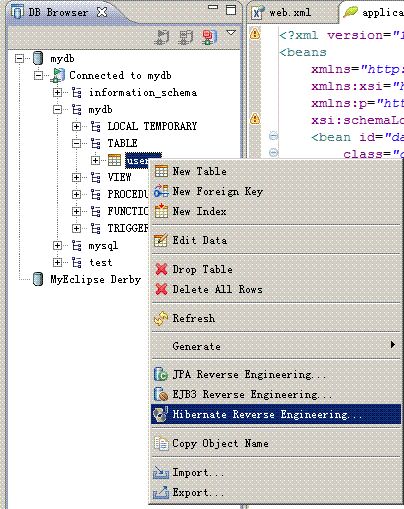
8.**搭建项目结构（model、dao生成）**

利用Myeclipse生成model和dao（最好自己构建可以减少冗余代码）

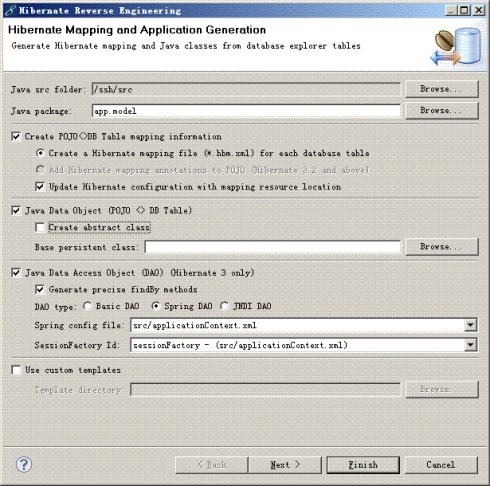
转到MyEclipse数据库浏览视图下，双击mydb进行连接



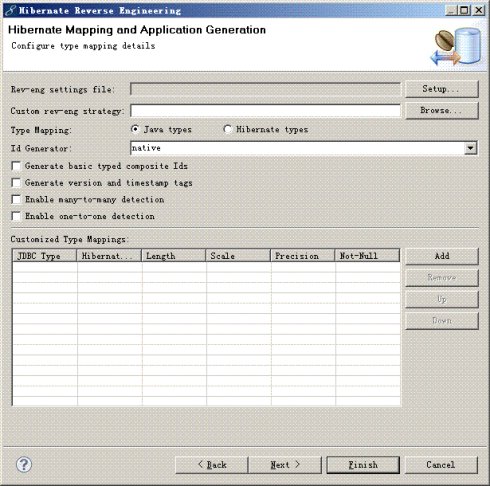
右键点击users表，选择Hibernate反向工程

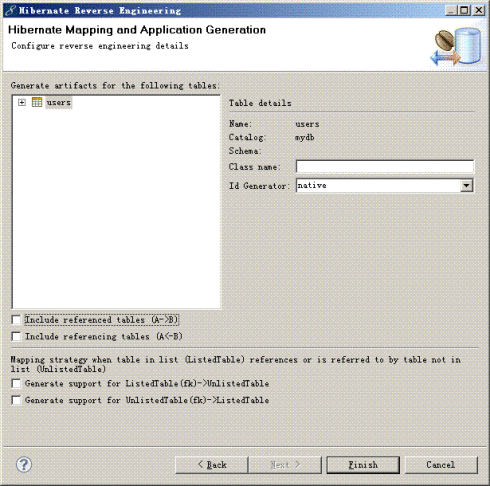


选择生成目录为app.model，勾选创建POJO即创建表对应的Javabean，同时创建映射文件hbm.xml，勾选更新hibernate配置，勾选创建spring dao。

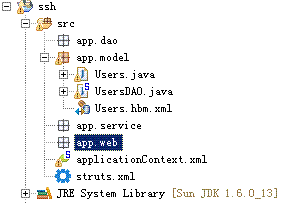


下一步，选择id生成策略为native，因为users主键自增，或者在下一步中选择也可以

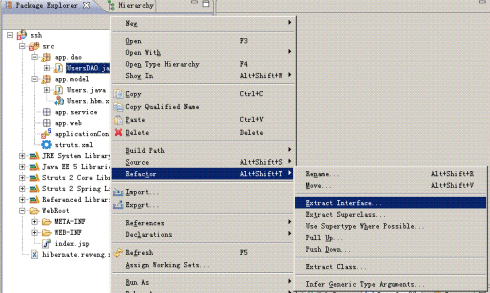


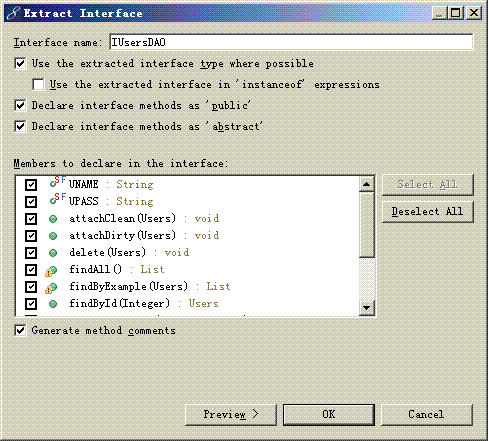


点击完成，回到代码视图查看工程变化，model包下面增加了Users.java、UsersDAO.java、Users.hbm.xml三个文件



将UsersDAO移动到dao包中，并抽出接口IUsersDAO





因为位置移动需要修改applicationContex.xml中的配置

<beanid=*"UsersDAO"*class=*"app.dao.UsersDAO"*>

<propertyname=*"sessionFactory"*>

<refbean=*"sessionFactory"*></ref>

</property>

</bean>

9.**搭建项目结构（service、web编写）**

a)编写service层代码，创建IUsersService接口和UsersService实现类，编写CRUD的方法。

**packageapp.service;**

**importjava.util.List;**

**importapp.model.Users;**

**public interface IUsersService {**

**public List<Users>findAll();**

**public List<Users>findByName(String uname);**

**public Users findByUid(Integer uid);**

**public void addUsers(Users u) throws Exception ;**

**public void updateUsers(Users u) throws Exception ;**

**public void delUsers(Integer uid) throws Exception ;**

**public Users login(String uname,Stringupass);**

**}**

**packageapp.service;**

**importjava.util.List;**

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

**importapp.dao.IUsersDAO;**

**importapp.dao.UsersDAO;**

**importapp.model.Users;**

**public class UsersService implements IUsersService {**

**private static final Logger log = LoggerFactory.getLogger(UsersService.class);**

**privateIUsersDAOusersDAO=null;**

**public void setUsersDAO(IUsersDAOusersDAO) {**

**this.usersDAO = usersDAO;**

**}**

**public void addUsers(Users u) throws Exception {**

**usersDAO.save(u);**

**}**

**public void delUsers (Integer uid) throws Exception {**

**usersDAO.delete(findByUid(uid));**

**}**

**public List<Users>findAll() {**

**returnusersDAO.findAll();**

**}**

**public List<Users>findByName(String uname) {**

**returnusersDAO.findByUname(uname);**

**}**

**public Users findByUid(Integer uid) {**

**returnusersDAO.findById(uid);**

**}**

**public Users login(String uname, String upass) {**

**List list = usersDAO.findByUname(uname);**

**if(null==list||list.size()==0)**

**return null;**

**Users u=(Users) usersDAO.findByUname(uname).get(0);**

**if(null==u)**

**return null;**

**if(uname.equals(u.getUname())&&upass.equals(u.getUpass()))**

**return u;**

**else**

**return null;**

**}**

**public void updateUsers (Users u) throws Exception {**

**usersDAO.attachDirty(u);**

**}**

**}**

b)编写web层UsersAction代码，用于对用户增删改查进行处理

**packageapp.web;**

**importjava.util.ArrayList;**

**importjava.util.List;**

**import org.slf4j.Logger;**

**import org.slf4j.LoggerFactory;**

**importorg.springframework.context.ApplicationContext;**

**importapp.model.Users;**

**importapp.service.IUsersService;**

**import com.opensymphony.xwork2.ActionContext;**

**import com.opensymphony.xwork2.ActionSupport;**

**public class UsersAction extends ActionSupport{**

**private static final Logger log = LoggerFactory.getLogger(UsersAction.class);**

**private Integer uid=new Integer(0);//用于修改删除时传参数**

**private Users u=new Users();//用于增改**

**private List<Users>userList=new ArrayList<Users>();//用于查询**

**privateIUsersServiceusersService=null;**

**public String addUser() throws Exception{**

**try {**

**usersService.addUsers(u);**

**} catch (Exception e) {**

**log.error("保存用户错误",e);**

**return ERROR;**

**}**

**return SUCCESS;**

**}**

**public String listUser() throws Exception{**

**userList=usersService.findAll();**

**return SUCCESS;**

**}**

**public String editUser() throws Exception{**

**try {**

**u=usersService.findByUid(uid);**

**} catch (Exception e) {**

**log.error("修改用户错误",e);**

**return ERROR;**

**}**

**return SUCCESS;**

**}**

**public String updateUser() throws Exception{**

**try {**

**usersService.updateUsers(u);**

**} catch (Exception e) {**

**log.error("更新用户错误",e);**

**return ERROR;**

**}**

**return SUCCESS;**

**}**

**public String delUser() throws Exception{**

**try {**

**usersService.delUsers(uid);**

**} catch (Exception e) {**

**log.error("删除用户错误",e);**

**return ERROR;**

**}**

**return SUCCESS;**

**}**

**public String login() throws Exception{**

**try {**

**Users loginUser=usersService.login(u.getUname(), u.getUpass());**

**if(null!=loginUser){**

**ActionContext context=ActionContext.getContext();**

**context.getSession().put("LOGINUSER",loginUser);**

**return SUCCESS;**

**}**

**} catch (Exception e) {**

**log.error("登录查询错误",e);**

**return ERROR;**

**}**

**return INPUT;**

**}**

**public void setUsersService(IUsersServiceusersService) {**

**this.usersService = usersService;**

**}**

**public Users getU() {**

**return u;**

**}**

**public void setU(Users u) {**

**this.u = u;**

**}**

**public List<Users>getUserList() {**

**returnuserList;**

**}**

**public void setUserList(List<Users>userList) {**

**this.userList = userList;**

**}**

**public Integer getUid() {**

**returnuid;**

**}**

**public void setUid(Integer uid) {**

**this.uid = uid;**

**}**

**}**

10.**搭建项目结构（配置文件编写）**

a)补充applicationContext.xml配置文件，装配注入关系和事务处理

增加spring配置文件节点的命名空间，否则事务处理节点不识别：

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<beansxmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:aop=*"http://www.springframework.org/schema/aop"*xmlns:tx=*"http://www.springframework.org/schema/tx"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd*

*http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

增加装配关系

<beanid=*"UsersDAO"*class=*"app.dao.UsersDAO"*>

<propertyname=*"sessionFactory"*>

<refbean=*"sessionFactory"*></ref>

</property>

</bean>

<beanid=*"UsersService"*class=*"app.service.UsersService"*>

<propertyname=*"usersDAO"*ref=*"UsersDAO"*/>

</bean>

<beanid=*"UsersAction"*class=*"app.web.UsersAction"*>

<propertyname=*"usersService"*ref=*"UsersService"*/>

</bean>

增加事务处理机制

<!--配置事务管理器-->

<beanid=*"transactionManager"* class=*"org.springframework.orm.hibernate3.HibernateTransactionManager"*>

<propertyname=*"sessionFactory"*>

<refbean=*"sessionFactory"*/>

</property>

</bean>

<!--配置事务的传播特性-->

<tx:adviceid=*"txAdvice"*transaction-manager=*"transactionManager"*>

<tx:attributes>

<tx:methodname=*"save\*"*propagation=*"REQUIRED"*/>

<tx:methodname=*"update\*"*propagation=*"REQUIRED"*/>

<tx:methodname=*"delete\*"*propagation=*"REQUIRED"*/>

<tx:methodname=*"add\*"*propagation=*"REQUIRED"*/>

<tx:methodname=*"del\*"*propagation=*"REQUIRED"*/>

<tx:methodname=*"\*"*read-only=*"true"*/>

</tx:attributes>

</tx:advice>

<!—哪些类的哪些方法参与事务-->

<aop:config>

<aop:pointcutid=*"allServiceMethod"*expression=*"execution(\* app.service.\*.\*(..))"*/>

<aop:advisorpointcut-ref=*"allServiceMethod"*advice-ref=*"txAdvice"*/>

</aop:config>

b)配置Struts.xml

<?xmlversion=*"1.0"*encoding=*"UTF-8"*?>

<!DOCTYPEstrutsPUBLIC"-//Apache Software Foundation//DTD Struts Configuration 2.1//EN""http://struts.apache.org/dtds/struts-2.1.dtd">

<struts>

<constantname=*"struts.objectFactory"*value=*"spring"*></constant>

<packagename=*"usermanage"*extends=*"struts-default"*>

<actionname=*"addUser"*class=*"UsersAction"*method=*"addUser"*>

<resultname=*"success"*type=*"redirect"*>listUser.action</result>

<resultname=*"error"*>/error.jsp</result>

</action>

<actionname=*"editUser"*class=*"UsersAction"*method=*"editUser"*>

<resultname=*"success"*>/user/user\_edit.jsp</result>

</action>

<actionname=*"updateUser"*class=*"UsersAction"*method=*"updateUser"*>

<resultname=*"success"*type=*"redirect"*>listUser.action</result>

<resultname=*"error"*>/error.jsp</result>

</action>

<actionname=*"delUser"*class=*"UsersAction"*method=*"delUser"*>

<resultname=*"success"*type=*"redirect"*>listUser.action</result>

<resultname=*"error"*>/error.jsp</result>

</action>

<actionname=*"listUser"*class=*"UsersAction"*method=*"listUser"*>

<resultname=*"success"*>/user/user\_list.jsp</result>

</action>

<actionname=*"login"*class=*"UsersAction"*method=*"login"*>

<resultname=*"success"*>/index.jsp</result>

<resultname=*"input"*>/login.jsp</result>

<resultname=*"error"*>/error.jsp</result>

</action>

</package>

</struts>

c)配置web.xml，设定applicationContex.xml的加载位置，如果放在WEB-INF下则不需要设定，配置监听。

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>

classpath:applicationContext.xml

</param-value>

</context-param>

<listener>

<listener-class>

org.springframework.web.context.ContextLoaderListener

</listener-class>

</listener>

11.**搭建项目结构（页面代码）**

a)页面关键代码

User\_add.jsp

<s:formaction=*"addUser"*>

<tableborder=*"1"*>

<tr><s:textfieldlabel=*"用户名"*name=*"u.uname"*></s:textfield></tr>

<tr><s:textfieldlabel=*"密码"*name=*"u.upass"*></s:textfield></tr>

<tr><tdcolspan=*"2"*><s:submitvalue=*"保存"*></s:submit></td></tr>

</table>

</s:form>

User\_edit.jsp

<s:formaction=*"updateUser"*>

<tableborder=*"1"*>

<s:hiddenname=*"u.uid"*></s:hidden>

<tr><s:textfieldlabel=*"用户名"*name=*"u.uname"*></s:textfield></tr>

<tr><s:textfieldlabel=*"密码"*name=*"u.upass"*></s:textfield></tr>

<tr><tdcolspan=*"2"*><s:submitvalue=*"保存"*></s:submit></td></tr>

</table>

</s:form>

User\_list.jsp

<tableborder=*"1"*>

<tr><td>编号</td><td>用户名</td><td>密码</td><td>操作</td></tr>

<s:iteratorvalue=*"userList"*>

<tr>

<td>${uid}</td>

<td>${uname }</td>

<td>${upass }</td>

<td><ahref=*'*<s:urlaction=*"editUser"*><s:paramname=*"uid"*value=*"uid"*/></s:url>*'*>修改</a>&nbsp;

<aonclick="return confirm('确定要删除吗？');"href=*'*<s:urlaction=*"delUser"*><s:paramname=*"uid"*value=*"uid"*/></s:url>*'*>删除</a></td>

</tr>

</s:iterator>

</table>

Login.jsp

<s:formaction=*"login"*>

<tableborder=*"1"*>

<tr><s:textfieldlabel=*"用户名"*name=*"u.uname"*></s:textfield></tr>

<tr><s:textfieldlabel=*"密码"*name=*"u.upass"*></s:textfield></tr>

<tr><tdcolspan=*"2"*><s:submitvalue=*"登录"*></s:submit></td></tr>

</table>

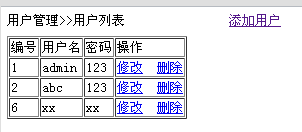
</s:form>

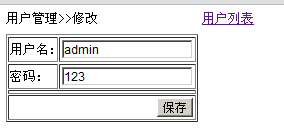
**12.测试运行**

[**http://localhost:8080/ssh/login.jsp**](http://localhost:8080/ssh/login.jsp)

****

**Struts2.1、Spring3.0、Hibernate3.3框架整合详细图解-12测试运行**

****

****

****

**1.异常排除**

**执行hql查询时发现异常java.lang.NoSuchMethodError: antlr.collections.AST.getLine()I**

**原因是由于strut2类库中包含antlr-2.7.2.jar而hibernate使用antlr-2.7.6.jar，版本冲突导致的，查看tomcat发布目录lib文件夹下发现同时存在这两个文件，删除antlr-2.7.2.jar即可**

**HTTP Status 500 -**

**type**Exception report

**message**

**description**The server encountered an internal error () that prevented it from fulfilling this request.

**exception**

java.lang.reflect.InvocationTargetException

sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)

sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)

java.lang.reflect.Method.invoke(Method.java:597)

com.opensymphony.xwork2.DefaultActionInvocation.invokeAction(DefaultActionInvocation.java:441)

com.opensymphony.xwork2.DefaultActionInvocation.invokeActionOnly(DefaultActionInvocation.java:280)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:243)

com.opensymphony.xwork2.interceptor.DefaultWorkflowInterceptor.doIntercept(DefaultWorkflowInterceptor.java:165)

com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.validator.ValidationInterceptor.doIntercept(ValidationInterceptor.java:252)

org.apache.struts2.interceptor.validation.AnnotationValidationInterceptor.doIntercept(AnnotationValidationInterceptor.java:68)

com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ConversionErrorInterceptor.intercept(ConversionErrorInterceptor.java:122)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ParametersInterceptor.doIntercept(ParametersInterceptor.java:195)

com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ParametersInterceptor.doIntercept(ParametersInterceptor.java:195)

com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.StaticParametersInterceptor.intercept(StaticParametersInterceptor.java:179)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.interceptor.MultiselectInterceptor.intercept(MultiselectInterceptor.java:75)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.interceptor.CheckboxInterceptor.intercept(CheckboxInterceptor.java:94)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.interceptor.FileUploadInterceptor.intercept(FileUploadInterceptor.java:235)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ModelDrivenInterceptor.intercept(ModelDrivenInterceptor.java:89)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ScopedModelDrivenInterceptor.intercept(ScopedModelDrivenInterceptor.java:130)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.interceptor.debugging.DebuggingInterceptor.intercept(DebuggingInterceptor.java:267)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ChainingInterceptor.intercept(ChainingInterceptor.java:126)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.PrepareInterceptor.doIntercept(PrepareInterceptor.java:138)

com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.I18nInterceptor.intercept(I18nInterceptor.java:165)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.interceptor.ServletConfigInterceptor.intercept(ServletConfigInterceptor.java:164)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.AliasInterceptor.intercept(AliasInterceptor.java:179)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ExceptionMappingInterceptor.intercept(ExceptionMappingInterceptor.java:176)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.impl.StrutsActionProxy.execute(StrutsActionProxy.java:52)

org.apache.struts2.dispatcher.Dispatcher.serviceAction(Dispatcher.java:488)

org.apache.struts2.dispatcher.ng.ExecuteOperations.executeAction(ExecuteOperations.java:77)

org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter.doFilter(StrutsPrepareAndExecuteFilter.java:91)

**root cause**

java.lang.NoSuchMethodError: antlr.collections.AST.getLine()I

org.hibernate.hql.ast.HqlSqlWalker.generatePositionalParameter(HqlSqlWalker.java:896)

org.hibernate.hql.antlr.HqlSqlBaseWalker.parameter(HqlSqlBaseWalker.java:4819)

org.hibernate.hql.antlr.HqlSqlBaseWalker.expr(HqlSqlBaseWalker.java:1373)

org.hibernate.hql.antlr.HqlSqlBaseWalker.exprOrSubquery(HqlSqlBaseWalker.java:4243)

org.hibernate.hql.antlr.HqlSqlBaseWalker.comparisonExpr(HqlSqlBaseWalker.java:3725)

org.hibernate.hql.antlr.HqlSqlBaseWalker.logicalExpr(HqlSqlBaseWalker.java:1864)

org.hibernate.hql.antlr.HqlSqlBaseWalker.whereClause(HqlSqlBaseWalker.java:818)

org.hibernate.hql.antlr.HqlSqlBaseWalker.query(HqlSqlBaseWalker.java:604)

org.hibernate.hql.antlr.HqlSqlBaseWalker.selectStatement(HqlSqlBaseWalker.java:288)

org.hibernate.hql.antlr.HqlSqlBaseWalker.statement(HqlSqlBaseWalker.java:231)

org.hibernate.hql.ast.QueryTranslatorImpl.analyze(QueryTranslatorImpl.java:254)

org.hibernate.hql.ast.QueryTranslatorImpl.doCompile(QueryTranslatorImpl.java:185)

org.hibernate.hql.ast.QueryTranslatorImpl.compile(QueryTranslatorImpl.java:136)

org.hibernate.engine.query.HQLQueryPlan.<init>(HQLQueryPlan.java:101)

org.hibernate.engine.query.HQLQueryPlan.<init>(HQLQueryPlan.java:80)

org.hibernate.engine.query.QueryPlanCache.getHQLQueryPlan(QueryPlanCache.java:94)

org.hibernate.impl.AbstractSessionImpl.getHQLQueryPlan(AbstractSessionImpl.java:156)

org.hibernate.impl.AbstractSessionImpl.createQuery(AbstractSessionImpl.java:135)

org.hibernate.impl.SessionImpl.createQuery(SessionImpl.java:1651)

org.springframework.orm.hibernate3.HibernateTemplate$30.doInHibernate(HibernateTemplate.java:914)

org.springframework.orm.hibernate3.HibernateTemplate$30.doInHibernate(HibernateTemplate.java:1)

org.springframework.orm.hibernate3.HibernateTemplate.doExecute(HibernateTemplate.java:406)

org.springframework.orm.hibernate3.HibernateTemplate.executeWithNativeSession(HibernateTemplate.java:374)

org.springframework.orm.hibernate3.HibernateTemplate.find(HibernateTemplate.java:912)

org.springframework.orm.hibernate3.HibernateTemplate.find(HibernateTemplate.java:908)

app.dao.UsersDAO.findByProperty(UsersDAO.java:98)

app.dao.UsersDAO.findByUname(UsersDAO.java:109)

app.service.UsersService.login(UsersService.java:35)

app.web.UsersAction.login(UsersAction.java:75)

sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)

sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)

java.lang.reflect.Method.invoke(Method.java:597)

com.opensymphony.xwork2.DefaultActionInvocation.invokeAction(DefaultActionInvocation.java:441)

com.opensymphony.xwork2.DefaultActionInvocation.invokeActionOnly(DefaultActionInvocation.java:280)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:243)

com.opensymphony.xwork2.interceptor.DefaultWorkflowInterceptor.doIntercept(DefaultWorkflowInterceptor.java:165)

com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.validator.ValidationInterceptor.doIntercept(ValidationInterceptor.java:252)

org.apache.struts2.interceptor.validation.AnnotationValidationInterceptor.doIntercept(AnnotationValidationInterceptor.java:68)

com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ConversionErrorInterceptor.intercept(ConversionErrorInterceptor.java:122)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ParametersInterceptor.doIntercept(ParametersInterceptor.java:195)

com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ParametersInterceptor.doIntercept(ParametersInterceptor.java:195)

com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.StaticParametersInterceptor.intercept(StaticParametersInterceptor.java:179)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.interceptor.MultiselectInterceptor.intercept(MultiselectInterceptor.java:75)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.interceptor.CheckboxInterceptor.intercept(CheckboxInterceptor.java:94)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.interceptor.FileUploadInterceptor.intercept(FileUploadInterceptor.java:235)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ModelDrivenInterceptor.intercept(ModelDrivenInterceptor.java:89)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ScopedModelDrivenInterceptor.intercept(ScopedModelDrivenInterceptor.java:130)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.interceptor.debugging.DebuggingInterceptor.intercept(DebuggingInterceptor.java:267)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ChainingInterceptor.intercept(ChainingInterceptor.java:126)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.PrepareInterceptor.doIntercept(PrepareInterceptor.java:138)

com.opensymphony.xwork2.interceptor.MethodFilterInterceptor.intercept(MethodFilterInterceptor.java:87)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.I18nInterceptor.intercept(I18nInterceptor.java:165)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.interceptor.ServletConfigInterceptor.intercept(ServletConfigInterceptor.java:164)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.AliasInterceptor.intercept(AliasInterceptor.java:179)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

com.opensymphony.xwork2.interceptor.ExceptionMappingInterceptor.intercept(ExceptionMappingInterceptor.java:176)

com.opensymphony.xwork2.DefaultActionInvocation.invoke(DefaultActionInvocation.java:237)

org.apache.struts2.impl.StrutsActionProxy.execute(StrutsActionProxy.java:52)

org.apache.struts2.dispatcher.Dispatcher.serviceAction(Dispatcher.java:488)

org.apache.struts2.dispatcher.ng.ExecuteOperations.executeAction(ExecuteOperations.java:77)

org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter.doFilter(StrutsPrepareAndExecuteFilter.java:91)

**note**The full stack trace of the root cause is available in the Apache Tomcat/6.0.13 logs.