

# MyBatis-CMEU 使用说明

MyBatis-CMEU 全称为:MyBatis Config Mapper Entity Util

官方 QQ 交流群: 519714660;群里提供各平台无 JDK 要求版

简单来说 CMEU 是一款使用 Javafx 开发的 MyBatis 帮助工具,使用该工具你只需要连接数据库,选择你需要的表(单表/多表),点击执行创建,该工具就会帮你生成实体类以及接口与 Mapper 的增删改查,Assist 帮助类让分页变得像 hibernate 一样简单,同时也简化并提供了动态的条件查询;

温馨提示:不要太依赖 CMEU 因为它有可能使你忘记 mapper 怎么写!

MyBatis-Config-Mapper-Entity-Util

数据库连接

配置信息

orcl

root

sqlserver

项目所在目录: 请输入或选择项目所在目录,例如:E:\JavaFX

选择

根目录: src/

表名: 双击左侧表名加载

实体类包名: com.entity

类名:

修改属性

接口包名: com.dao

接口名:

映射文件包名: com.dao

映射文件名:

查询工具包名: com.common

工具名: Assist

配置文件包名: resource

文件名: mybatis-config.xml

帮助工具包名: com.util

工具名: MyBatisUtil

☒ 创建查询工具

☐ 创建配置文件

☐ 创建帮助工具

执行创建

保存配置

QQ交流群:519714660

[关于数据库架包:]

系统默认集成的版本分别是:

Oracle=Oracle6.jar







SqlServer=sqljdbc2008.jar

MySql=mysql-connector-java-5.1.38.jar

PostgreSql=postgresql-9.4.1209.jar

如果跟你的版本不一样到 libs 里面改成你的架包,但是命名规则为:数据库小写 JBDC 大写

示例:sqlserverJDBC.jar

 JSONUtil-1.2.7.jar	2017/3/29 7:25	Executable Jar File	408 KB
 mysqlJDBC.jar	2017/3/29 7:25	Executable Jar File	961 KB
 oracleJDBC.jar	2003/1/5 17:37	Executable Jar File	2,062 KB
 postgresqlJDBC.jar	2017/3/29 7:25	Executable Jar File	674 KB
 sqliteJDBC.jar	2017/3/29 7:25	Executable Jar File	5,445 KB
 sqlserverJDBC.jar	2017/3/29 7:25	Executable Jar File	570 KB

工具与 Assist 使用说明与注意事项:

[基本上所有带选择或者不太明确的将鼠标移动到上面停留几秒就会出现相应的提示]

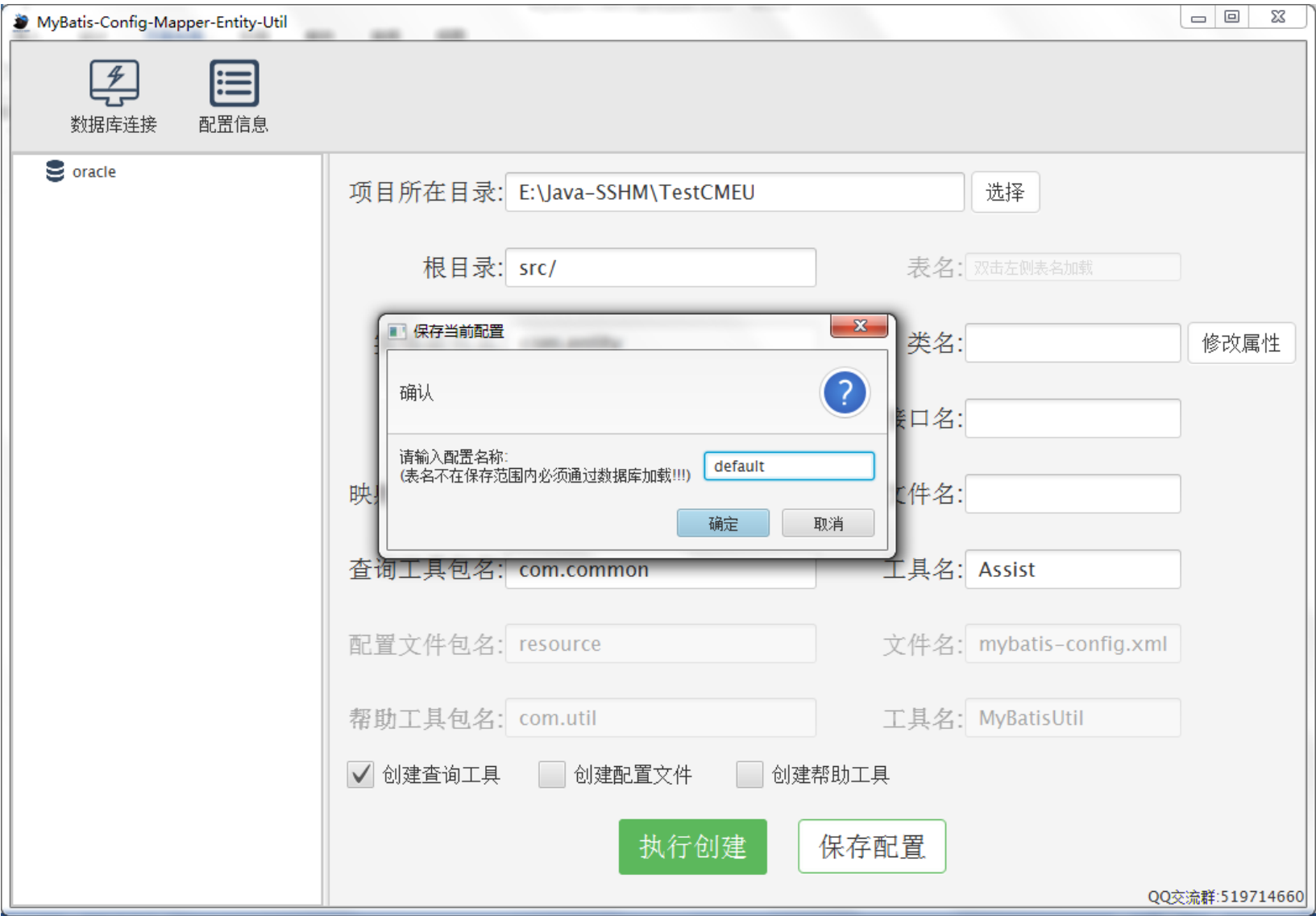
1. 打开数据库连接.接通你的数据库
2. 双击打开数据库并双击选择表
3. 选择你的项目所在路径
4. 系统默认将属性设置为驼峰命名,你可以点击修改属性进行数据的修改,或者添加子表
5. [重点]如果是多表首页选择主表(也就是少的一方比如一对多的一),进入修改属性添加子表  
[特别注意:表关联是以主表的主键作为关联,如果子表的外键跟主表名字不一样需要生成后自己修改子表列];
6. 名字不合适的话修改成合适的名字;
7. 保存常用的配置-->系统默认会读取 default,所以如果是常用的名字可以保存为 default
8. 执行创建!

1: 打开数据库连接--->接通你的数据库

连接名称是自己定义,sqlserver 与 Oracle 无视编码格式,除了密码外别的都为必填



2: 可以将常用的保存起来,下一次可以从配置信息读取设置







## Assist 的使用以及介绍:

[[需要特别注意的是因为是采用动态查询所以没有办法使用预编译,Assist 采用的是屏蔽特殊字体进行数据校验,相应的方法为:valiSQL(str),所以添加条件使用 setRequire 条件集添加]]

- 1. distinct 相应 set true 与 false
- 2. order 相应的 set true 为 asc,false 为 desc
- 3. startRow 相应的 set 方法 从第几行开始取数据 (示例:id>0)
- 4. endRow 相应的 set 方法 取数据到第几行结束 (示例:id<=5)
- 5.rowSize 相应的 set 方法 每次取多少行数据

[这 3 个分页属性一般分为两类数据库分页,比如典型的 Oracle rownum>0 and rownum<=5 这个时候就是用 startRow 跟 endRow,另外一种类型就是 MySql 的 limit 5 跟 offset 0,这时候就是使用 rowSize 跟 startRow;帮助类也是以这个方式进行]

- 6.require 相应的 set 方法为 setRequires 条件集 Assist 提供相应的静态 and 与 or 方法,因为动态查询无法预编译所以使用静态方法添加,才有进行数据校验;

```
9 public class Assist {
10     // 去重
11     private String distinct;
12     // 自定义排序
13     private String order;
14     // 分页开始行
15     private Integer startRow;
16     // 分页结束行 Oracle/SqlServer
17     private Integer endRow;
18     // 分页取多少行数据 Mysql/PostgreSql
19     private Integer rowSize;
20     // 条件集
21     private List<String> require = null;
22
23     /**
24      * 该方法用于使用Assist的静态条件方法,动态添加条件
25      *
26      * @param require 示例:Assist.and_lt(a,10),...
27      */
28     public void setRequires(String... require) {
29         if (this.require==null) {
```



## 使用实例:

```
DeptDao dao=MyBatisUtil.getSession().getMapper(DeptDao.class);
//如果并不需要Assist直接使用null就可以了
long count=dao.getDeptRowCount(null);
Assist assist=new Assist();
assist.setDistinct(true);//去重
//以id asc排序, 需要注意一点的就是id一般是类名加属性名
assist.setOrder("id", true);
//sqlserver与Oracle的分页
assist.setStartRow(0);//从第三行开始取
assist.setEndRow(10);//到第6行结束
//MySQL与PostgreSql的分页
assist.setRowSize(10);//每次取10行
assist.setStartRow(10);//从第10行开始取
//添加条件集
assist.setRequires(Assist.and_gte("id", "10"),Assist.);
//执行查询
dao.selectDept(assist);
```

