

# 如何批量测试Mybatis项目中SQL是否正确

Java知音 昨天

点击上方“Java知音”，选择“置顶公众号”  
技术文章第一时间送达！

作者：不学无数的程序员  
juejin.im/post/5c0108fdf265da613e21f997

## 去Oracle行动

最近公司要发展海外项目，所以要将现有的系统全部平移过去，另外数据库也要从原来的Oracle变为Mysql。公司的数据库交互层面使用的是Mybatis，而Oracle与Mysql也有一些语法上的不同。所以在项目中的Sql要改动，但是多个项目中涉及到的Sql非常多，如果仅凭人工一条一条辨别的话，工作量有点大。

所以就萌发出了直接将数据源变为Mysql，利用反射批量执行Mapper中的方法，然后如果有参数的话，就设置为默认的初始值，然后记录下来成功的数据和失败的数据，这样就可以根据失败原因进行修改。能够节省很大的时间。

## 执行效果

错误标识	表名+方法名	错误信息
Error:	AisQueryMapper.getaisErrorStmCount	FUNCTION ais.to_char does not exist
Error:	AisQueryMapper.getSbjInfoBySbjNo	FUNCTION ais.to_char does not exist
Error:	GlVochDMapper.querySbjBIList	Every derived table must have its own alias
Error:	GlVochDMapper.querySbjBIDetailSumList	Table ais.t_ais_gl_dtl_d_sum doesn't exist
Error:	GlVochDMapper.updateDtIDSum	Table ais.t_ais_gl_dtl_d_sum doesn't exist
Error:	GlVochDMapper.insertDtIDSumNotTz	Table ais.t_ais_gl_dtl_d_sum doesn't exist
Error:	GlVochDMapper.insertDtIDSumTz	Table ais.t_ais_gl_dtl_d_sum doesn't exist

## 代码介绍

## 总体思路就三步

- 通过反射获得要执行的Mapper类的所有方法
- 获得方法中的参数，并赋值
- 执行

```
AutoTestMapper autoTestMapper = new AutoTestMapper("存放Mapper全路径名");
autoTestMapper.openSqlSession(sqlSessionFactory);
```

在构造函数中传入全路径名后，进行解析，解析出包名和所有的文件名并存储起来

```
public AutoTestMapper(String path) throws IOException, ClassNotFoundException {
    String mapperContent = getFileContent(path);
    String pathPattern = "import [a-z,A-Z,/.]+;";
    String[] pathArr = matchMethod(pathPattern, mapperContent).split(";");
    for (int i = 0; i < pathArr.length; i++) {
        pathArr[i] = pathArr[i].replaceAll("import ", "");
        Class cls = Class.forName(pathArr[i]);
        if (!cls.isInterface()) {
            TYPE_ARRAY.add(cls);
        }
    }
    //获得全路径名的前缀
    String packPattern = "package [a-z,A-Z,/.]+;";
    String[] packPathArr = matchMethod(packPattern, mapperContent).split(";");
    String packPath = packPathArr[0].replaceAll("package ", "").replaceAll(";", "");
    this.PACK_PATH = packPath;
}
```

然后调用openSqlSession的方法，传入SqlSessionFactory参数

```
List<Map<Class, Object>> list = new ArrayList<>();
List<String> invokeSuccess = new ArrayList<>();
List<String> invokeFail = new ArrayList<>();
for (String fileName : FILE_NAME) {
    Class cls = Class.forName(PACK_PATH + "." + fileName);
    //添加Mapper
    if (!sqlSessionFactory.getConfiguration().hasMapper(cls)){
        sqlSessionFactory.getConfiguration().addMapper(cls);
    }
    //获得Mapper
    Object mapper = sqlSessionFactory.openSession().getMapper(cls);
    //反射执行Mapper的方法
    Map<String, List<String>> resultMap = autoTestInvoke(cls, mapper);
    invokeSuccess.addAll(resultMap.get(SUCCESS_FLG));
    invokeFail.addAll(resultMap.get(FAIL_FLG));
}
```

然后通过Mybatis提供的方法getMapper()传入类名获得所要Mapper类。核心方法就是autoTestInvoke()方法了

```
private Map<String, List<String>> autoTestInvoke(Class c, Object o)
{
    Method[] declaredMethods = c.getDeclaredMethods();
```

```

String fileName = c.getName().substring(c.getName().lastIndexOf("."));
List<String> invokeSuccess = new ArrayList<>();
List<String> invokeFail = new ArrayList<>();
Map<String, List<String>> resultMap = new HashMap<>();
//给参数赋初始值
for (Method method : declaredMethods) {
    List<Object> list = new ArrayList<>();
    for (Class cls : method.getParameterTypes()) {
        Object par = new Object();
        if (TYPE_ARRAY.contains(cls)) {
            if (cls.equals(String.class)) {
                par = "1";
            } else {
                try {
                    par = cls.newInstance();
                    assignment(cls, par);
                } catch (InstantiationException e) {
                    if (cls.isPrimitive()) {
                        cls = primitiveClazz.get(cls.getName());
                    }
                }
                try {
                    par = cls.getDeclaredConstructor(String.class).newInstance("1");
                } catch (NoSuchMethodException e1) {
                    System.out.println(cls.getName()+e);
                }
            }
        }
        } else if ("java.util.Map".equals(cls.getName())) {
            par = getMapData(c.getName()+"."+method.getName());
        }
        list.add(par);
    }
    try {
        method.invoke(o, list.toArray());
        invokeSuccess.add("Success: " + fileName + "." + method.getName());
    } catch (Exception e) {
        invokeFail.add("Error:" + method.getName() + " Error Info:" + e);
    }
}
resultMap.put(SUCCESS_FLG, invokeSuccess);
resultMap.put(FAIL_FLG, invokeFail);
return resultMap;
}

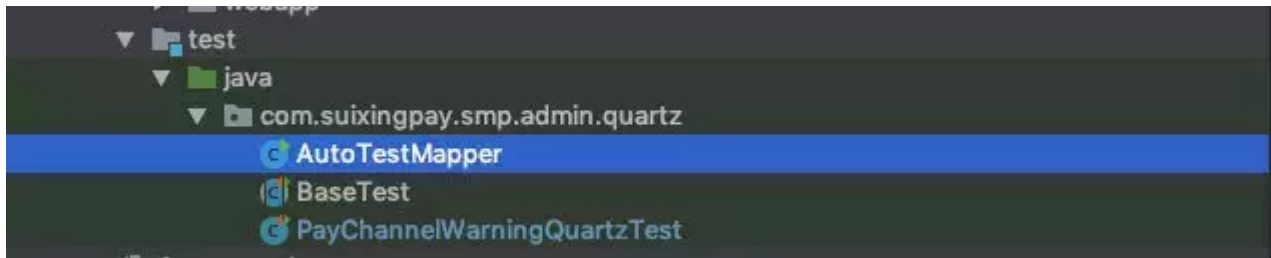
```

这里面完成为参数赋初始值，和执行的逻辑。

## 使用说明

自动测试Mapper除了传参为List和Set，其余都能测到。在xml中所有的if条件都会拼接到。

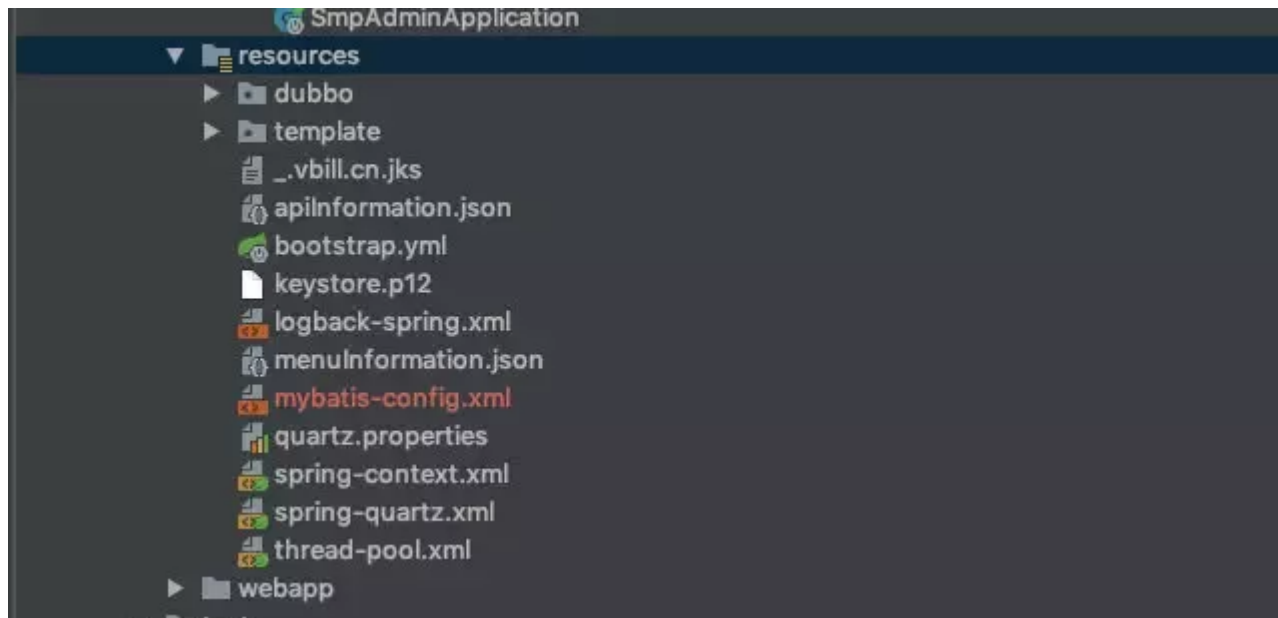
**将AutoTestMapper拷贝到测试模块中。如图所示**



AutoTestMapper文件存放在github

<https://github.com/modouxiansheng/convenientUtil/blob/master/src/main/java/com/github/autoTest/AutoTestMapper.java>

在resources模块中加入mybatis-config.xml文件，如图所示



mybatis-config.xml内容如下

```
<!DOCTYPE configuration
PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
  <environments default="dev">
    <environment id="dev">
      <transactionManager type="JDBC"></transactionManager>
      <dataSource type="POOLED">
        <property name="driver" value="com.mysql.jdbc.Driver"/>
        <property name="url" value="连接地址"/>
        <property name="username" value="账号"/>
        <property name="password" value="密码"/>
      </dataSource>
    </environment>
  </environments>
</configuration>
```

## 在根目录创建lib文件夹，并将测试的Mybatis版本放入其中，并在Gradle中引入此包

```
compile files('..lib/mybatis-3.5.0-hupengfeiTest.jar')此处路径填写相对路径
```

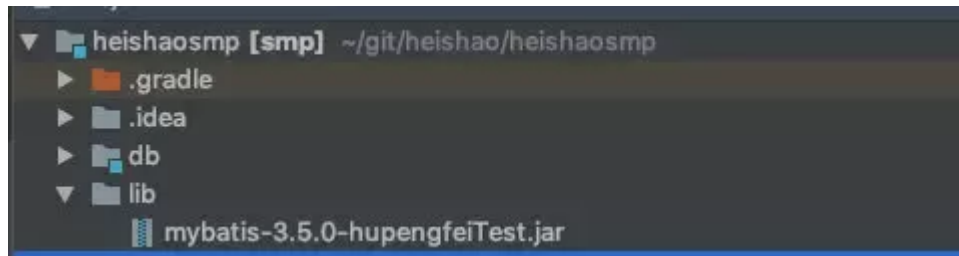
如果目录结构如下，那么就compile files('lib/mybatis-3.5.0-hupengfeiTest.jar')

mybatis-3.5.0-hupengfeiTest.jar在github下面的lib目录中

```
-lib
-- mybatis-3.5.0-hupengfeiTest.jar
-build.gradle
```

如果目录结构如下，那么就compile files('..lib/mybatis-3.5.0-hupengfeiTest.jar')

```
-lib
-- mybatis-3.5.0-hupengfeiTest.jar
-service
-- build.gradle
```



## 在单元测试中编写代码，进行测试

```
@RunWith(SpringJUnit4ClassRunner.class)
@SpringBootTest(classes = { AirApplication.class })//此处AirApplication.class为项目中的启动类，自行修改
public class TestMapper {

    @Test
    public void testCeshi()
        throws IllegalAccessException, IntrospectionException, InvocationTargetException, NoSuchMethodException, InstantiationException, IOException, ClassNotFoundException {
        //读取Mybatis配置
        Reader resourceAsReader = Resources.getResourceAsReader("mybatis-config.xml");
        //生成SqlSessionFactory
        SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(resourceAsReader);
        resourceAsReader.close();
        AutoTestMapper autoTestMapper = new AutoTestMapper(存放Mapper的Java文件夹的全路径名);
        //执行测试方法
        autoTestMapper.openSqlSession(sqlSessionFactory);
    }
}
```

就会在控制台中打印出执行失败的Mapper以及其原因。如下图所示

```
Success: RunUpActMapper.getBytranSno  
Success: AbnoMapper.update  
Success: AbnoMapper.queryAbno  
Success: AbnoMapper.getLast  
-----  
[Error: |ActErrorMapper.nextSNO| Unknown table 'ais.SEQ_ACT_ERROR' in field list|  
[Error: |GLDtlMapper.getMatterList| You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for t  
[Error: |GLDtlMapper.nextVchCdCode| Unknown table 'ais.SEQ_T_AIS_GL_DTL_D' in field list|  
[Error: |RunUpActMapper.nextSno| Unknown table 'ais.SEQ_T_AIS_RUN_UP_ACT' in field list|  
[Error: |AbnoMapper.insert| java.sql.SQLException: Incorrect integer value: 'b95192b4-a9d2-4f10-bf46-c8f890' for column 'uuid' at row 1|
```

## github地址

<https://github.com/modouxiansheng/convenientUtil>

### 推荐阅读(点击即可跳转阅读)

1. [SpringBoot内容聚合](#)
2. [面试题内容聚合](#)
3. [设计模式内容聚合](#)
4. [Mybatis内容聚合](#)
5. [多线程内容聚合](#)

觉得不错？ 欢迎转发分享给更多人



Java知音

微信扫描二维码，关注我的公众号

我知道你 “在看”

