

SSH综合应用

Struts + Spring + Hibernate

李兴华

JAVA培训报名热线: 010-51283346

.IT.电.子.教.育.门.户.

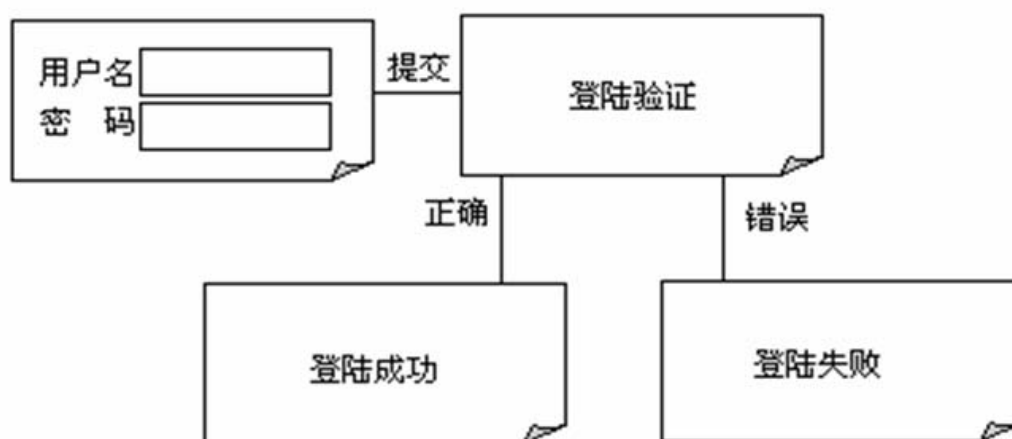
© MLDN INC.

更多学习资源尽在

www.MLDN.cn

本章目标

- ◆ 使用Struts、Spring、Hibernate完成一个用户登陆程序



JAVA培训报名热线：010-51283346

更多学习资源尽在 **MLDN.cn**

-- 删除表

```
DROP TABLE person ;
```

-- 创建表

```
CREATE TABLE person
```

```
(
```

```
    id varchar(32) not null primary key ,
```

```
    name varchar(20) ,
```

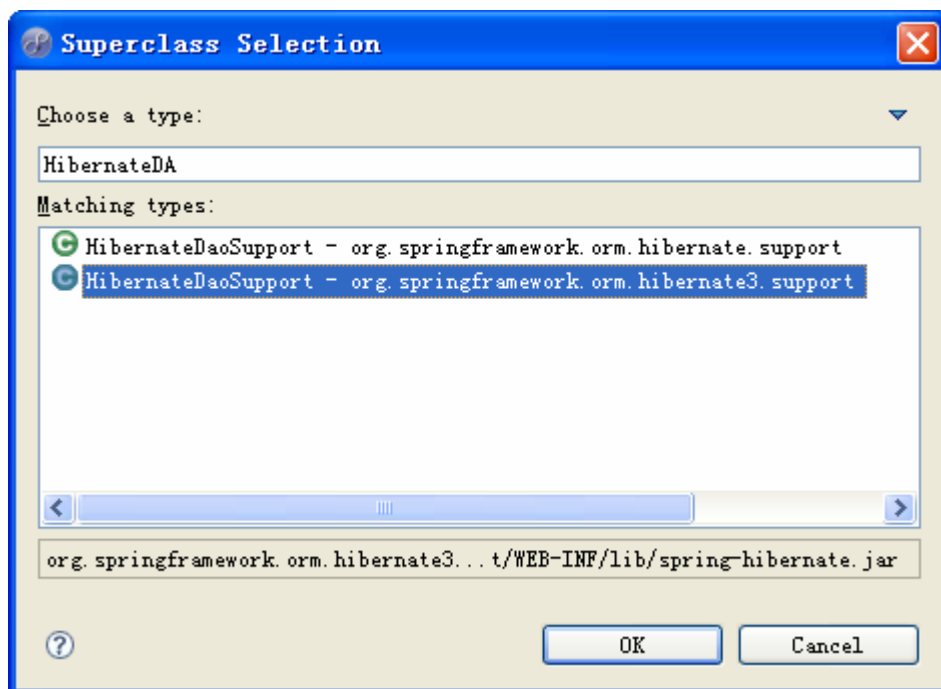
```
    password varchar(20)
```

```
);
```

-- 插入测试数据

```
INSERT INTO person (id,name,password) values ('lxh','李兴华','lixinghua');
```

```
INSERT INTO person (id,name,password) values ('mldn','mole','admin');
```



```
package org.lxh.sshlogin.impl;

import java.util.List;

import org.hibernate.Query;
import org.lxh.sshlogin.dao.PersonDAO;
import org.lxh.sshlogin.vo.Person;
import org.springframework.orm.hibernate3.support.HibernateDaoSupport;

public class PersonDAOImpl extends HibernateDaoSupport implements PersonDAO {
    public boolean isLogin(Person person) throws Exception {
        boolean flag = false;
        String hql = "FROM Person AS p WHERE p.id=? AND p.passwr=?";
        Query q = this.getSession().createQuery(hql);
        q.setString(0, person.getId());
        q.setString(1, person.getPassword());
        List all = q.list();
        if (all.size() > 0) {
            flag = true;
            person.setName(((Person) all.get(0)).getName());
        }
        return flag;
    }
}
```

Spring 框架包的问题是非常严重 → 重新建立项目，重新加入。将 spring 包下的全部文件拷贝过来。

struts-config.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts
Configuration 1.2//EN" "http://struts.apache.org/dtds/struts-config_1_2.dtd">
<struts-config>
  <data-sources />
  <form-beans>
    <form-bean name="personForm"
      type="org.apache.struts.validator.DynaValidatorForm">
      <form-property name="password" type="java.lang.String" />
      <form-property name="id" type="java.lang.String" />
    </form-bean>
  </form-beans>
  <global-exceptions />
  <global-forwards />
  <action-mappings>
    <action attribute="personForm" input="/login.jsp"
      name="personForm" parameter="status" path="/person" scope="request"
      type="org.lxx.sshlogin.struts.action.PersonAction" validate="true">
      <forward name="suc" path="/login.jsp"></forward>
    </action>
  </action-mappings>

  <message-resources
    parameter="org.lxx.sshlogin.struts.ApplicationResources" />
  <plug-in className="org.apache.struts.validator.ValidatorPlugIn">
    <set-property property="pathnames"
      value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml" />
  </plug-in>
  <plug-in
    className="org.springframework.web.struts.ContextLoaderPlugIn">
    <set-property property="contextConfigLocation"
      value="/WEB-INF/classes/applicationContext.xml" />
  </plug-in>
</struts-config>
```

在web.xml中配置spring

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>
        /WEB-INF/classes/applicationContext.xml
    </param-value>
</context-param>
<servlet>
    <servlet-name>context</servlet-name>
    <servlet-class>
        org.springframework.web.context.ContextLoaderServlet
    </servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
```

或者：org.springframework.web.context.ContextLoaderListener

JAVA培训报名热线：010-51283346

更多学习资源尽在 **MLDN.cn**

两个最重要的 XML 配置文件：

struts-config.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts-config PUBLIC "-//Apache Software Foundation//DTD Struts
Configuration 1.2//EN" "http://struts.apache.org/dtds/struts-config_1_2.dtd">
<struts-config>
    <data-sources />
    <form-beans>
        <form-bean name="personForm"
            type="org.apache.struts.validator.DynaValidatorForm">
            <form-property name="password" type="java.lang.String" />
            <form-property name="id" type="java.lang.String" />
        </form-bean>
    </form-beans>
    <global-exceptions />
    <global-forwards />
    <action-mappings>
        <action attribute="personForm" input="/login.jsp"
            name="personForm" parameter="status" path="/person" scope="request">
```

```

        type="org.lxx.sshlogin.struts.action.PersonAction" validate="true">
        <forward name="suc" path="/login.jsp"></forward>
    </action>
</action-mappings>
<controller
    processorClass="org.springframework.web.struts.DelegatingRequestProcessor">
</controller>
<message-resources
    parameter="org.lxx.sshlogin.struts.ApplicationResources" />
<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
    <set-property property="pathnames"
        value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml" />
</plug-in>
<plug-in
    className="org.springframework.web.struts.ContextLoaderPlugIn">
    <set-property property="contextConfigLocation"
        value="/WEB-INF/classes/applicationContext.xml" />
</plug-in>
</struts-config>

```

applicationContext.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">

<beans>
    <bean id="dataSource"
        class="org.apache.commons.dbcp.BasicDataSource"
        destroy-method="close">
        <property name="driverClassName">
            <value>org.gjt.mm.mysql.Driver</value>
        </property>
        <property name="url">
            <value>jdbc:mysql://localhost:3306/mln</value>
        </property>
        <property name="username">
            <value>root</value>
        </property>
        <property name="password">
            <value>mysqladmin</value>
        </property>
    </bean>

```

```

</bean>
<bean id="sessionFactory"
    class="org.springframework.orm.hibernate3.LocalSessionFactoryBean"
    destroy-method="close">
    <property name="dataSource">
        <ref bean="dataSource" />
    </property>
    <property name="hibernateProperties">
        <props>
            <prop key="hibernate.dialect">
                org.hibernate.dialect.MySQLDialect
            </prop>
        </props>
    </property>
    <property name="mappingResources">
        <list>
            <value>org/lxh/sshlogin/vo/Person.hbm.xml</value>
        </list>
    </property>
</bean>
<bean id="hibernateTemplate"
    class="org.springframework.orm.hibernate3.HibernateTemplate">
    <property name="sessionFactory">
        <ref bean="sessionFactory" />
    </property>
</bean>
<bean id="persondao" class="org.lxh.sshlogin.dao.PersonDAO"
    abstract="true">
</bean>
<bean id="persondaoimpl" class="org.lxh.sshlogin.impl.PersonDAOImpl"
    parent="persondao">
    <property name="hibernateTemplate">
        <ref bean="hibernateTemplate" />
    </property>
</bean>
<bean name="/person"
    class="org.lxh.sshlogin.struts.action.PersonAction">
    <property name="personDao">
        <ref bean="persondaoimpl" />
    </property>
</bean>
</beans>

```

结论：SSH 技术基本上提供的就是一个大的架构模型，用户按照此模型构建自己的代码
配置文件写的非常的多，完全是基于配置完成。

在讲解中实际上讲解的主要是 struts 与 spring 直接联系起来
不用去添加 controller，而直接使用 spring 中已经提供好的一个类：



修改struts-config.xml

```
<action attribute="personForm" input="/login.jsp"
        name="personForm" path="/person" scope="request"
        type="org.springframework.web.struts.DelegatingActionProxy"
        validate="true" parameter="status">
    <forward name="person" path="/login.jsp" />
</action>
```

JAVA培训报名热线：010-51283346

更多学习资源尽在
www.MLDN.cn

结合了以下技术：

- 1、 Spring
结合了 HibernateTemplate、HibernateDaoSupport 类
Spring 中的 Ioc 机制，及数据库连接技术
- 2、 Hibernate
只完成调用 HQL 查询的操作
- 3、 Struts
验证框架
分发 Action
动态验证 ActionForm