整合步骤流程：

1. 配置web.xml
2. 配置Spring监听器，使得项目加载Spring

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

1. 添加对struts2的支持

<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>

1. 配置Session延迟加载到页面

<filter>

<filter-name>openSessionInViewFilter</filter-name>

<filter-class>org.springframework.orm.hibernate4.support.OpenSessionInViewFilter

</filter-class>

<init-param>

<param-name>singleSession</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>openSessionInViewFilter</filter-name>

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

</filter-mapping>

1. 配置hibernate.cfg.xml文件

<session-factory>

<!--方言-->

<property name="dialect">org.hibernate.dialect.MySQL5Dialect</property>

<!-- 显示sql语句 -->

<property name="show\_sql">true</property>

<!-- 自动更新 -->

<property name="hbm2ddl.auto">update</property>

</session-factory>

1. 配置Spring的配置整合文件，applicationContext.xml
2. 配置定义数据源-- 定义数据源 -->

<bean id="dataSource"

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

<property name="driverClassName"

value="com.mysql.jdbc.Driver">

</property>

<property name="url"

value="jdbc:mysql://localhost:3306/test">

</property>

<property name="username" value="root"></property>

<property name="password" value="123456"></property>

</bean>

1. 配置session工厂

<bean id="sessionFactory"

class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">

<property name="dataSource">

<ref bean="dataSource" />

</property>

<!-- 通过配置该class的属性自动加载hibernate配置文件 -->

<property name="configLocation" value="classpath:hibernate.cfg.xml"/>

<!-- 自动扫描注解方式将<value>指定包下的带有注解的实体映射到数据库 -->

<property name="packagesToScan">

<list>

<value>com.zhou.entity</value>

</list>

</property>

</bean>

1. 配置事务管理

<bean id="transactionManager"

class="org.springframework.orm.hibernate4.HibernateTransactionManager">

<property name="sessionFactory" ref="sessionFactory" />

</bean>

1. 配置事务通知属性

<tx:advice id="txAdvice" transaction-manager="transactionManager">

<!-- 定义事务传播属性 -->

<tx:attributes>

<tx:method name="insert\*" propagation="REQUIRED" />

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

<tx:method name="edit\*" propagation="REQUIRED" />

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

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

<tx:method name="new\*" propagation="REQUIRED" />

<tx:method name="set\*" propagation="REQUIRED" />

<tx:method name="remove\*" propagation="REQUIRED" />

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

<tx:method name="change\*" propagation="REQUIRED" />

<tx:method name="get\*" propagation="REQUIRED" read-only="true" />

<tx:method name="find\*" propagation="REQUIRED" read-only="true" />

<tx:method name="load\*" propagation="REQUIRED" read-only="true" />

<tx:method name="\*" propagation="REQUIRED" read-only="true" />

</tx:attributes>

</tx:advice>

1. 配置事务切面

<aop:config>

<aop:pointcut id="serviceOperation"

expression="execution(\* com.zhou.service..\*.\*(..))" />

<aop:advisor advice-ref="txAdvice" pointcut-ref="serviceOperation" />

</aop:config>

1. 声明自动构建bean

<!-- 自动加载构建bean base-package指定的包及其子包下的所有 class文件，

所有标注了 @Component,@Service,@Controller,@Repository 注解的 类 纳入spring容器管理

在类上 ，使用以下注解，实现bean 的声明

@Component 泛指组件，当组件不好归类的时候，我们可以使用这个注解进行标注。

@Service 用于标注业务层组件

@Controller 用于标注控制层组件（如srping mvc的controller,struts中的action）

@Repository 用于标注数据访问组件，即DAO组件-->

<context:component-scan base-package="com.zhou" />

1. 编写接口dao,以及实现dao接口的实现类，并dao接口实现类中声明注解，使用@Repository标注，在方法上使用标注@Autowired，会自动查找被标注的方法的参数类型的bean，并调用方法自动注入这些bean
2. 编写service接口，以及声明自己的业务逻辑，实现service接口类，具体实现自己的业务逻辑使用@Service标注实现bean声明，使用@Resource标注声明要使用的BaseDao bean

@Autowired默认按类型装配（这个注解是属业spring的），默认情况下必须要求依赖对象必须存在@Resource（这个注解属于J2EE的），默认安照名称进行装配，名称可以通过name属性进行指定，   
如果没有指定name属性，当注解写在字段上时，默认取字段名进行按照名称查找，如果注解写在setter方法上默认取属性名进行装配。 当找不到与名称匹配的bean时才按照类型进行装配。但是需要注意的是，如果name属性一旦指定，就只会按照名称进行装配。

1. 编写action也就是controller层，使用标注@Controller，使用@Resource标注声明要使用的UserServicer bean
2. 配置strut.xml文件，就是配置自己的action

<package name="s2sh" namespace="/user" extends="struts-default">

<action name="user\_\*" method="{1}" class="com.zhou.action.UserAction">

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

<result name="error">/index.jsp</result>

</action>

</package>

1. 最后编写jsp，用户界面

<form action="${pageContext.request.contextPath }/user/user\_login.action" method="post">

userName:<input type="text" name="user.userName" value="${user.userName}"/><br/>

password:<input type="password" name="user.password" value="${user.password}"/><br/>

<input type="submit" value="login"/><font color="red">${error}</font>

</form>