链滴

【Mybatis】如何继承 Mybatis 中的 Mapper.xml 文件

作者：shirenchuang

原文链接：https://ld246.com/article/1565846915140

来源网站：链滴

最近在写一个 Mybatis 代码自动生成插件，用的是Mybatis来扩展，其中有一个需求就是 生成javaMpper文件和 xmlMapper文件的时候 希望另外生成一个扩展类和扩展xml文件。原文件不修改，只存一些基本的信息，开发过程中只修改扩展的Ext文件

形式如下：

SrcTestMapper.java

```
package com.test.dao.mapper.srctest;

import com.test.dao.model.srctest.SrcTest;
import com.test.dao.model.srctest.SrcTestExample;
import java.util.List;
import org.apache.ibatis.annotations.Param;

public interface SrcTestMapper {
    long countByExample(SrcTestExample example);

    int deleteByExample(SrcTestExample example);

    int deleteByPrimaryKey(Integer id);
```

```
    int insert(SrcTest record);

    int insertSelective(SrcTest record);

    List<SrcTest> selectByExample(SrcTestExample example);

    SrcTest selectByPrimaryKey(Integer id);

    int updateByExampleSelective(@Param("record") SrcTest record, @Param("example") SrcTe
tExample example);

    int updateByExample(@Param("record") SrcTest record, @Param("example") SrcTestExampl
 example);

    int updateByPrimaryKeySelective(SrcTest record);

    int updateByPrimaryKey(SrcTest record);
}
```

SrcTestMapperExt.java

```
package com.test.dao.mapper.srctest;

import com.test.dao.model.srctest.SrcTest;
import org.apache.ibatis.annotations.Param;

import javax.annotation.Resource;
import java.util.List;


/**
* SrcTestMapperExt接口
* Created by shirenchuang on 2018/6/30.
*/
@Resource
public interface SrcTestMapperExt extends SrcTestMapper {


    List<SrcTest> selectExtTest(@Param("age") int  age);

}
```

SrcTestMapper.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/
ybatis-3-mapper.dtd">
<mapper namespace="com.test.dao.mapper.srctest.SrcTestMapperExt">
  <resultMap id="BaseResultMap" type="com.test.dao.model.srctest.SrcTest">
```

```
    <id column="id" jdbcType="INTEGER" property="id" />
    <result column="name" jdbcType="VARCHAR" property="name" />
    <result column="age" jdbcType="INTEGER" property="age" />
    <result column="ctime" jdbcType="BIGINT" property="ctime" />
  </resultMap>
<!-- 省略....-->
</mapper>
```

SrcTestMapperExt.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/
ybatis-3-mapper.dtd">
<mapper namespace="com.test.dao.mapper.srctest.SrcTestMapperExt">
    <select id="selectExtTest" resultMap="BaseResultMap">
        select * from src_test where age>#{age}
    </select>

</mapper>
```

注意：这里返回的resultMap="BaseResultMap" 这个Map并没有再这个xml中定义，这样能使用?

上面是我生成的代码;并且能够正常使用;

那么SrcTestMapperExt.xml是如何继承SrcTestMapper.xml中的定义的呢?

1. 修改命名空间，使他们的命名空间相同，namespace="com.test dao.mapper.srctest.SrcTestMapperExt"

---

2. 光这样还不够，因为这个时候你去运行的时候会报错

```
Caused by: org.apache.ibatis.builder.BuilderException: Wrong namespace. Expected 'com.test
dao.mapper.srctest.SrcTestMapper' but found 'com.test.dao.mapper.srctest.SrcTestMapperExt'
```

因为Mybatis中是必须要 xml的文件包名和文件名必须跟 Mapper.java对应起来的

比如com.test.dao.mapper.srctest.SrcTestMapper.java这个相对应的是

com.test.dao.mapper.srctest.SrcTestMapper.xml

必须是这样子，没有例外，否则就会报错

show the code

MapperBuilderAssistant

```
  public void setCurrentNamespace(String currentNamespace) {
    if (currentNamespace == null) {
```

```
      throw new BuilderException("The mapper element requires a namespace attribute to be s
ecified.");
    }

    if (this.currentNamespace != null && !this.currentNamespace.equals(currentNamespace)) {
      throw new BuilderException("Wrong namespace. Expected '"
          + this.currentNamespace + "' but found '" + currentNamespace + "'.");
    }

    this.currentNamespace = currentNamespace;
  }
```

这个this.currentNamespace 和参数传进来的currentNamespace比较是否相等；

参数传进来的currentNamespace就是我们xml中的

<mapper namespace="com.test.dao.mapper.srctest.SrcTestMapperExt">值；

然后this.currentNamespace是从哪里设置的呢？this.currentNamespace = currentNamespace;

跟下代码：**MapperAnnotationBuilder**

```
  public void parse() {
    String resource = type.toString();
    if (!configuration.isResourceLoaded(resource)) {
      loadXmlResource();
      configuration.addLoadedResource(resource);
      assistant.setCurrentNamespace(type.getName());
      parseCache();
      parseCacheRef();
      Method[] methods = type.getMethods();
      for (Method method : methods) {
        try {
          // issue #237
          if (!method.isBridge()) {
            parseStatement(method);
          }
        } catch (IncompleteElementException e) {
          configuration.addIncompleteMethod(new MethodResolver(this, method));
        }
      }
    }
    parsePendingMethods();
  }
```

看到　　assistant.setCurrentNamespace(type.getName());

它获取的是 type.getName()；这个type的最终来源是

MapperFactoryBean<T>

```
  @Override
  protected void checkDaoConfig() {
    super.checkDaoConfig();

    notNull(this.mapperInterface, "Property 'mapperInterface' is required");

    Configuration configuration = getSqlSession().getConfiguration();
```

```
    if (this.addToConfig && !configuration.hasMapper(this.mapperInterface)) {
      try {
        configuration.addMapper(this.mapperInterface);
      } catch (Exception e) {
        logger.error("Error while adding the mapper '" + this.mapperInterface + "' to configurati
n.", e);
        throw new IllegalArgumentException(e);
      } finally {
        ErrorContext.instance().reset();
      }
    }
  }
```

看configuration.addMapper(this.mapperInterface);这行应该就明白了

加载mapperInterface的时候会跟相应的xml映射，并且会去检验namespace是否跟mapperInterfac相等！

那么既然命名空间不能修改,那第一条不白说了？还怎么实现Mapper.xml的继承啊？

别慌，既然是这样子，那我们可以让 **MapperInterface** 中的**SrcTestMapper.java**别被加载进来就了啊！！

只加载 MapperExt.java不就行了？

3. 修改applicationContext.xml，让Mapper.java不被扫描

Mapper.java接口扫描配置

```
    <!-- Mapper接口所在包名，Spring会自动查找其下的Mapper -->
    <bean id="mapperScannerConfigurer" class="org.mybatis.spring.mapper.MapperScanner
onfigurer">
        <property name="basePackage" value="com.test.dao.mapper"/>
<!--
        该属性实际上就是起到一个过滤的作用，如果设置了该属性，那么MyBatis的接口只有包含该注
，才会被扫描进去。
-->
        <property name="annotationClass" value="javax.annotation.Resource"/>
        <property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"/>
    </bean>
```

basePackage 把Mapper.java扫描进去没有关系，重点是

<property name="annotationClass" value="javax.annotation.Resource"/>

这样 MapperScanner会把没有配置注解的过滤掉;

回头看我们的MapperExt.java配置文件是有加上注解的

```
/**
* SrcTestMapperExt接口
* Created by shirenchuang on 2018/6/30.
*/
@Resource
public interface SrcTestMapperExt extends SrcTestMapper {
```

```
    List<SrcTest> selectExtTest(@Param("age") int  age);
}
```

这样子之后，基本上问题就解决了，还有一个地方特别要注意一下的是.xml文件的配置

```
    <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
        <property name="dataSource" ref="dataSource"/>
        <property name="configLocation" value="classpath:mybatis-config.xml"/>
        <!-- 必须将mapper,和mapperExt也一起扫描-->
        <property name="mapperLocations" value="classpath:com/test/dao/mapper/**/*.xml"/

    </bean>
```

这样配置没有错，但是我之前的配置写成了

```
<property name="mapperLocations" value="classpath:com/test/dao/mapper/**/*Mapper.xm
"/>
```

这样子 MapperExt.xml 没有被扫描进去，在我执行单元测试的时候

```
  @Test
    public void selectExt(){
        List<SrcTest> tests = srcTestService.selectExtTest(9);
        System.out.println(tests.toString());
    }
```

err_console

```
org.apache.ibatis.binding.BindingException: Invalid bound statement (not found): com.test.da
.mapper.srctest.SrcTestMapperExt.selectExtTest
```

但是执行 ````srcTestService.insertSelective(srcTest);不会出错 原因就是 insertSelective是在SrcTes Mapper.xml中存在 ， 已经被注册到 com.test.dao.mapper.srctest.SrcTestMapperExt```命名空间 ，但是selectExtTest由于没有被注册，所以报错了；

有兴趣可以下载阅读或者直接使用我整合的

自动生成扩展插件