# Mybatis Mapper文件和 接口配置

问题引出：

首先要说明的问题是，Mybatis中接口和对应的mapper文件不一定要放在同一个包下，放在一起的目的是为了Mybatis进行自动扫描，并且要**注意此时Java接口的名称和mapper文件的名称要相同，否则会报异常**，由于此时Mybatis会自动解析对应的接口和相应的配置文件，所以就不需要配置mapper文件的位置了。

# 1. 接口和文件在同一个包中

## 1.1 默认maven构建

如果在工程中使用了maven构建工具，那么就会出现一个问题：我们知道在典型的maven工程中，目录结构有：src/main/java和src/main/resources，前者是用来存放java源代码的，后者则是存放一些资源文件，比如配置文件等，在默认的情况下maven打包的时候，对于src/main/java目录只打包源代码，而不会打包其他文件。所以此时如果把对应的mapper文件放到src/main/java目录下时，不会打包到最终的jar文件夹中，也不会输出到target文件夹中，由于在进行单元[测试](http://lib.csdn.net/base/softwaretest" \o "软件测试知识库" \t "http://blog.csdn.net/lmy86263/article/details/_blank)的时候执行的是/target目录下/test-classes下的代码，所以在测试的时候也不会成功。

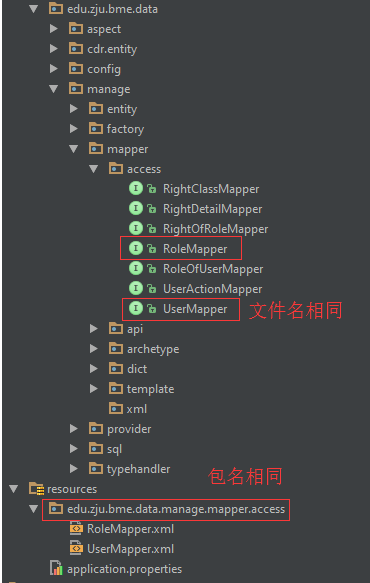
为了实现在maven默认环境下打包时，Mybatis的接口和mapper文件在同一包中，可以通过将接口文件放在src/main/java某个包中，而在src/main/resources目录中建立同样的包，这是一种约定优于配置的方式，这样在maven打包的时候就会将src/main/java和src/main/resources相同包下的文件合并到同一包中。

在默认maven打包的环境下，不要将接口文件和mapper文件全部放到src/main/java，这样也不会把mapper文件打包进去

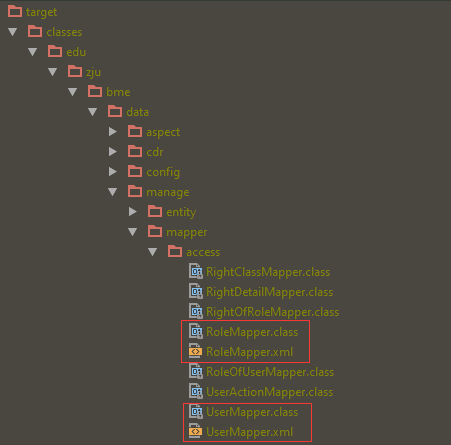
简要的过程如下（与如下的例子无关）：

|  |
| --- |
| **- src/main/java**  **- edu.zju.mapper**  **- UserMapper.java**  **- src/main/resources**  **- config.xml**  **- edu.zju.mapper**  **- UserMapper.xml**  如上这种方式在maven打包之后的目录如下：  **- src/main/java**  **- config.xml**  **- edu.zju.mapper**  **- UserMapper.java**  **- UserMapper.xml** |

具体的例子如下：

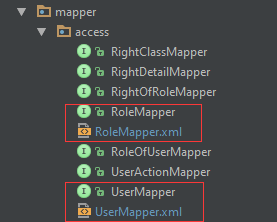


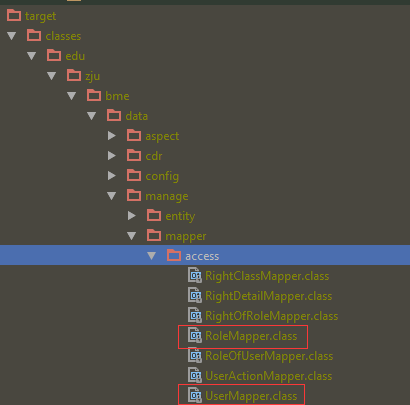
在src/main/java和src/main/resources中相同的包名，相同的文件名，默认打包后的结构如下：



发现打包之后已经到同一个包下了。

而当你把接口和mapper文件全部放到src/main/java中同一包时，返回结果如下：





发现编译打包之后并没有mapper文件，所以此时不管是测试还是正式执行都会出错！！

### 2.1 更改maven构建配置

如果不想将接口和mapper文件分别放到src/main/java和src/main/resources中，而是全部放到src/main/java，那么在构建的时候需要指定maven打包需要包括xml文件，具体配置如下：

<build>

<resources>

<resource>

<directory>src/main/java</directory>

<includes>

<include>\*\*/\*.xml</include>

</includes>

<filtering>false</filtering>

</resource>

</resources></build>

这样在打包的时候也会将mapper文件打包到/target文件夹中。

## 2. 接口和文件不在同一个包下

如果接口和mapper文件不在同一个包下，就不能进行自动扫描解析了，需要对接口和文件分别进行配置。

### 2.1 XML配置方式

#### 2.1.1 不使用Spring

使用Mybatis的配置文件如下：

|  |
| --- |
| <?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>  <mappers>  <!-- 扫描路径下的mapper映射文件 -->  <mapper resource="mappers/UserMapper.xml"/>  <!-- 扫描包下的接口文件 -->  <package name="edu.zju.bme.data.manage.mapper" />  </mappers>  </configuration> |

#### 2.1.2 使用Spring

使用[spring](http://lib.csdn.net/base/javaee" \o "Java EE知识库" \t "http://blog.csdn.net/lmy86263/article/details/_blank)的配置文件如下：

|  |
| --- |
| <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:mybatis="http://mybatis.org/schema/mybatis-spring"  xsi:schemaLocation="  http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd  http://mybatis.org/schema/mybatis-spring http://mybatis.org/schema/mybatis-spring.xsd">  <!-- 配置接口存储的包，用来扫描mapper接口 -->  <mybatis:scan base-package="edu.zju.bme.data.manage.mapper" />  <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">  <!-- 配置mapper文件位置，扫描映射文件，可以使用Ant风格的路径格式 -->  <property name="mapperLocations" value="classpath\*:mappers/\*\*/\*.xml" />  // ...  </bean></beans> |

### 2.2 Java 配置方式

#### 2.2.1 不使用Spring

Configuration configuration = new Configuration();

configuration.setMapUnderscoreToCamelCase(true);

configuration.setLazyLoadingEnabled(true);

configuration.setCacheEnabled(false);// 扫描该包下的接口和mapper文件

configuration.addMappers("edu.zju.bme.data.manage.mapper");

使用这种方法，只能将接口和mapper文件放到同一个包下，并且同名，相比其他方法具有局限性。

#### 2.2.2 使用Spring

// 配置类@Configuration(value = "manageConfig")@Import(value = {DataSourceConfig.class})// 扫描接口类，这个配置只能扫描该包下的接口，不能扫描mapper文件@MapperScan("edu.zju.bme.data.manage.mapper")public class ManageConfig {

@Autowired

private DataSourceConfig dataSourceConfig;

@Bean(name = "manageSessionBean")

public SqlSessionFactoryBean sqlSessionFactoryBean() throws IOException{

SqlSessionFactoryBean factoryBean = new SqlSessionFactoryBean();

factoryBean.setDataSource(dataSourceConfig.manageDataSource());

Configuration configuration = new Configuration();

// 扫描对应的mapper文件

factoryBean.setMapperLocations(new Resource[]{new ClassPathResource("UserMapper.xml")});

factoryBean.setConfiguration(configuration);

return factoryBean;

}

}

上面总结了4种配置方式，包括使用Spring以及不使用Spring的环境下，但是要注意以上的配置方式并不是唯一的，还有其他方法，如配置org.mybatis.spring.mapper.MapperScannerConfigurer，这里只是选择了相对来说较为简单的方式。