背景：

在使用idea+maven搭建ssm（即spring+springMVC+mybatis）的时候遇到两个问题。

一、spirng和mybatis的配置文件放在哪里？

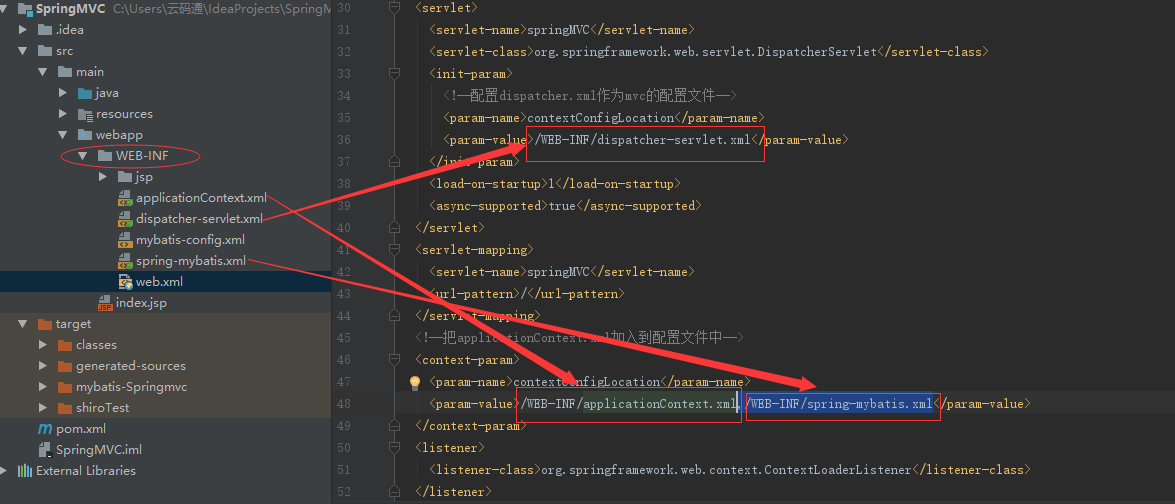
二、src/main/java/下的xxMapper.xml不编译，那么xxMapper.xml应该放在哪里？

之前不用maven，所以直接在src下，创建一个resource包或者config包亦或者mapping包，将xml放在那里，然后配置路径的时候使用classpath：resource/xxx.xml即可。然而在使用maven之后，以上办法是行不通，根据你的操作不同，会有各种不同的报错。当然以上两条其实是一个问题，只是在我整个过程中，对于以上两个错误，采用了两种方法解决。

意图：

理解maven，在构建时，默认只会编译src/main/java下的java文件和src/main/resource的资源文件（在idea下，你需要把前者标记为Sources，后者标记为Resources，后面补上如何标记）。

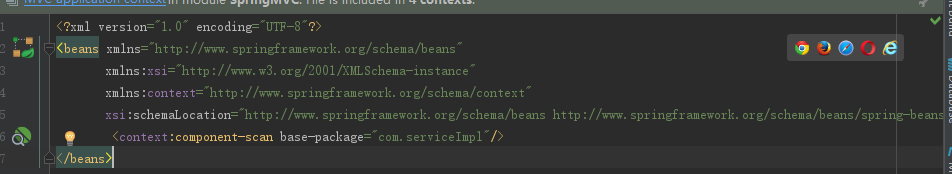
在解决第一个问题的时候，我将spirng和mybatis的配置文件放在了，/WEB-INF/下面。因为springMVC是会默认读取/WEB-INF/applicationContext.xml。所以项目是可以找到/WEB-INF/这个路径的。因此在配置springmvc的时候，可以如下图配置



配这个图，重点是想说明一下maven项目下，**/WEB-INF/下的资源文件也是会编译的，能够被项目正确的读取到。**

当然另外也想把此文写成springmvc整合mybatis的文章，说明一下三个文件的配置

applicationContext.xml 是上下文文件，而我这个文件里面就只有扫描service的配置



dispatcher-servlet.xml是springmvc的组件

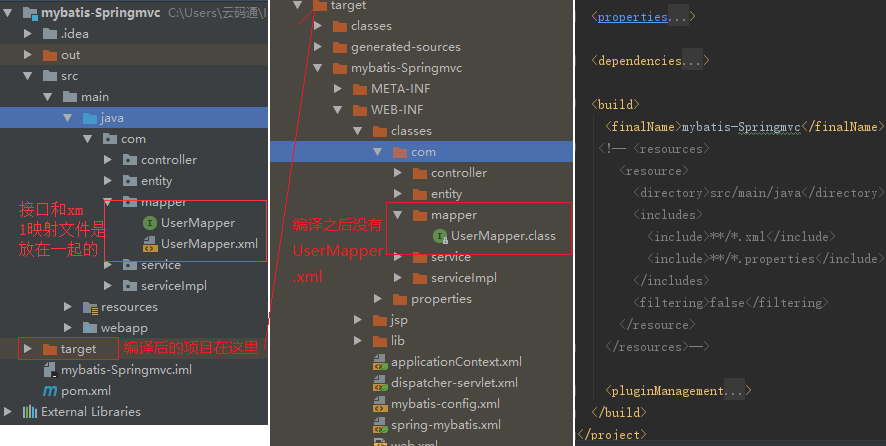
<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xmlns:mvc="http://www.springframework.org/schema/mvc"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd">  
 <!--此文件负责整个mvc中的配置-->  
  
 <!--启用spring的一些annotation -->  
 <context:annotation-config/>  
  
 <!-- 配置注解驱动 可以将request参数与绑定到controller参数上 -->  
 <mvc:annotation-driven/>  
  
 <!--静态资源映射-->  
 <mvc:default-servlet-handler/>  
  
 <!-- 对模型视图名称的解析，即在模型视图名称添加前后缀(如果最后一个还是表示文件夹,则最后的斜杠不要漏了) 使用JSP-->  
 <!-- 默认的视图解析器 在上边的解析错误时使用 (默认使用html)- -->  
 <bean id="defaultViewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">  
 <property name="viewClass" value="org.springframework.web.servlet.view.JstlView"/>  
 <property name="prefix" value="/WEB-INF/jsp/"/><!--设置JSP文件的目录位置-->  
 <property name="suffix" value=".jsp"/>  
 </bean>  
 <!-- 自动扫描装配 -->  
 <context:component-scan base-package="com.controller"/>  
</beans>

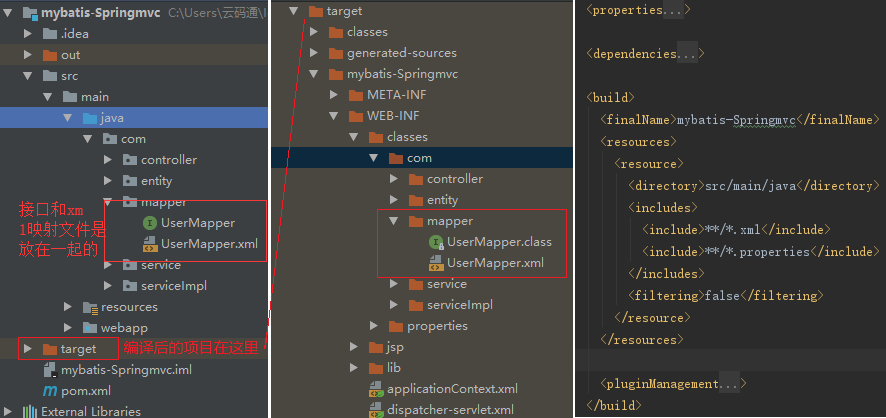
spring-mybatis.xml是spring整合mybatis的配置文件，里面有 引入.properties、数据源 扫描mapper、sqlSessionFactory配置

<?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"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 xsi:schemaLocation=" http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans-3.0.xsd  
 http://www.springframework.org/schema/aop  
 http://www.springframework.org/schema/aop/spring-aop-3.0.xsd  
 http://www.springframework.org/schema/context  
 http://www.springframework.org/schema/context/spring-context-3.0.xsd">  
  
 <!-- 此文件负责配置数据源和mybatis的配置-->  
  
 <!-- 引入jdbc配置文件 -->  
 <bean id="propertyConfigurer" class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">  
 <property name="locations">  
 <list>  
 <value>classpath:properties/db.properties</value>  
 <!--要是有多个配置文件，只需在这里继续添加即可 -->  
 </list>  
 </property>  
 </bean>  
  
 <!-- 配置数据源 -->  
 <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">  
 <!-- 不使用properties来配置 -->  
 <!-- <property name="driverClassName" value="com.mysql.jdbc.Driver" /> <property name="url" value="jdbc:mysql://localhost:3306/learning" /> <property name="username" value="root" /> <property name="password" value="christmas258@" /> -->  
 <!-- 使用properties来配置 -->  
 <property name="driverClassName">  
 <value>${jdbc\_driver}</value>  
 </property>  
 <property name="url">  
 <value>${jdbc\_url}</value>  
 </property>  
 <property name="username">  
 <value>${jdbc\_username}</value>  
 </property>  
 <property name="password">  
 <value>${jdbc\_password}</value>  
 </property>  
 </bean>  
  
 <!-- 自动扫描了所有的XxxxMapper.xml对应的mapper接口文件，这样就不用一个一个手动配置Mpper的映射了，只要Mapper接口类和Mapper映射文件对应起来就可以了。 -->  
 <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">  
 <property name="basePackage" value="com.mapper" />  
 <property name="sqlSessionFactoryBeanName" value="sqlSessionFactory" />  
 </bean>  
  
 <!-- 配置Mybatis的文件 ，mapperLocations配置\*\*Mapper.xml文件位置，configLocation配置mybatis-config文件位置-->  
 <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">  
 <property name="dataSource" ref="dataSource" />  
 <property name="mapperLocations" value="classpath:com/mapper/\*.xml"/>  
 <!--<property name="configLocation" value="classpath:mybatis/mybatis-config.xml" />-->  
 </bean>  
  
</beans>

第一个问题算是解决了，那么第二个问题来了。像这种spirngmvc这样的全局式的配置文件放在/WEB-INF/下还勉强可以理解，但是xxxmapper.xml这种非全局性的配置，而且个数还很多,放在/WEB-INF/下就显得不美观，结构也不够清晰。

先来看两张对比图。



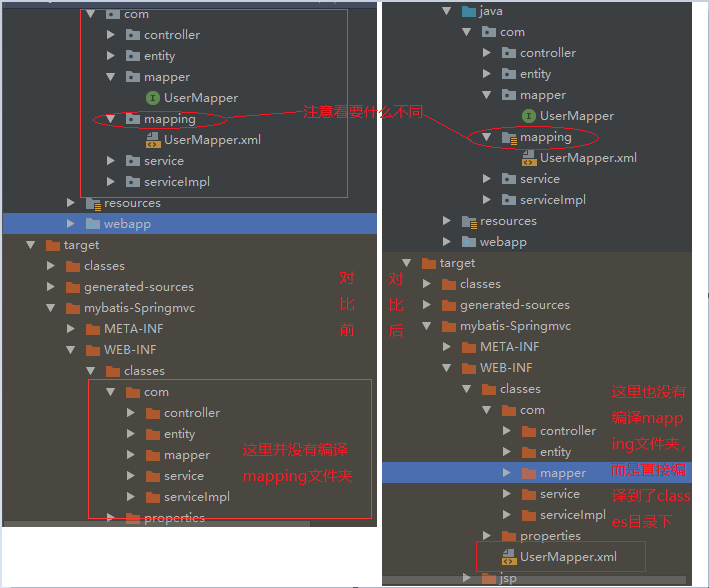


通过两张图，能够很清楚的发现有什么不同。另外附上注释掉的代码。

<resources>  
 <resource>  
 <directory>src/main/java</directory>  
 <includes>  
 <include>\*\*/\*.xml</include>  
 <include>\*\*/\*.properties</include>  
 </includes>  
 <filtering>false</filtering>  
 </resource>  
</resources>

写到这里，开篇提出的两个问题就解决完了。当然，更改pom文件，配置maven编译的文件的路径是一种方法。还有另外两种方法，也能够将UserMapper.xml编译那个位置。

在src/main/java下面新建一个包com.mapping，专门用来放xxMapper.xml文件。然后，将这个包标记为Resources.不过既然路径改变了，接口和映射文件对应的时候，路径也要改变，这里完全是mybatis的内容，就不详说了。



本以为编译后也会在com下面有个mapping文件夹，可惜并没有，而是直接在classes下面直接编译了userMapper.xml文件。所以这个时候UserMapper的文件路径应该是 “classpath\*：UserMapper.xml”,而不是 “classpath：com/mapping/ UserMapper.xml”.不过这样一来，就没办法配置自动扫描xxMapper.xml,而是需要手动配置引入xxMapper.xml。

在src/main/resources/下新建一个com文件夹，再在里面新建mapper文件夹（因为我的xxMappper.xml是在com.mapper包下），然后将xxMappper.xml移动到/ src/main/resources/com/mapper，这种情况下pom文件里就不需要再配置resources那串代码了，这样也是能编译的。

