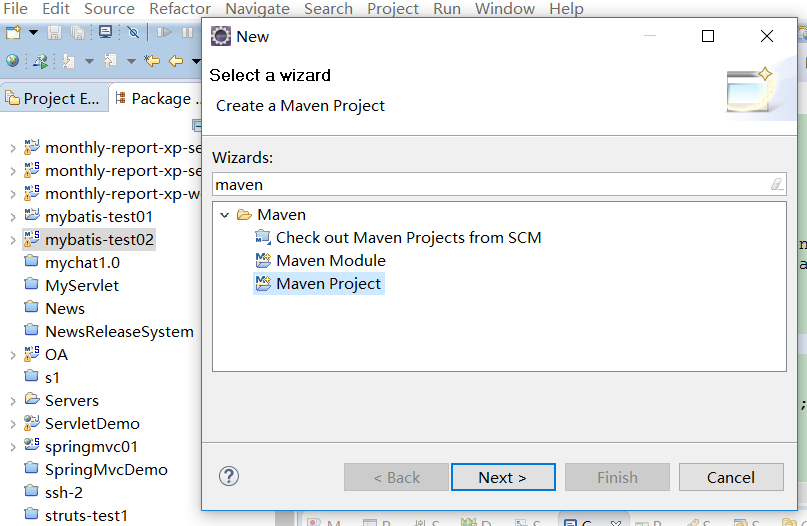
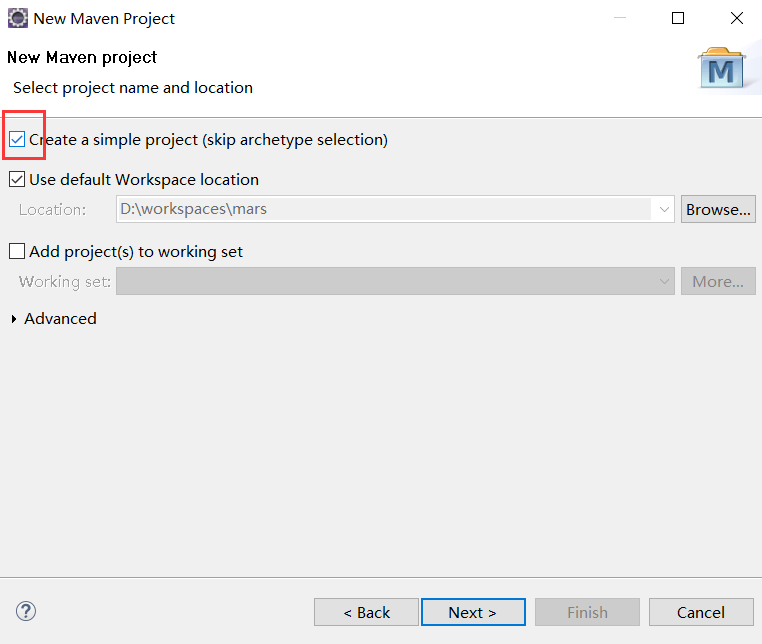
Maven-SSM环境搭建文档

1. 创建Maven Web项目
   1. 右键空白搜索maven



* 1. 勾选Create a simple project (创建一个简单的项目)　，并选择Next



* 1. 录入相关的Artiface参数

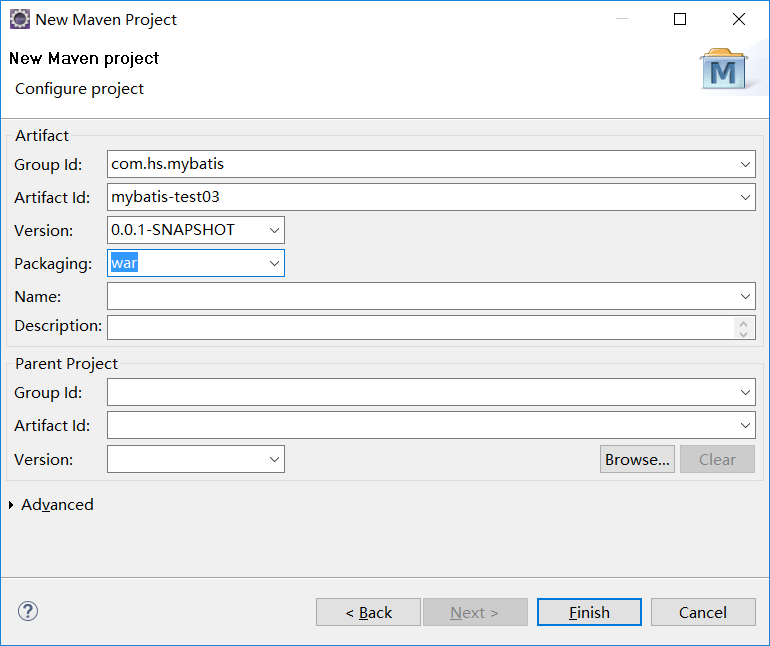
Group Id：类似于包名

Artiface Id:项目的名称

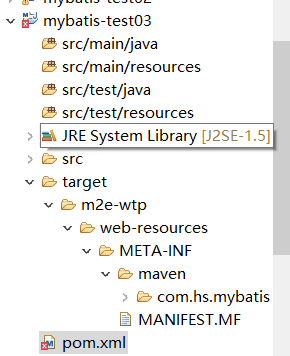
Version：初始的版本号，一般不需要改动

Packaging：选择项目的打包类型，选择jar就是一个java项目，选择war就是一个web项目

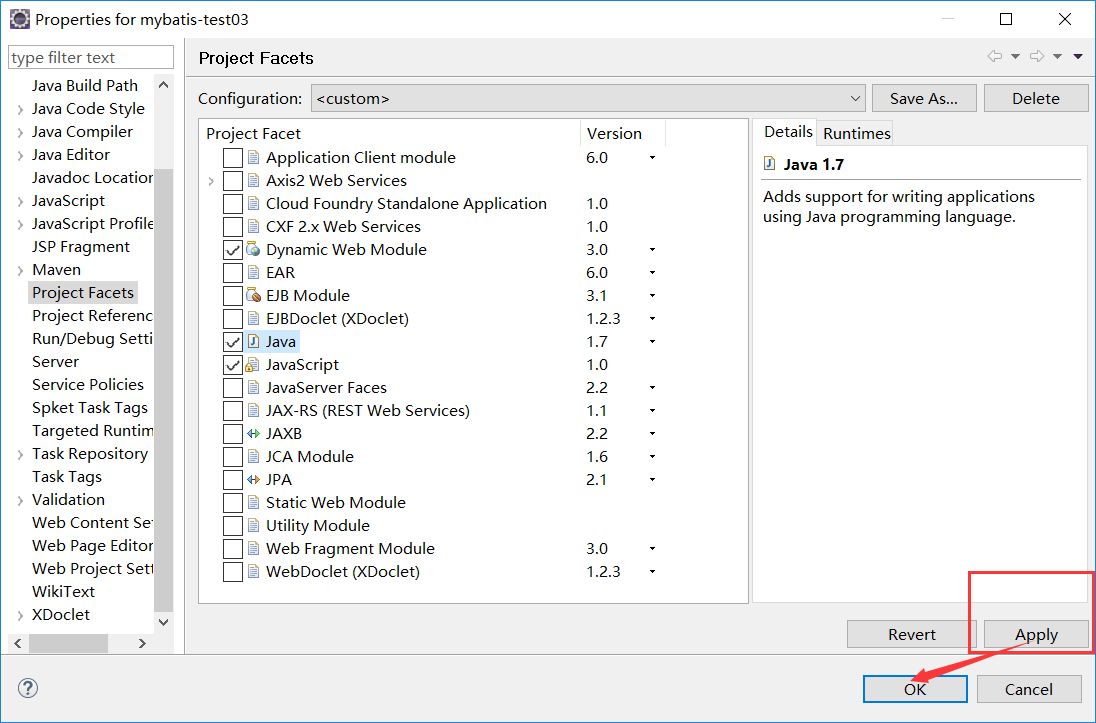
其他选项设置为空，点击Next创建项目，如下图：



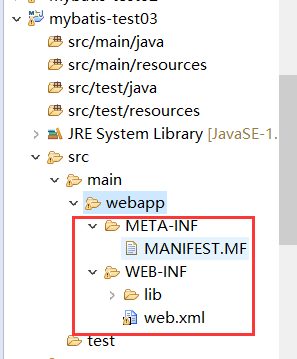
目录结构如下：



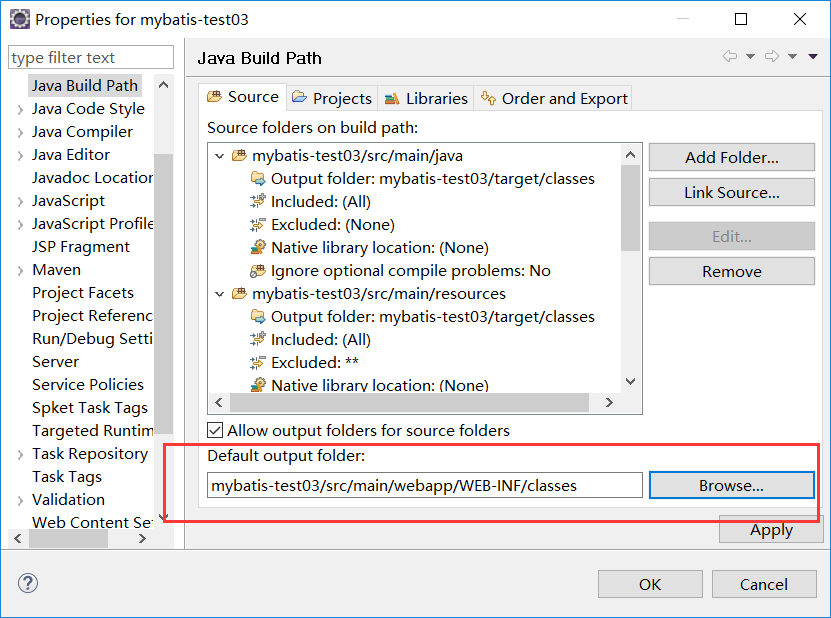
* 1. 创建好的项目是一个java项目，我们需要将这个项目转化为java web项目，在项目上右键，选择Properties，在打开的对话框中点击Project Facts，并且勾选"Dynamic Web Module"和"java"这两项，apply--》OK



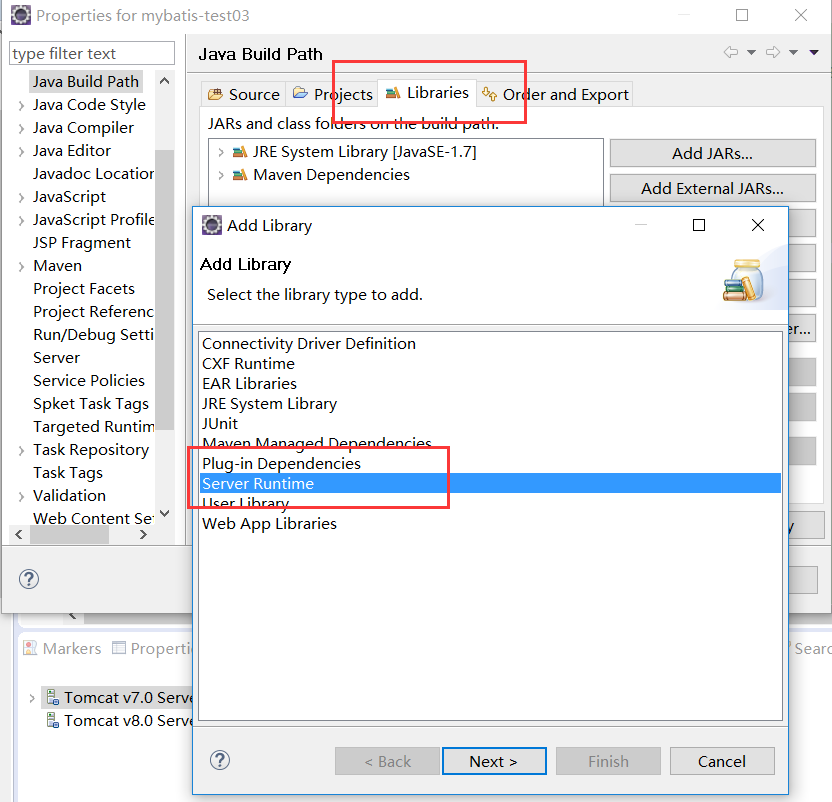
* 1. 标准的maven项目是没有WebContent 文件夹的，我们需要从普通动态web项目中复制WebContent文件夹的下的两个文件夹：META-INF、WEB-INF并粘贴到webapp文件夹下，如下图：



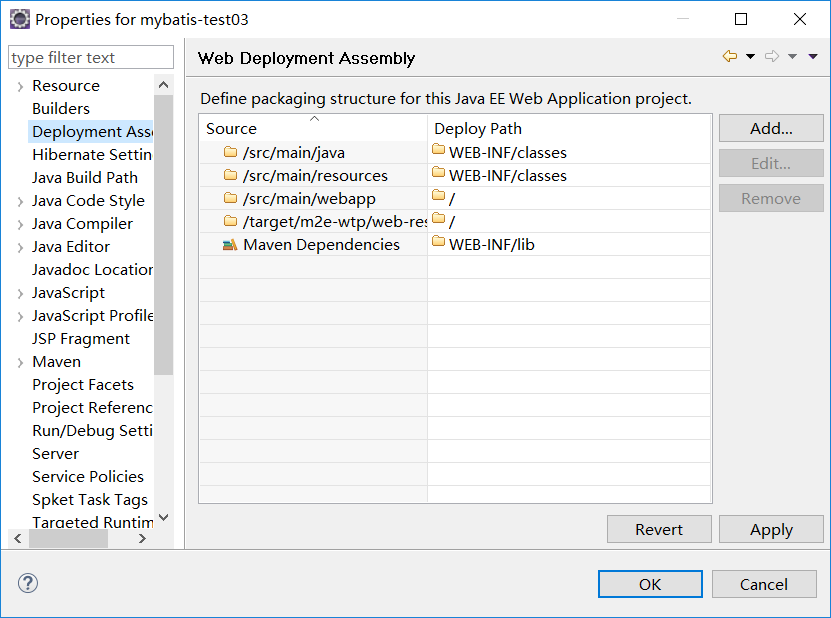
* 1. 项目右键-》Build Path--》Configure Build Path。在**Source**选项卡中修改**Default output folder**路径为"/src/main/webapp/WEB-INF/classes" （WEB-INF下没有classes文件夹则新建）



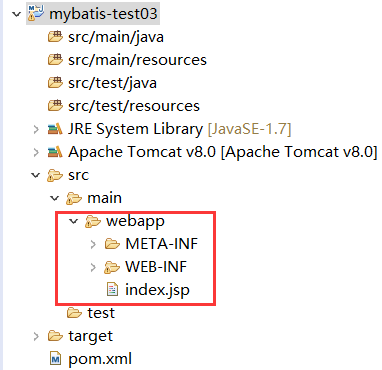
* 1. 添加servlet容器

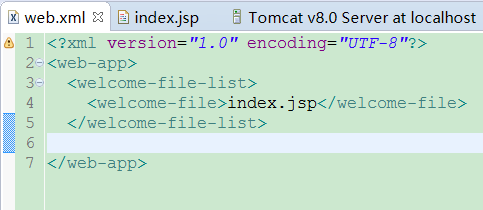


* 1. 添加部署依赖。项目右键-》Properties--》Deployment Assembly。依赖清单如下图，可自行添加和删除。

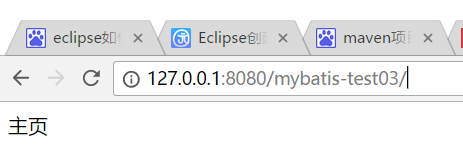


* 1. 新建index.jsp文件并在web.xml中将其配置成主页





* 1. 将此web项目发布到servlet容器中并测试



参考：<http://www.cnblogs.com/codejackanapes/p/5461940.html>

1. 整合spring和mybatis
   1. 在pom.xml中配置依赖

需要的jar包如下：

junit-4.12（版本较低可能对测试有影响）

commons-logging-1.1.2

mybatis -3.4.1，

c3p0-0.9.2，

mysql-connector-java-5.1.22，

mybatis-spring-1.3.1，

spring-context-4.0.0.RELEASE，

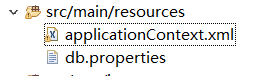
spring-core-4.0.0.RELEASE，

spring-beans-4.0.0.RELEASE，spring-aop-4.0.0.RELEASE，

spring-jdbc-4.0.0.RELEASE，spring-tx-4.0.0.RELEASE，

spring-web-4.0.0.RELEASE，spring-webmvc-4.0.0.RELEASE，

* 1. 在resources中创建applicationContext.xml文件
     1. 配置数据源

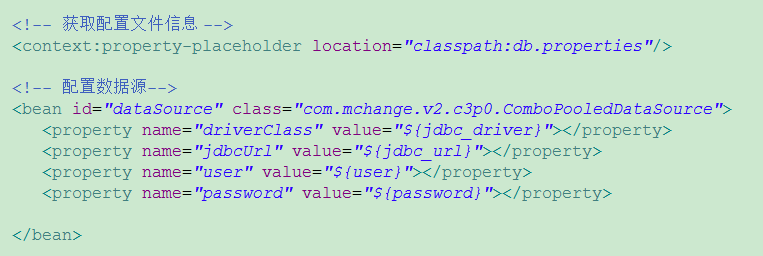


jdbc\_driver = com.mysql.jdbc.Driver

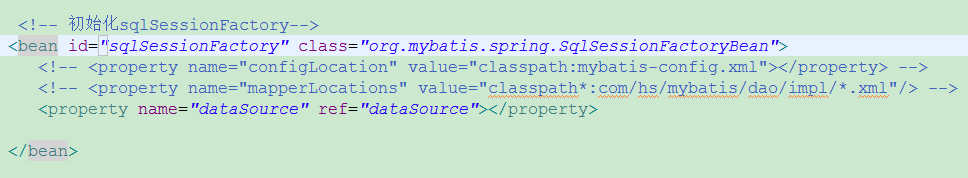
jdbc\_url = jdbc:mysql://localhost:3306/testdb?useUnicode=true&amp;characterEncoding=UTF-8

user = root

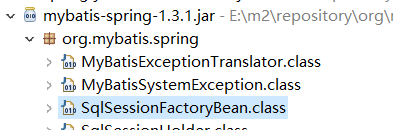
password = root



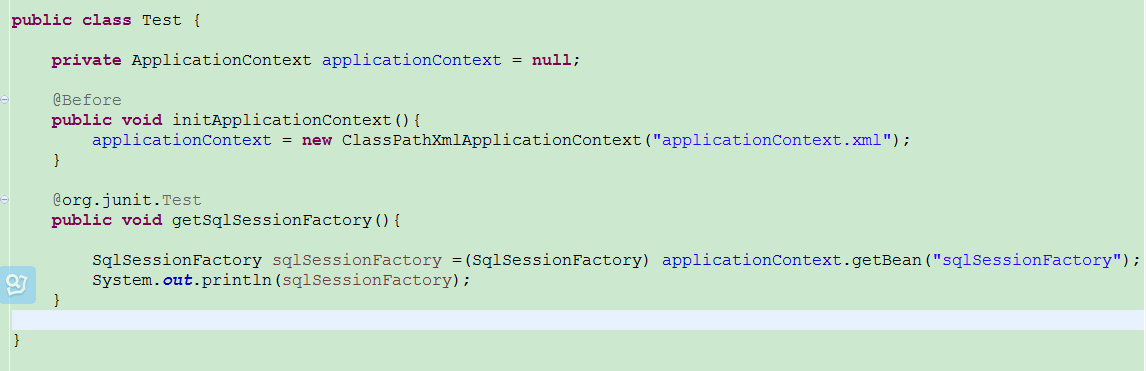
* + 1. 将数据源注入到org.mybatis.spring.SqlSessionFactoryBean类中



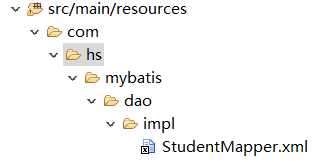
具体配置可以查看此类

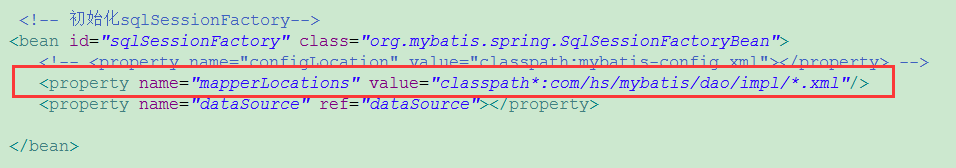


Ps：测试是否正确初始化org.apache.ibatis.session.SqlSessionFactory;

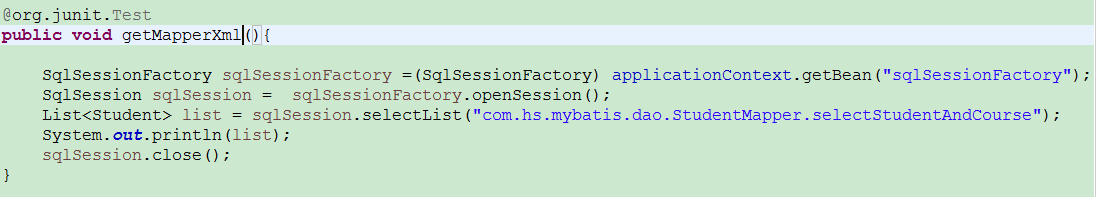


3．创建mapper.xml文件，并在org.mybatis.spring.SqlSessionFactoryBean中注入xml文件路径，然后使用SqlSession测试

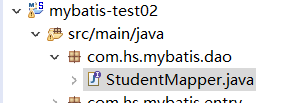


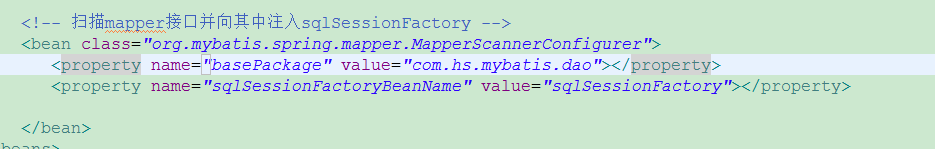


测试：

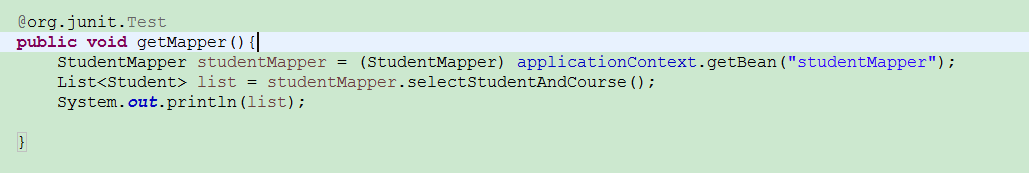


4．新建dao接口与之前创建的xml名字对应，并在spring容器中配置扫描接口对应的类，并向接口中注入sqlSessionFFactory





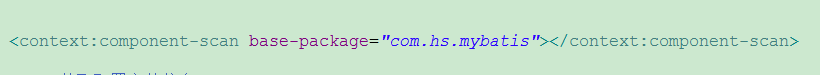
测试：



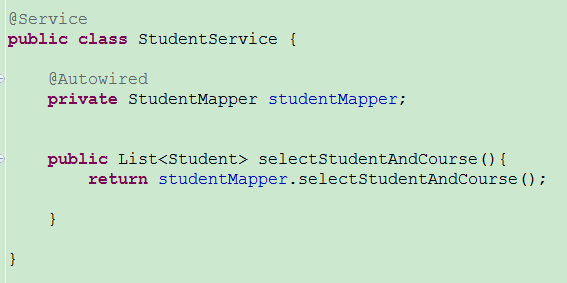
Ps：测试成功则证明spring与mybatis已整合成功！

三，通过注解的方式将dao接口注入到service层并配置相关事务

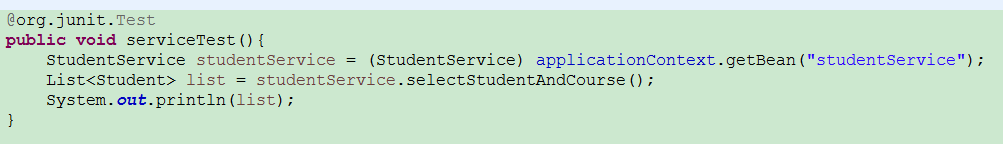
1. 在spring IOC容器中配置扫描器



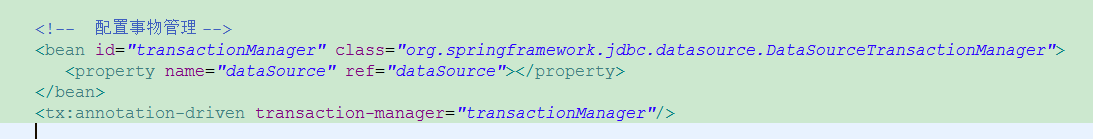
1. 编写service层，并通过注解注入的方式将dao注入到service类中



测试：

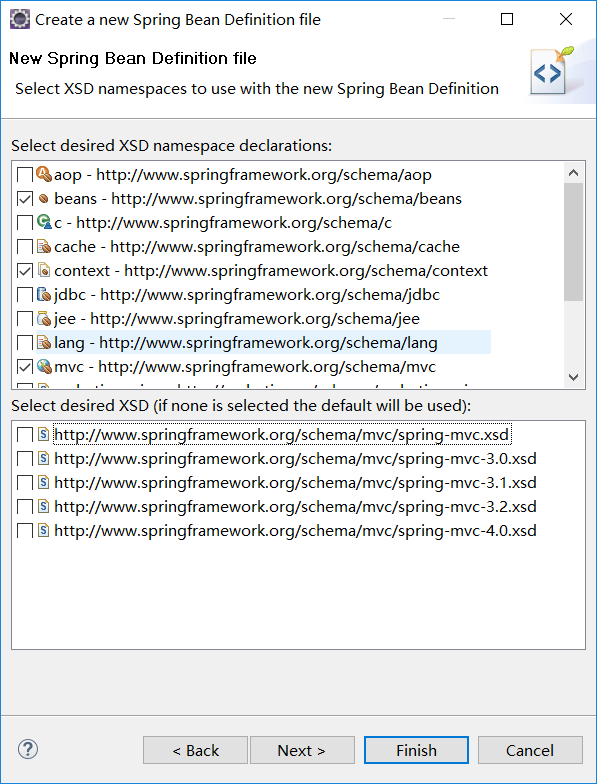


1. 在spring IOC容器中配置事务

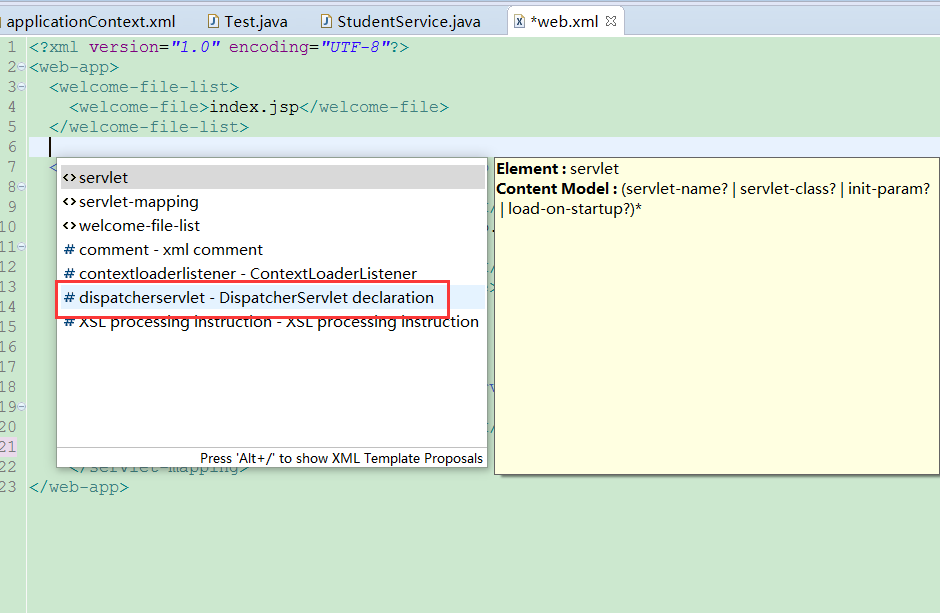


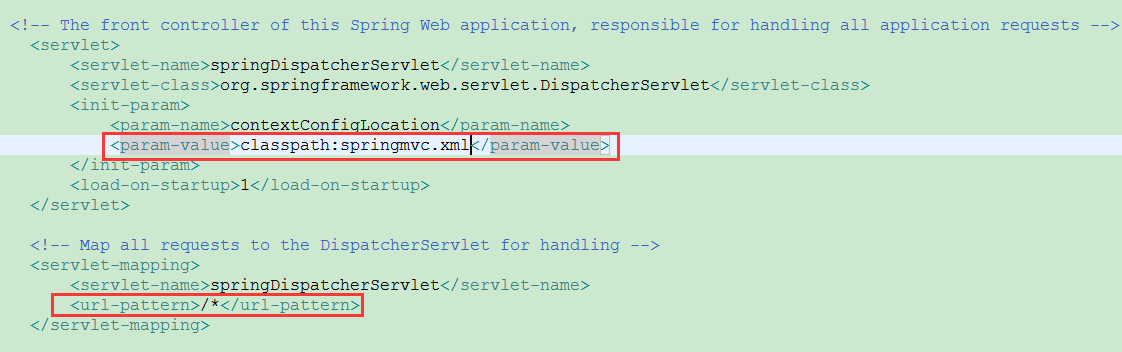
四，整合springmvc

1. 创建springmvc.xml文件，并加入beans,context,mvc的xsd文档

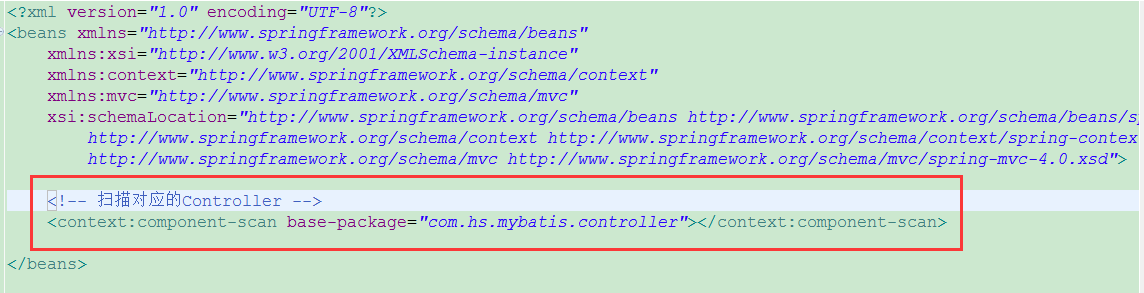


1. 使用Alt+/在web.xml中配置org.springframework.web.servlet.DispatcherServlet（eclipse安装了spring开发插件后会提示）

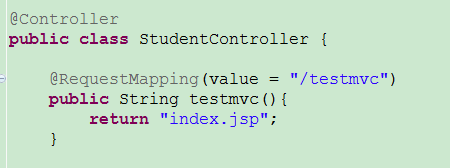




1. 配置springmvc.xml
2. 配置context:component-scan扫描controller类

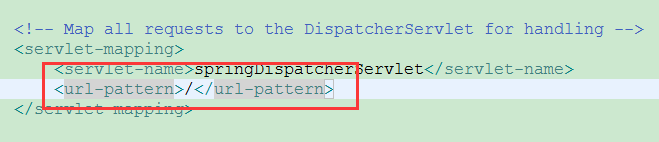


测试：



如果出现以下警告请将springDispatcherServlet中的url-pattern改成 /

警告: No mapping found for HTTP request with URI [/springmvc01/] in DispatcherServlet with name 'springDispatcherServlet'

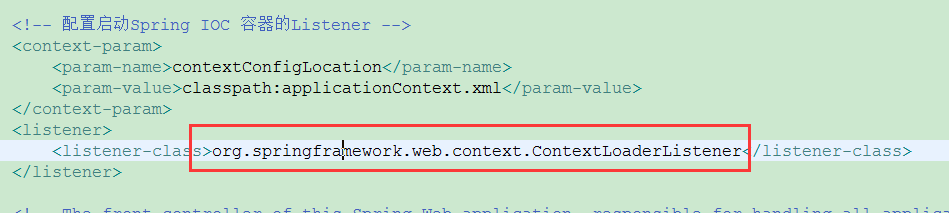


1. 在web.xml中配置启动spring IOC容器的ContextLoaderListener

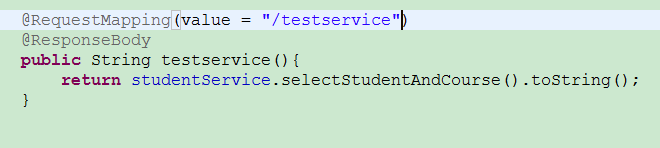
原因：

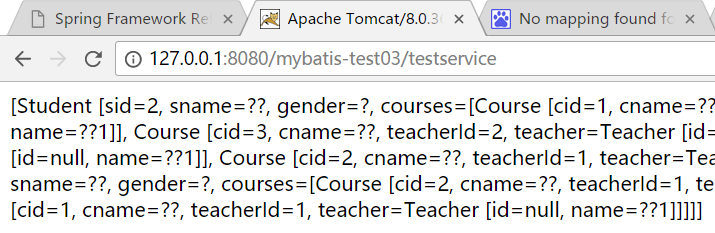


配置org.springframework.web.context.ContextLoaderListener



测试：





1. 解决乱码问题

在springmvc.xml文件中配置mvc:annotation-driven

