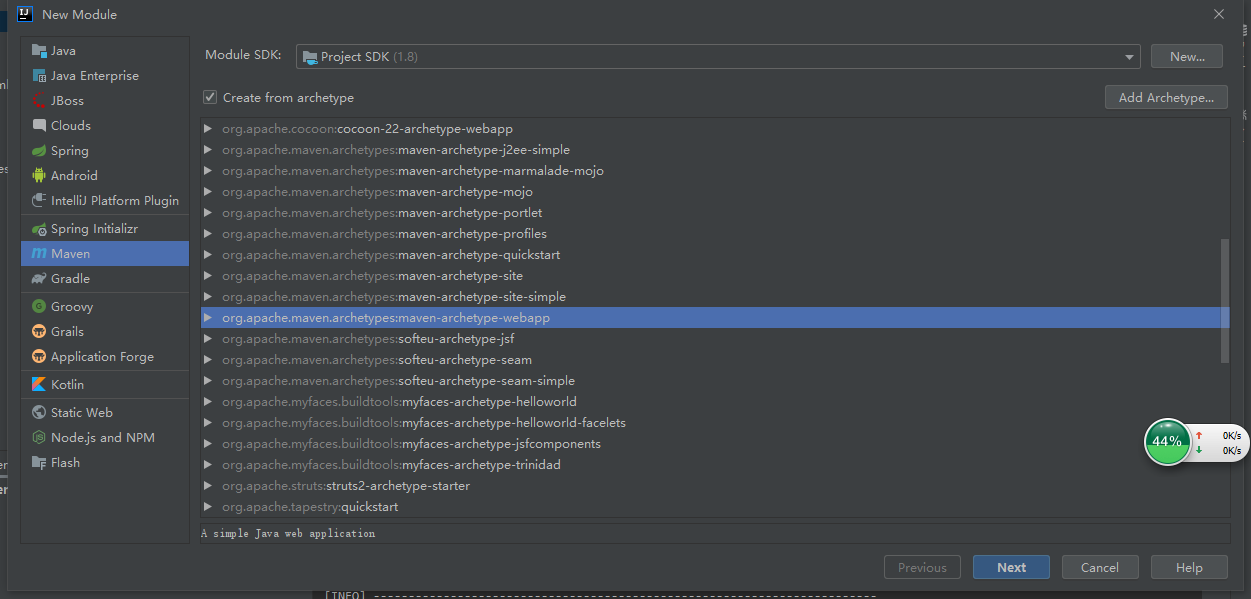
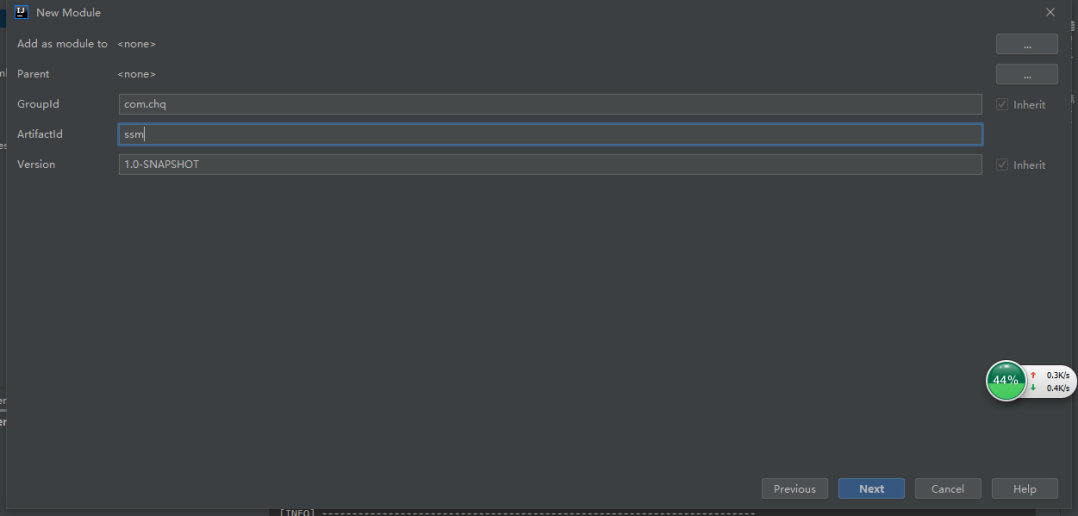
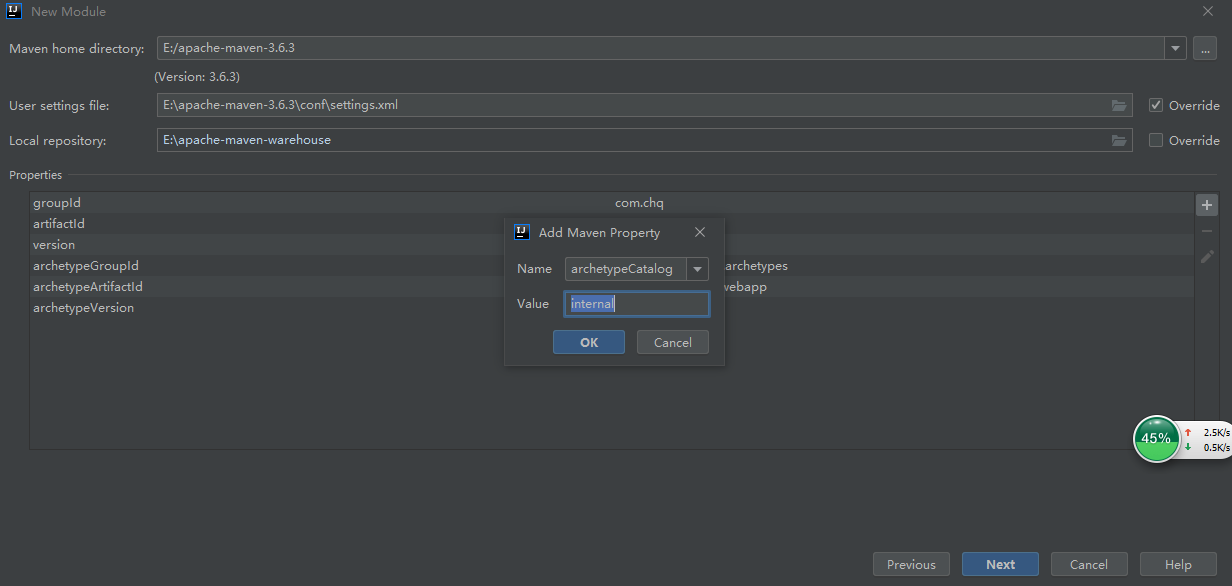
# SSM

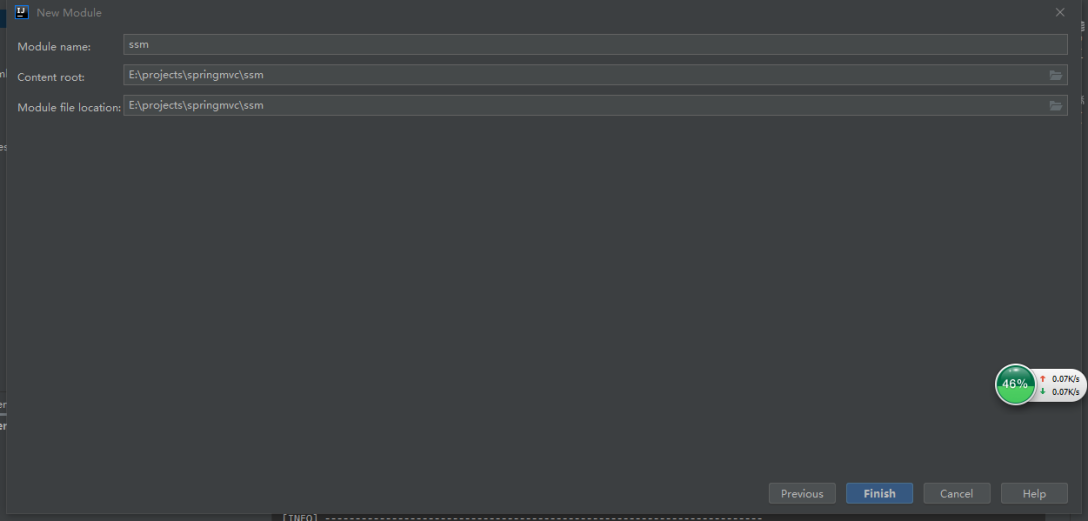
# 环境搭建

新建一个module，new>module>...









新建java目录并右键make directory as为source root

新建resources目录比右键make directory as为 resource root

## 导入依赖

<properties>  
 <spring.version>5.0.2.RELEASE</spring.version>  
 <slf4j.version>1.6.6</slf4j.version>  
 <log4j.version>1.2.12</log4j.version>  
 <mysql.version>5.1.6</mysql.version>  
 <mybatis.version>3.4.5</mybatis.version>  
</properties>

<dependencies>  
 <!-- spring -->  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>1.9.5</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-web</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <!-- 整合junit，spring5需要junit4.12以上 -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-test</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-tx</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-jdbc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <!-- 数据库驱动 -->  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>${mysql.version}</version>  
 </dependency>  
 <!-- servlet和jsp -->  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>servlet-api</artifactId>  
 <version>2.5</version>  
 <scope>provided</scope>  
 </dependency>  
 <dependency>  
 <groupId>javax.servlet.jsp</groupId>  
 <artifactId>jsp-api</artifactId>  
 <version>2.0</version>  
 <scope>provided</scope>  
 </dependency>  
 <dependency>  
 <groupId>jstl</groupId>  
 <artifactId>jstl</artifactId>  
 <version>1.2</version>  
 </dependency>  
 <!-- log start -->  
 <dependency>  
 <groupId>log4j</groupId>  
 <artifactId>log4j</artifactId>  
 <version>${log4j.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-api</artifactId>  
 <version>${slf4j.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-log4j12</artifactId>  
 <version>${slf4j.version}</version>  
 </dependency>  
 <!-- log end -->  
 <dependency>  
 <groupId>org.mybatis</groupId>  
 <artifactId>mybatis</artifactId>  
 <version>${mybatis.version}</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.mybatis</groupId>  
 <artifactId>mybatis-spring</artifactId>  
 <version>1.3.0</version>  
 </dependency>  
 <dependency>  
 <groupId>c3p0</groupId>  
 <artifactId>c3p0</artifactId>  
 <version>0.9.1.2</version>  
 <type>jar</type>  
 <scope>compile</scope>  
 </dependency>  
 <!-- junit -->  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.12</version>  
 <scope>test</scope>  
 </dependency>  
</dependencies>

新建一些演示文件

package com.chq.bean;  
  
public class Account {  
 private Long id;  
 private String name;  
 private Double balance;  
  
 public Long getId() {  
 return id;  
 }  
  
 public void setId(Long id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public Double getBalance() {  
 return balance;  
 }  
  
 public void setBalance(Double balance) {  
 this.balance = balance;  
 }  
  
 @Override  
 public String toString() {  
 return "Account{" +  
 "id=" + id +  
 ", name='" + name + '\'' +  
 ", balance=" + balance +  
 '}';  
 }  
}

package com.chq.dao;  
  
import java.util.List;  
  
public interface Account {  
 List<Account> findAll();  
}

package com.chq.service;  
  
import com.chq.dao.Account;  
  
import java.util.List;  
  
public interface AccountService {  
 List<Account> findAll();  
}

package com.chq.service.impl;  
  
import com.chq.dao.Account;  
import com.chq.service.AccountService;  
  
import java.util.List;  
  
public class AccountServiceImpl implements AccountService {  
 @Override  
 public List<Account> findAll() {  
 System.*out*.println("查询所有");  
 return null;  
 }  
}

package com.chq.controller;  
  
public class AccountCountroller {  
}

## 整合spring框架

在resources目录下new>xml...file>spring config新建applicationContext.xml配置文件

<?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.xsd  
 http://www.springframework.org/schema/context  
 http://www.springframework.org/schema/context/spring-context.xsd">  
  
 <context:component-scan base-package="com.chq">  
 <context:exclude-filter type="annotation" expression="org.springframework.stereotype.Controller"/>  
 </context:component-scan>  
</beans>

在resources下新建log4j.properties

# Configure logging for testing: optionally with log file  
log4j.rootLogger=WARN, stdout  
# log4j.rootLogger=WARN, stdout, logfile  
  
log4j.appender.stdout=org.apache.log4j.ConsoleAppender  
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout  
log4j.appender.stdout.layout.ConversionPattern=%d %p [%c] - %m%n  
  
log4j.appender.logfile=org.apache.log4j.FileAppender  
log4j.appender.logfile.File=target/spring.log  
log4j.appender.logfile.layout=org.apache.log4j.PatternLayout  
log4j.appender.logfile.layout.ConversionPattern=%d %p [%c] - %m%n

在AccountServiceImpl文件加上注解并测试一下

@Service("accountService")  
public class AccountServiceImpl implements AccountService {

public static void main(String[] args) {  
 ApplicationContext ac = new ClassPathXmlApplicationContext("applicationContext.xml");  
 AccountService accountService = (AccountService) ac.getBean("accountService");  
 accountService.findAll();  
}

## 整合springmvc

步骤和springmvc文档入门案例建立一样，只是在springmvc.xml中这个不一样

<!-- 注解扫描的包 -->  
<context:component-scan base-package="com.chq" use-default-filters="false">  
 <context:include-filter type="annotation" expression="org.springframework.stereotype.Controller"/>  
</context:component-scan>

## Spring 整合 springmvc

在web.xml添加

<!-- 配置加载 Spring 容器的监听器，默认只加载 WEB-INF 下的 applicationContext.xml -->  
<listener>  
 <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  
</listener>  
<!-- 指定配置文件位置 -->  
<context-param>  
 <param-name>contextConfigLocation</param-name>  
 <param-value>classpath:applicationContext.xml</param-value>  
</context-param>

即项目启动的时候加载spring的配置文件

这样就可以在controller中使用@Autowired注入service对象。

## Spring整合mybatis

在applicationContext.xml添加

<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">  
 <property name="driverClass" value="com.mysql.jdbc.Driver"/>  
 <property name="jdbcUrl" value="jdbc:mysql://localhost:3306/db\_demo"/>  
 <property name="user" value="root"/>  
 <property name="password" value=""/>  
</bean>  
  
<!-- 配置 SqlSessionFactory -->  
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">  
 <!-- 注入数据源 -->  
 <property name="dataSource" ref="dataSource"/>  
 <!-- 配置 Mapper 配置文件的扫描路径，如果配置文件与接口目录结构相同可以不用配置 -->  
 <property name="mapperLocations">  
 <array>  
 <value>classpath:mapper/\*.xml</value>  
 </array>  
 </property>  
</bean>  
  
<!-- 配置 Mapper 扫描器 -->  
<bean id="mapperScannerConfigurer" class="org.mybatis.spring.mapper.MapperScannerConfigurer">  
 <property name="basePackage" value="com.chq.dao"/>  
</bean>

在dao层修改

@Repository  
public interface AccountMapper {

在resources下新建mapper/AccountMapper.xml

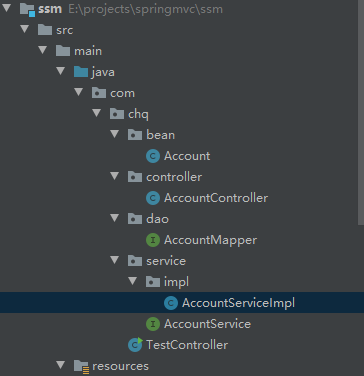
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper  
 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="com.chq.dao.AccountMapper">  
 <!-- 配置查询所有用户 -->  
 <select id="findAll" resultType="com.chq.bean.Account">  
 SELECT \* FROM account  
 </select>  
</mapper>

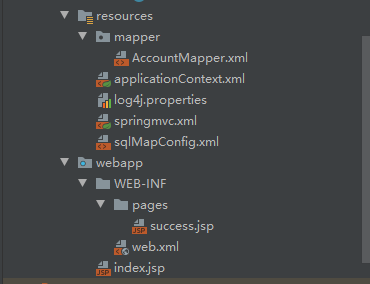
## Spring整合mybatis事务

在applicationContext.xml添加

<!-- 配置事务管理器 -->  
<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">  
 <property name="dataSource" ref="dataSource"/>  
</bean>  
  
<!-- 配置事务通知 -->  
<tx:advice id="txAdvice" transaction-manager="transactionManager">  
 <tx:attributes>  
 <!-- 默认所有方法都有事务支持，读写型事务 -->  
 <tx:method name="\*" read-only="false" propagation="REQUIRED"/>  
 <!-- 查询方法支持事务，只读型 -->  
 <tx:method name="list\*" read-only="true" propagation="SUPPORTS"/>  
 </tx:attributes>  
</tx:advice>  
  
<!-- 事务织入 -->  
<aop:config>  
 <aop:advisor advice-ref="txAdvice" pointcut="execution(\* com.chq.service.impl.\*.\*(..))"/>  
</aop:config>

最后目录如下：





项目文件存在百度云》学习》java