**1.基本配置：**

<?xml version="1.0" encoding="UTF-8"?>

<beans

xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-2.5.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-2.5.xsd

">

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

<!-- 开启组件扫描 -->

</context:component-scan>

<context:annotation-config>

<!--开启注解处理器-->

</context:annotation-config>

<!-- 使用注解,省去了propertity的xml配置，减少xml文件大小 -->

<bean id="personServiceAnno" class="com.persia.PersonServiceAnnotation"></bean>

<bean id="personDaoBeanAnno" class="com.persia.PersonDaoBean"></bean>

<bean id="personDaoBeanAnno2" class="com.persia.PersonDaoBean"></bean>

<!-- 自动注解 -->

<bean id="personServiceAutoInject" class="com.persia.PersonServiceAutoInject" autowire="byName"></bean>

<bean id="personService" class="com.persia.PersonServiceBean">

<!-- 由spring容器去创建和维护，我们只要获取就可以了 -->

</bean>

<bean id="personService2" class="com.persia.PersonServiceBeanFactory" factory-method="createInstance" lazy-init="true"

init-method="init" destroy-method="destory">

<!-- 静态工厂获取bean -->

</bean>

<bean id="fac" class="com.persia.PersonServiceBeanInsFactory"></bean>

<bean id="personService3" factory-bean="fac" factory-method="createInstance" scope="prototype">

<!-- 实例工厂获取bean，先实例化工厂再实例化bean-->

</bean>

<!-- ref方式注入属性 -->

<bean id="personDao" class="com.persia.PersonDaoBean"></bean>

<bean id="personService4" class="com.persia.PersonServiceBean">

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

</bean>

<!-- 内部bean方式注入 -->

<bean id="personService5" class="com.persia.PersonServiceBean">

<property name="personDao">

<bean class="com.persia.PersonDaoBean"></bean>

</property>

<property name="name" value="persia"></property>

<property name="age" value="21"></property>

<property name="sets">

<!-- 集合的注入 -->

<set>

<value>第一个</value>

<value>第二个</value>

<value>第三个</value>

</set>

</property>

<property name="lists">

<!-- 集合的注入 -->

<list>

<value>第一个l</value>

<value>第二个l</value>

<value>第三个l</value>

</list>

</property>

<property name="properties">

<props>

<prop key="key1">value1</prop>

<prop key="key2">value2</prop>

<prop key="key3">value3</prop>

</props>

</property>

<property name="map">

<map>

<entry key="key1" value="value-1"></entry>

<entry key="key2" value="value-2"></entry>

<entry key="key3" value="value-3"></entry>

</map>

</property>

</bean>

<bean id="personService6" class="com.persia.PersonServiceBean">

<constructor-arg index="0" value="构造注入的name" ></constructor-arg>

<!-- 基本类型可以不写type -->

<constructor-arg index="1" type="com.persia.IDaoBean" ref="personDao">

</constructor-arg>

</bean>

</beans>

**2.开启AOP：**

<?xml version="1.0" encoding="UTF-8"?>

<beans

xmlns="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"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-2.5.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-2.5.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-2.5.xsd

">

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<bean id="myInterceptor" class="com.persia.service.MyInterceptor"></bean>

<bean id="personServiceImpl" class="com.persia.service.impl.PersonServiceImpl"></bean>

</beans>

AOP的xml版本

<?xml version="1.0" encoding="UTF-8"?>

<beans

xmlns="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"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-2.5.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-2.5.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-2.5.xsd

">

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<bean id="personService" class="com.persia.service.impl.PersonServiceImpl"></bean>

<bean id="aspectBean" class="com.persia.service.MyInterceptor"></bean>

<aop:config>

<aop:aspect id="myaop" ref="aspectBean">

<aop:pointcut id="mycut" expression="execution(\* com.persia.service.impl.PersonServiceImpl.\*(..))"/>

<aop:pointcut id="argcut" expression="execution(\* com.persia.service.impl.PersonServiceImpl.\*(..)) and args(name)"/>

<aop:before pointcut-ref="mycut" method="doAccessCheck" />

<aop:after-returning pointcut-ref="mycut" method="doAfterReturning"/>

<aop:after-throwing pointcut-ref="mycut" method="doThrowing"/>

<aop:after pointcut-ref="argcut" method="doAfter" arg-names="name"/>

<aop:around pointcut-ref="mycut" method="arround"/>

</aop:aspect>

</aop:config>

</beans>

**3.开启事务和注解：**

<?xml version="1.0" encoding="UTF-8"?>

<beans

xmlns="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-2.5.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-2.5.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-2.5.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.5.xsd

">

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<!-- 配置数据源 -->

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">

<property name="driverClassName" value="com.mysql.jdbc.Driver"/>

<property name="url" value="jdbc:mysql://localhost:3306/test?useUnicode=true&amp;characterEncoding=utf-8"/>

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

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

<!-- 连接池启动时的初始值 -->

<property name="initialSize" value="1"/>

<!-- 连接池的最大值 -->

<property name="maxActive" value="500"/>

<!-- 最大空闲值.当经过一个高峰时间后，连接池可以慢慢将已经用不到的连接慢慢释放一部分，一直减少到maxIdle为止 -->

<property name="maxIdle" value="2"/>

<!-- 最小空闲值.当空闲的连接数少于阀值时，连接池就会预申请去一些连接，以免洪峰来时来不及申请 -->

<property name="minIdle" value="1"/>

</bean>

<!-- 配置事务管理器-->

<bean id="txManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">

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

</bean>

<!-- 配置业务bean -->

<bean id="personService" class="com.persia.service.impl.PersonServiceImpl">

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

</bean>

<!-- 采用@Transactional注解方式来使用事务 -->

<tx:annotation-driven transaction-manager="txManager"/>

</beans>

XML版本：

<?xml version="1.0" encoding="UTF-8"?>

<beans

xmlns="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-2.5.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-2.5.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-2.5.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.5.xsd

">

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<!-- 配置数据源 -->

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">

<property name="driverClassName" value="com.mysql.jdbc.Driver"/>

<property name="url" value="jdbc:mysql://localhost:3306/test?useUnicode=true&amp;characterEncoding=utf-8"/>

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

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

<!-- 连接池启动时的初始值 -->

<property name="initialSize" value="1"/>

<!-- 连接池的最大值 -->

<property name="maxActive" value="500"/>

<!-- 最大空闲值.当经过一个高峰时间后，连接池可以慢慢将已经用不到的连接慢慢释放一部分，一直减少到maxIdle为止 -->

<property name="maxIdle" value="2"/>

<!-- 最小空闲值.当空闲的连接数少于阀值时，连接池就会预申请去一些连接，以免洪峰来时来不及申请 -->

<property name="minIdle" value="1"/>

</bean>

<!-- 配置事务管理器 -->

<bean id="txManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">

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

</bean>

<!-- 配置业务bean -->

<bean id="personService" class="com.persia.service.impl.PersonServiceImpl">

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

</bean>

<!-- 使用XML来使用事务管理-->

<aop:config>

<!-- 配置一个切面，和需要拦截的类和方法 -->

<aop:pointcut id="transactionPointcut" expression="execution(\* com.persia.service..\*.\*(..))"/>

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

</aop:config>

<!-- 配置一个事务通知 -->

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

<tx:attributes>

<!-- 方法以get开头的，不使用事务 -->

<tx:method name="get\*" read-only="true" propagation="NOT\_SUPPORTED"/>

<!-- 其他方法以默认事务进行 -->

<tx:method name="\*"/>

</tx:attributes>

</tx:advice>

</beans>

**4.SSH:**

<?xml version="1.0" encoding="UTF-8"?>

<beans

xmlns="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-2.5.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-2.5.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-2.5.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.5.xsd

">

<!-- 配置数据源 -->

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">

<property name="driverClassName" value="com.mysql.jdbc.Driver"/>

<property name="url" value="jdbc:mysql://localhost:3306/test?useUnicode=true&amp;characterEncoding=utf-8"/>

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

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

<!-- 连接池启动时的初始值 -->

<property name="initialSize" value="1"/>

<!-- 连接池的最大值 -->

<property name="maxActive" value="500"/>

<!-- 最大空闲值.当经过一个高峰时间后，连接池可以慢慢将已经用不到的连接慢慢释放一部分，一直减少到maxIdle为止 -->

<property name="maxIdle" value="2"/>

<!-- 最小空闲值.当空闲的连接数少于阀值时，连接池就会预申请去一些连接，以免洪峰来时来不及申请 -->

<property name="minIdle" value="1"/>

</bean>

<!-- 配置hibernate的sessionFactory -->

<bean id="sessionFactory" class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">

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

<property name="mappingResources">

<list>

<value>com/persia/model/Person.hbm.xml</value>

</list>

</property>

<!-- 1.首先在sessionFactory里面配置以上3条设置 -->

<!-- 2.然后得在类路径下面添加一个ehcache.xml的缓存配置文件 -->

<!-- 3.最后在要使用缓存的实体bean的映射文件里面配置缓存设置 -->

<!--使用二级缓存-->

<!-- 不使用查询缓存，因为命中率不是很高 -->

<!-- 使用Ehcache缓存产品 -->

<property name="hibernateProperties">

<value>

hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

hibernate.hbm2ddl.auto=update

hibernate.show\_sql=false

hibernate.format\_sql=false

hibernate.cache.use\_second\_level\_cache=true

hibernate.cache.use\_query\_cache=false

hibernate.cache.provider\_class=org.hibernate.cache.EhCacheProvider

</value>

</property>

</bean>

<!-- 配置Spring针对hibernate的事务管理器 -->

<bean id="txManager" class="org.springframework.orm.hibernate3.HibernateTransactionManager">

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

</bean>

<!-- 配置使用注解的方式来使用事务 -->

<tx:annotation-driven transaction-manager="txManager"/>

<!-- 使用手工配置的注解方式来注入bean -->

<context:annotation-config></context:annotation-config>

<!--定义要注入的业务bean -->

<bean id="personService" class="com.persia.service.impl.PersonServiceImpl"></bean>

<!--将Struts的action交给Spring容器来管理 -->

<bean name="/person/list" class="com.persia.struts.PersonListAction">

<!--1.这里要求name和struts-config里面的action的path名称一致，因为id不允许有特殊字符-->

<!--2.还得在Struts-config文件里面添加Spring的请求处理器，该处理器会根据action的path属性到Spring容器里面寻找这个bean，若找到了则用这个bean来处理用户的请求-->

<!--3.然后去掉action的type标签和值（可选），当Spring处理器找不到该bean时，才会使用Struts的action-->

<!--4.最后在action里面使用Spring的注入方式来注入业务bean-->

</bean>

<bean name="/person/manage" class="com.persia.struts.PersonManageAction"></bean>

</beans>

**5.SSH2:**

<?xml version="1.0" encoding="UTF-8"?>

<beans

xmlns="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-2.5.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-2.5.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-2.5.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.5.xsd

">

<!-- 配置数据源 -->

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">

<property name="driverClassName" value="com.mysql.jdbc.Driver"/>

<property name="url" value="jdbc:mysql://localhost:3306/test?useUnicode=true&amp;characterEncoding=utf-8"/>

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

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

<!-- 连接池启动时的初始值 -->

<property name="initialSize" value="1"/>

<!-- 连接池的最大值 -->

<property name="maxActive" value="500"/>

<!-- 最大空闲值.当经过一个高峰时间后，连接池可以慢慢将已经用不到的连接慢慢释放一部分，一直减少到maxIdle为止 -->

<property name="maxIdle" value="2"/>

<!-- 最小空闲值.当空闲的连接数少于阀值时，连接池就会预申请去一些连接，以免洪峰来时来不及申请 -->

<property name="minIdle" value="1"/>

</bean>

<!-- 配置hibernate的sessionFactory -->

<bean id="sessionFactory" class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">

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

<property name="mappingResources">

<list>

<value>com/persia/model/Person.hbm.xml</value>

</list>

</property>

<!-- 1.首先在sessionFactory里面配置以上3条设置 -->

<!-- 2.然后得在类路径下面添加一个ehcache.xml的缓存配置文件 -->

<!-- 3.最后在要使用缓存的实体bean的映射文件里面配置缓存设置 -->

<!--使用二级缓存-->

<!-- 不使用查询缓存，因为命中率不是很高 -->

<!-- 使用Ehcache缓存产品 -->

<property name="hibernateProperties">

<value>

hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

hibernate.hbm2ddl.auto=update

hibernate.show\_sql=false

hibernate.format\_sql=false

hibernate.cache.use\_second\_level\_cache=true

hibernate.cache.use\_query\_cache=false

hibernate.cache.provider\_class=org.hibernate.cache.EhCacheProvider

</value>

</property>

</bean>

<!-- 配置Spring针对hibernate的事务管理器 -->

<bean id="txManager" class="org.springframework.orm.hibernate3.HibernateTransactionManager">

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

</bean>

<!-- 配置使用注解的方式来使用事务 -->

<tx:annotation-driven transaction-manager="txManager"/>

<!-- 使用手工配置的注解方式来注入bean -->

<context:annotation-config></context:annotation-config>

<!--定义要注入的业务bean -->

<bean id="personService" class="com.persia.service.impl.PersonServiceImpl"></bean>

<!--注入Struts 2的action -->

<bean id="personList" class="com.persia.struts2.action.PersonListAction"></bean>

</beans>

**6.SSJ:**

<?xml version="1.0" encoding="UTF-8"?>

<beans

xmlns="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-2.5.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-2.5.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop-2.5.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.5.xsd

">

<!-- 使用手工配置的注解方式来注入bean -->

<context:annotation-config></context:annotation-config>

<!-- 1.配置Spring集成JPA -->

<bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalEntityManagerFactoryBean">

<property name="persistenceUnitName" value="SpringJPAPU"/>

</bean>

<!--2.配置Spring针对JPA的事务 -->

<bean id="txManager" class="org.springframework.orm.jpa.JpaTransactionManager">

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

</bean>

<!--3.开启事务注解 -->

<tx:annotation-driven transaction-manager="txManager"/>

<!--以上3个Spring集成JPA的配置，在web项目先添加Spring支持，后添加JPA支持时会自动生成 -->

<!-- 配置业务bean -->

<bean id="personService" class="com.persia.service.impl.PersonServiceImpl"></bean>

<!-- 配置Struts的action -->

<bean name="/person/list" class="com.persia.struts.PersonListAction"/>

<bean name="/person/manage" class="com.persia.struts.PersonManageAction"/>

</beans>