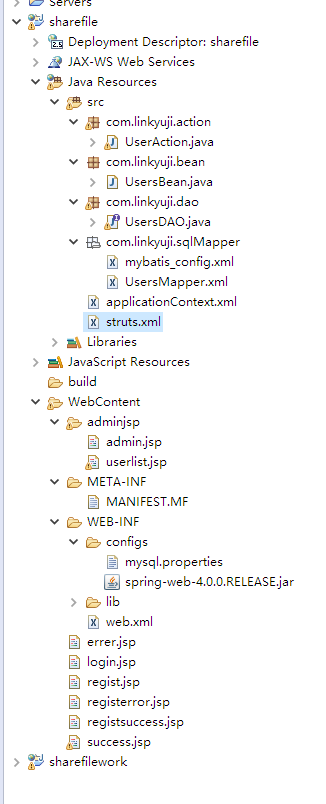
**ssm（spring Struts2 mybatis）程序开发笔记**

**总览：**



**配置文件相关**

写在前面：

spring的配置文件中，所有的变量名开头必须小写，不然可能会出现问题。

***配置文件：applicationContext.xml***

导入properties文件

<!--配置数据源属性文件 -->

<bean id=*"propertyConfigurer"*

class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*>

<property name=*"location"*>

<value>/WEB-INF/configs/mysql.properties</value>

</property>

</bean>

该配置是一个全局配置，之后直接进行调用就可以了

调用方式

${属性名}

像这样

<!--配置数据源 -->

<bean id=*"dataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"*>

<value>${jdbc.driver}</value>

</property>

<property name=*"url"*>

<value>${jdbc.url}</value>

</property>

<property name=*"username"*>

<value>${jdbc.user}</value>

</property>

<property name=*"password"*>

<value>${jdbc.pwd}</value>

</property>

</bean>

上面这段配置文件是对数据源的配置，其中id这个属性名是为了后面的调用，下文中出现的id的功能都是相同的。

sqlsessionFactory

<bean id=*"sqlSessionFactory"* class=*"org.mybatis.spring.SqlSessionFactoryBean"*>

<property name=*"configLocation"* value=*"classpath:com/linkyuji/sqlMapper/mybatis\_config.xml"* />

<property name=*"dataSource"* ref=*"dataSource"* />

</bean>

这个配置的作用是引入前面所提到的数据源，并且将mybatis的配置文件导入，让mybatis能够使用到数据源。

userDao

<bean id=*"usersDao"* class=*"org.mybatis.spring.mapper.MapperFactoryBean"*>

<property name=*"mapperInterface"* value=*"com.linkyuji.dao.UsersDAO"*/>

<property name=*"sqlSessionFactory"* ref=*"sqlSessionFactory"* />

</bean>

这里是配置Dao类，其中mapperInterface这个属性为了告诉mybatis引用的是一个接口，sqlSessionFactory这个属性是一个工厂类，引用上文所创建的sqlSessionFactory来生成实例。

Action

<bean id=*"userAction"* class=*"com.linkyuji.action.UserAction"*>

<property name=*"usersDao"* ref=*"usersDao"*></property>

</bean>

这里的id的属性内容将会在Struts2的配置文件里引用到，class属性内容为我们编写的.java文件的具体路径。

property标签内的name属性与我们所引用的.java文件中的一个变量对应，请务必保持一致，并且创建get set方法。ref属性为上文的userAction的id。

相关片段：

userAction：

**public** **class** UserAction {

**private** UsersDAO usersDao;

**private** UsersBean user;

**public** UsersDAO getUsersDao() {

**return** usersDao;

}

**public** **void** setUsersDao(UsersDAO usersDao) {

**this**.usersDao = usersDao;

}

**public** UsersBean getUser() {

**return** user;

}

**public** **void** setUser(UsersBean user) {

**this**.user = user;

}

/\*相关功能模块\*/

}

Struts2：

<struts>

<package name=*"myPackage"* extends=*"struts-default"*>

<action name=*"user\_\*"* class=*"userAction"* method=*"{1}"*>

<result name=*"success"*>/success.jsp</result>

<result name=*"error"*>/errer.jsp</result>

<result name=*"regsuccess"*>/registsuccess.jsp</result>

<result name=*"regerror"*>/registerror.jsp</result>

<result name=*"list"*>/adminjsp/userlist.jsp</result>

</action>

</package>

</struts>

mybatis相关配置文件

<configuration>

<typeAliases>

<typeAlias alias=*"userbean"* type=*"com.linkyuji.bean.UsersBean"*/>

</typeAliases>

<mappers>

<mapper resource=*"com/linkyuji/sqlMapper/UsersMapper.xml"*/>

</mappers>

</configuration>

这里的typeAliases标签用来引用bean文件，可以在之后的配置文件里使用到

mapper是为了引用映射文件

映射文件：

<mapper namespace=*"com.linkyuji.dao.UsersDAO"*>

<select id=*"getUserByid"* parameterType=*"String"*

resultType=*"userbean"*>

select \* from `users` where userid = #{id};

</select>

<insert id=*"regist"* parameterType=*"userbean"*>

INSERT INTO

`sharefile`.`users` (`userid`, `userpsw`, `username`,`usertype`)

VALUES (#{userid},#{userpsw}, #{userid},'2');

</insert>

<select id=*"getalluser"* resultType=*"userbean"*>

select \* from `users` where

usertype = 2;

</select>

<update id=*"editoruser"* parameterType=*"userbean"*>

UPDATE `sharefile`.`users`

SET `userpsw`=#{userpsw}, `username`=#{username} WHERE

`userid`=#{userid};

</update>

<delete id=*"deleteuser"* parameterType=*"userbean"*>

DELETE FROM `sharefile`.`users` WHERE `userid`=#{userid};

</delete>

</mapper>

namespace属性是DAO的类型，下面的都是一些SQL语句操作

标签的类型

select查询

insert增加

update更新

delete删除

id属性是对应DAO的方法名，parameter是输入属性类型 resultTyoe是返回属性类型（这里可以使用mybatis配置文件中配置好的typeAlias属性）

在sql语句中 #{属性名}指的是输入属性的变量名，比如代码证#{userid}就是UsersBean的一个变量。

tips：

如果需要通过sql查询出一个list<属性名>，在返回类型属性中填写的是“属性名”而不是list<属性名>,至于接受只要在DAO那设置成list就可以了

比如

<select id=*"getalluser"* resultType=*"com.linkyuji.bean.UsersBean"*>

select \* from `users` where usertype = 2;

</select>

对应的DAO

**public** List getalluser();

**有关文件**

创建一个文件夹时，如果你的路径是：

File file = **new** File("F:\\Share\\User\\"+id);

那么在创建时，不能够使用

file.mkdir();

而是用

file.mkdirs();

**有关jsp**

页面跳转

<meta http-equiv=*"refresh"* content=*"1;url=userjsp/user.jsp"*>

content里的第一个属性代表时间（单位是 秒）

<a href=*"javascript:history.back(-1)"*>上一级</a>

这段代码实现的是返回到上一个界面，功能相当于浏览器的后退按钮。

**其他非常关键的东西**

当准备使用session来传递int类型时，在get这个session的值时应该这样做：

**Integer.parseInt**(session.getAttribute("oldp").toString());

非常关键！因为这里不能使用强制转换，不能使用valueof();

**Spring有个BUG**

当创建<bean>的时候，他只会认文件刚刚创建时的名字，当文件名字改变后，即使你在bean里去修改成已经改变的名字，依旧是无法进行正确配置的，解决方式是将原文件删除后重新创建！