

作者：杨旭东

时间：8/14/2017 9:50:26 PM

描述

构建场景：软件开发过程中，当谈论起mybatis，脑海里第一个念头就是需要写众多的sql语句，作为半自动ORM框架的mybatis，能否有自动生成sql语句和其对应的Mpper接口以及pojo类的工具呢？福利来了，哈哈，那就是下面要说的逆向工程。

逆向工程：mabatis官方为了减轻程序员的压力，提供了一套自动生成sql语句和其对应的Mpper接口以及pojo类的工具。但是！生成的sql语句自限于单表的增删改查，联合操作需要程序员手动书写sql语句以及对应的Mapper接口。

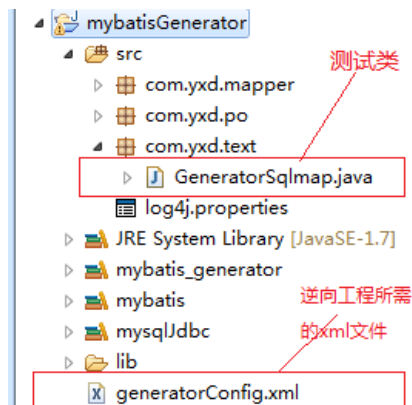
所需jar包：

1. 数据库jdbc的jar（+mysql-connector-java-5.1.42-bin.jar）
2. 逆向工程工具jar（+mybatis-generator-core-1.3.3.jar
3. mybatis的jar(ognl-3.1.12.jar[ognl所需jar包，不知道的自行百度或者mybatis我会提到]，mybatis-3.4.2.jar[mybatis的主jar包])[可以不用，强迫症的人可以用上，不用的情况下mpper接口文件会报错]
4. log4j的jar（+log4j-1.2.17.jar）[生产过程可以不用日志文件，但是推荐使用]

补充——> +：表示案例中使用的jar

逆向工程工具下载网址：<https://github.com/mybatis/generator/releases>

操作



generatorConfig.xml文件

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE generatorConfiguration PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
"http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
<generatorConfiguration>
  <!-- id此上下文的唯一标识符。此值将用于一些错误消息中。（自定义）targetRuntime：查文档（MyBatis3为默认，同时我自认为是最优设置）-->
  <context id="testTables" targetRuntime="MyBatis3">
    <commentGenerator>
      <!-- 是否去除自动生成的注释 true：是 false：否 -->
      <property name="suppressAllComments" value="true" />
    </commentGenerator>
    <!-- 数据库连接的信息：驱动类、连接地址、用户名、密码 -->
    <jdbcConnection driverClass="com.mysql.jdbc.Driver"
      connectionURL="jdbc:mysql://localhost:3306/demo?useUnicode=true&characterEncoding=utf-8&useSSL=false"
      userId="root" password="root">
    </jdbcConnection>
    <jdbcTypeResolver>
      <property name="forceBigDecimals" value="false" />
    </jdbcTypeResolver>
    <!-- targetProject:生成PO类的位置 -->
```

```

<javaModelGenerator targetPackage="com.yxd.po"
    targetProject=".\\src">
    <!--enableSubPackages:是否让schema作为包的后缀 -->
    <property name="enableSubPackages" value="false" />
    <!-- 从数据库返回的值被清理前后的空格 -->
    <property name="trimStrings" value="true" />
</javaModelGenerator>
<!-- targetProject:mapper映射文件生成的位置 -->
<sqlMapGenerator targetPackage="com.yxd.mapper"
    targetProject=".\\src">
    <!--enableSubPackages:是否让schema作为包的后缀 -->
    <property name="enableSubPackages" value="false" />
</sqlMapGenerator>
<!-- targetPackage: mapper接口生成的位置 -->
<javaClientGenerator type="XMLMAPPER"
    targetPackage="com.yxd.mapper" targetProject=".\\src">
    <!--enableSubPackages:是否让schema作为包的后缀 -->
    <property name="enableSubPackages" value="false" />
</javaClientGenerator>
<!-- 指定数据库表 -->
<table tableName="order"></table>
<table tableName="user"></table>
</context>
</generatorConfiguration>

```

GeneratorSqlmap.java文件

```

/**
 * @Title: GeneratorSqlmap.java
 * @Package mybatisGenerator
 * @Description:逆向工程测试
 * @author 杨旭东
 * @date 2017年8月14日 下午8:02:24
 * @version V1.0
 */
public class GeneratorSqlmap {
    //该类重点知道generatorConfig.xml引即可，如果xml文件放置错误，下面引入就不能正确引入，放置位置看上图。
    public void generator()throws Exception{

        List<String>warnings = new ArrayList<String>();

        boolean overwrite =true;

        File configFile = new File("generatorConfig.xml");

        ConfigurationParser cp = new ConfigurationParser(warnings);

        Configuration config = cp.parseConfiguration(configFile);

        DefaultShellCallback callback = new DefaultShellCallback(overwrite);

        MyBatisGenerator myBatisGenerator = new MyBatisGenerator(config,

callback,warnings);

        myBatisGenerator.generate(null);

    }
    public static void main(String[] args) throws Exception {

        try {

            GeneratorSqlmap generatorSqlmap = new GeneratorSqlmap();

            generatorSqlmap.generator();


```

```
        } catch (Exception e) {  
  
            e.printStackTrace();  
  
        }  
    }  
}
```

log4j.properties文件

```
# 注意：设置成debug级别的，发布后设置成error,stdout为未来日志的目的地名称  
log4j.rootLogger=DEBUG, stdout  
lo  
# Console控制台设置  
log4j.appender.stdout=org.apache.log4j.ConsoleAppender  
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout  
# %l:添加该属性，可以在日志中看出该条日志在哪里发生的 （一般不用）  
log4j.appender.stdout.layout.ConversionPattern= %d{yyyy MMM dd HH:mm:ss} %p [%t] - %m%n
```

总结

在主函数中跑起来，不出错的前提下，refresh项目，就可在项目下看见对于得pojo包和mapper包。最后将得到的包拷到将来的web项目中。逆向工程结束！

补充：

- 工具生成的updateByPrimaryKeySelective和updateByPrimaryKey方法的区别：
 1. updateByPrimaryKeySelective会对字段进行判断再更新(如果为Null就忽略更新)，如果你只想更新某一字段，可以用这个方法。
 2. updateByPrimaryKey对你注入的字段全部更新。
- 有问题看文档
 - 例：mybatis-generator-core-1.3.3\docs下有index.html入口文件。

——>End(完)