考虑到通用性，实现的方式是用java代码启动的。

配置文件：

```
<?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>

    <context id="DB2Tables" targetRuntime="MyBatis3">

        <!-- 去除自动生成的注释 -->
        <commentGenerator>
            <property name="suppressDate" value="true" />
            <property name="suppressAllComments" value="true" />
        </commentGenerator>

        <!-- 连接数据库 -->
        <jdbcConnection driverClass="com.mysql.jdbc.Driver"
            connectionURL="jdbc:mysql://127.0.0.1:3306/myhome?
useUnicode=true&amp;characterEncoding=utf-8"
            userId="root" password="123456">
        </jdbcConnection>

        <!-- 默认false，把JDBC DECIMAL 和 NUMERIC 类型解析为 Integer true， 把 JDBC
DECIMAL
            和 NUMERIC 类型解析为java.math.BigDecimal -->
        <javaTypeResolver>
            <property name="forceBigDecimals" value="false" />
        </javaTypeResolver>

        <!-- 实体类的生成信息 -->
        <javaModelGenerator targetPackage="com.xyz.domain"
            targetProject="src/main/java">
            <property name="enableSubPackages" value="true" />
            <property name="trimStrings" value="true" />
        </javaModelGenerator>

        <!-- sqlmappper xml 文件的生成信息 -->
        <sqlMapGenerator targetPackage="com.xyz.dao"
            targetProject="src/main/resource">
            <property name="enableSubPackages" value="true" />
        </sqlMapGenerator>

        <!-- 接口的生成信息 -->
        <javaClientGenerator type="XMLMAPPER"
            targetPackage="com.xyz.dao" targetProject="src/main/java">
            <property name="enableSubPackages" value="true" />
        </javaClientGenerator>
		<!-- 表的信息 -->
		<table tableName="t_entity_parameter" domainObjectName="EntityParameter"
			enableCountByExample="true" enableUpdateByExample="true"
			enableDeleteByExample="true" enableSelectByExample="true"
			selectByExampleQueryId="true"></table>

	</context>
</generatorConfiguration>
```

java代码：

```
package com.xyz.util;

import java.io.File;
import java.io.IOException;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import org.mybatis.generator.api.MyBatisGenerator;
import org.mybatis.generator.config.Configuration;
import org.mybatis.generator.config.xml.ConfigurationParser;
import org.mybatis.generator.exception.InvalidConfigurationException;
import org.mybatis.generator.exception.XMLParserException;
import org.mybatis.generator.internal.DefaultShellCallback;

/**
 * mybatis-generator Java启动方式
 *
 * @author xiaoyezi
 *
 */
public final class MybatisGeneratorUtil {

	/**
	 * 开始自动生成
	 */
	public static void gen() {
		try {
			System.out.println("开始自动生成.......");
			List<String> warnings = new ArrayList<String>();
			boolean overwrite = true;
			File configFile = new
File(MybatisGeneratorUtil.class.getResource("generatorConfig.xml").getFile());
			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);
			System.out.println("结束自动生成.......");
		} catch (IOException e) {
```

```
            e.printStackTrace();
        } catch (XMLParserException e) {
            e.printStackTrace();
        } catch (InvalidConfigurationException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        MybatisGeneratorUtil.gen();
    }

}
```

码云项目路径：https://gitee.com/yelizi/mybatisGen.git