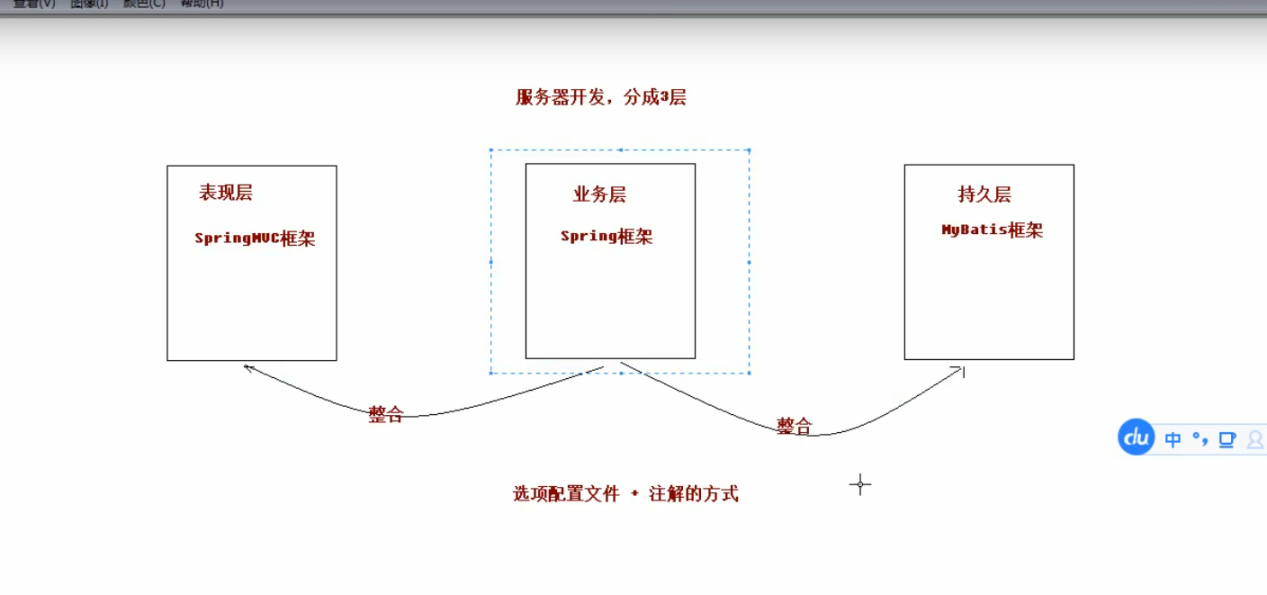
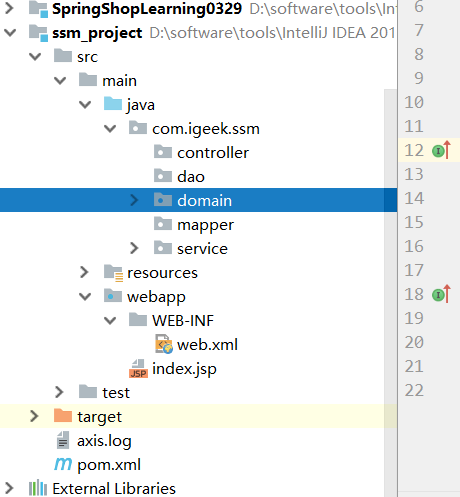
1. SSM介绍



# 首先整合spring框架，创建项目

1. 首先正常创建maven项目。创建相关目录



1. 引入依赖pom.xml



1. 引入web.xml配置文件



1. 配置SpringMVC的配置文件 springMVC-config.xml



1. 配置spring的配置文件 applicationContext.xml。用来扫描dao和service。



1. 编写dao和service层



1. 编写单元测试



# springMVC框架

就是在页面上写一个按钮或超链接，点击能够调用相应controller并返回相关页面。

1. web.xml
2. springMVC-config.xml
3. controller编写

# spring整合SpringMVC框架

即在服务器开启的时候就加载springmvc-config.xml配置文件。可以通过监听器实现。

|  |
| --- |
| *<!--监听springMVC--> <!--配置spring的监听器，默认只加载WEB-INF目录下的applicationContext.xml配置文件  目前spring的配置文件不在这个目录下，  我们需要再配置 context-param。 -->* <**listener**>  <**listener-class**>org.springframework.web.context.ContextLoaderListener</**listener-class**> </**listener**> <**context-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:spring/applicationContext.xml</**param-value**> </**context-param**> |

# 整合mybatis框架

1. sqlMapConfig.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* **<!DOCTYPE configuration  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-config.dtd"*>*** <**configuration**>   *<!--引入jdbcConfig.properties-->  <!--加载db.properties-->* <**properties resource="mybatis/jdbcConfig.properties"**/>   *<!-- 配置连接数据库的信息 -->* <**environments default="dev"**>  *<!-- 配置mybatis的环境 -->* <**environment id="dev"**>  <**transactionManager type="JDBC"**/>  *<!-- 配置事务管理 -->* <**dataSource type="POOLED"**>  <**property name="driver" value="${jdbc.driver}"**/>  <**property name="url" value="${jdbc.url}"**/>  <**property name="username" value="${jdbc.username}"**/>  <**property name="password" value="${jdbc.password}"**/>  </**dataSource**>  </**environment**>  </**environments**>   *<!--指定映射文件的位置-->* <**mappers**>  *<!--配置文件的方式-->  <!-- 1、 <mapper resource="com/igeek/ssm/xxxDao"/>-->  <!-- 2、 <mapper class="com.igeek.ssm.mapper.xxxDao"/>-->  <!-- 3、 <package name="com.igeek.ssm.mapper"/>-->   <!--扫描此包下的所有文件-->  <!--<mapper class="com.igeek.ssm.mapper.AccountDao"/>-->* <**package name="com.igeek.ssm.mapper"**/>  </**mappers**>  </**configuration**> |

1. AccountDao.java接口

|  |
| --- |
| **package** com.igeek.ssm.mapper;  **import** com.igeek.ssm.domain.Account; **import** org.apache.ibatis.annotations.Insert; **import** org.apache.ibatis.annotations.Select;  **import** java.util.List;  **public interface** AccountDao {   @Select(value = **"select \* from account"**)  **public** List<Account> selectAll();   @Insert(value = **"insert into account(id,name,money) values('2','Mike',1000)"**)  **public void** saveAccount(Account account); } |

1. 原始测试mybatis类

|  |
| --- |
| **package** com.igeek.ssm.mapper;  **import** com.igeek.ssm.domain.Account; **import** org.apache.ibatis.io.Resources; **import** org.apache.ibatis.session.SqlSession; **import** org.apache.ibatis.session.SqlSessionFactory; **import** org.apache.ibatis.session.SqlSessionFactoryBuilder; **import** org.junit.After; **import** org.junit.Before; **import** org.junit.Test;  **import** java.io.IOException; **import** java.io.InputStream; **import** java.util.List;  **public class** AccountDaoTest\_mybatis {  **private** AccountDao **accountDao**;  **private** InputStream **in** = **null**;  **private** SqlSession **sqlSession** = **null**;   @Before  **public void** init() **throws** Exception {  System.***out***.println(**"进入AccountDaoTest\_mybatis.init()............."**);  *//加载配置文件* **in** = Resources.*getResourceAsStream*(**"mybatis/sqlMapConfig.xml"**);  *//创建sqlSessionFactory对象* SqlSessionFactory sqlSessionFactory = **new** SqlSessionFactoryBuilder().build(**in**);  *//获取SqlSession对象* **sqlSession** = sqlSessionFactory.openSession();  }    @Test  **public void** testSelectAll(){  **accountDao** = **sqlSession**.getMapper(AccountDao.**class**);  List<Account> accountList = **accountDao**.selectAll();  **for**(Account account:accountList){  System.***out***.println(account);  }  }   @After  **public void** after() **throws** IOException {  System.***out***.println(**"进入AccountDaoTest\_mybatis.after()............."**);  *//关闭资源* **sqlSession**.close();  **in**.close();   } } |
|  |

1. 测试完毕，开始springMVC整合mybatis。即有springContext.xml配置mybatis的配置
   1. 配置连接池
   2. 配置SqlSessionFactory
   3. 配置AccountDao接口所在包

|  |
| --- |
| *<?***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:aop="http://www.springframework.org/schema/aop"  xmlns:tx="http://www.springframework.org/schema/tx"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/tx  http://www.springframework.org/schema/tx/spring-tx.xsd  http://www.springframework.org/schema/aop  http://www.springframework.org/schema/aop/spring-aop.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd"**>   *<!--开启注解的扫描，只希望处理dao和service，不处理controller(controller由springMVC进行处理)-->* <**context:component-scan base-package="com.igeek.ssm"**>  *<!--配置controller不扫描-->* <**context:exclude-filter type="annotation" expression="org.springframework.stereotype.Controller"**/>  </**context:component-scan**>   *<!--Spring整合Mybatis框架-->  <!--1、配置连接池-->  <!--加载db.properties-->* <**bean id="propertyConfigurer"  class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"**>  <**property name="location" value="classpath:jdbcConfig.properties"** />  </**bean**>  <**bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"**>  <**property name="driverClass" value="${jdbc.driver}"**/>  <**property name="jdbcUrl" value="${jdbc.url}"**/>  <**property name="user" value="${jdbc.username}"**/>  <**property name="password" value="${jdbc.password}"**/>  </**bean**>   *<!--2、配置SqlSessionFactory-->* <**bean id="sqlSessionFactoryBean" class="org.mybatis.spring.SqlSessionFactoryBean"**>  <**property name="dataSource" ref="dataSource"**/>  </**bean**>   *<!--3、配置AccountDao接口所在包-->* <**bean id="mapperScannerConfigurer" class="org.mybatis.spring.mapper.MapperScannerConfigurer"**>  <**property name="basePackage" value="com.igeek.ssm.mapper"**/>  </**bean**> </**beans**> |
|  |

然后再dao层注解：@Repository

service层 @Service

慢慢搞即可

# 配置事务

|  |
| --- |
| *<!--4、配置Spring框架声明式事务管理--> <!--a、配置事务管理器-->* <**bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager"**>  <**property name="dataSource" ref="dataSource"**/> </**bean**> *<!--b、配置事务通知-->* <**tx:advice id="txAdvice" transaction-manager="transactionManager"**>  <**tx:attributes**>  <**tx:method name="find\*" read-only="true"**/> *<!--设置以find开头的方法，都为只读-->* <**tx:method name="\*" isolation="DEFAULT"**/>  </**tx:attributes**> </**tx:advice**> *<!--c、配置AOP增强-->* <**aop:config proxy-target-class="true"**>  <**aop:advisor advice-ref="txAdvice" pointcut="execution(\* com.igeek.ssm.service.impl.\*ServiceImpl.\*(..))"**/> </**aop:config**> |