**MyBatisGenerator使用说明**

@author Anderson Mao, 2015-03-25, 2015-07-01

## # 基本情况

MyBatisGenerator 使用从官网下载的mybatis-generator-core-1.3.2-bundle.zip为基础。  
<https://github.com/mybatis/generator/releases>

根据公司的需求做了定制。定制的代码在：mybatis-generator-core-1.3.2-patch.jar（包含Java源文件）

## # 主要功能

* 会生成Example, 并以"Condition.java"结尾。
* 支持分页查询，分页查询函数为DOCondition.java中的基类：   
  com.netfinworks.restx.persist.jdbc.**QueryConditionBase**
* 支持字符串大小写不敏感查询例如：  
  DOCondition.createCriteria().andFieldLikeInsensitive(value)
* 可以配置字段生成的类型，比如把某些decimal字段生成Money类型。或者把某些字段生成枚举类型。

## # 生成步骤

1. 配置generatorConfig.xml里面的jdbcConnection数据源。

2. 更改：generatorConfig.xml

修改<sqlMapGenerator>的targetPackage为”ps”, “ls”等，建议把同类的table放在一个generatorConfig.xml文件里面，可以同时生成多个表的mapping。

修改或者增长<table>里面的tableName, domainObjectName

3. 执行run.bat

4. 结果在Output目录下面

把结果”core” 目录里面的文件拷贝到loan/core下面即可。

## # 枚举配置

如果需要把某个字段映射成枚举，需要按以下步骤进行

1. 生成枚举类，继承com.netfinworks.ifp.common.interfaces.EnumBase

public enum **ScheduleTestStatus** implements EnumBase {…}

1. 在mybatis-conf.xml里面添加TypeHandler

<typeHandlers>

<typeHandler

handler=" com.netfinworks.ifp.common.template.dao.typehandler.EnumBaseHandler"

jdbcType="VARCHAR"

javaType="com.netfinworks.loan.common.domain.enums.**ScheduleTestStatus**"/>

</typeHandlers>

1. 在generatorConfig.xml的<table>对应的枚举字段，定义columnOverride

<table tableName="t\_schedule\_test" domainObjectName="ScheduleTestDO"

enableDeleteByExample="false" enableUpdateByExample="true">

<**columnOverride** column="STATUS"

javaType="com.netfinworks.loan.common.domain.enums.**ScheduleTestStatus**"/>

</table>

然后，生成代码的时候，会把DO.java和DOCondition.java里面都使用枚举类型。

## # 自定义SQL

为了方便增加数据库表字段的时候，使用重新生成的代码，自定义SQL不在原来的DOMapper.xml文件里面修改，而是手动新增同名文件DOMapperSD.xml

(SD代表“手动”，也可以理解为”Self Define”)

例如：代码生成文件：ScheduleTestDOMapper.xml

<mapper namespace="**com.netfinworks.loan.schedule.core.dal.dao.ScheduleTestDAO**" >

<resultMap id="BaseResultMap"

type="com.netfinworks.loan.schedule.core.dal.dataobject.ScheduleTestDO" >

需要在**同一个目录**下面，建立一个新文件：ScheduleTestDOMapperSD.xml

<mapper namespace="**com.netfinworks.loan.schedule.core.dal.dao.ScheduleTestDAO**" >

<select id="selectBySelfDefine"

resultMap="BaseResultMap" parameterType="java.lang.String" >

注意：namespace需要和DOMapper.xml一样，这样SQL中就可以直接引用DOMapper.xml里面的定义，比如BaseResultMap。

新定义的方法，需要手动添加到DAO.java中。

第二次生成代码的时候，不需要覆盖DAO.java文件。

## # 代码例子

用如下测试表

CREATE TABLE `T\_SCHEDULE\_TEST` (

`ID` int NOT NULL AUTO\_INCREMENT COMMENT 'ID' ,

`LOAN\_PROJ\_NO` varchar(32) NOT NULL COMMENT '贷款项目编号' ,

`TYPE` varchar(32) NOT NULL COMMENT '类型' ,

`TASK\_IDENTITY` varchar(64) NOT NULL COMMENT '任务识别' ,

`CONTEXT` varchar(1024) DEFAULT NULL COMMENT '任务上下文' ,

`STATUS` varchar(32) DEFAULT NULL COMMENT '状态' ,

`TASK\_RESULT` varchar(32) DEFAULT NULL COMMENT '任务处理结果' ,

`CASH` decimal(15,2) DEFAULT NULL COMMENT '现金测试' ,

`EXT` varchar(1024) DEFAULT NULL COMMENT '附加信息' ,

`GMT\_CREATED` datetime DEFAULT NULL COMMENT '创建时间' ,

`GMT\_MODIFIED` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (id),

UNIQUE KEY (`LOAN\_PROJ\_NO`, `TYPE`, `TASK\_IDENTITY`)

);

配置generatorConfig.xml (其中CASH字段生成类型Money)

<table tableName="**t\_schedule\_test**" domainObjectName="**ScheduleTestDO**"

enableDeleteByExample="false"

enableUpdateByExample="false">

<columnOverride column="ID" javaType="java.lang.String"/>

<columnOverride column="CASH" javaType="com.netfinworks.common.util.money.**Money**"/>

</table>

### ## 分页例子

ScheduleTestDOCondition cond = **new** ScheduleTestDOCondition();

cond.setCurrentPage(2); // 页数从1开始

cond.setPageSize(10); // 每页显示数量

List<ScheduleTestDO> list = scheduleTestDAO.selectByCondition(cond);

注释：MySQL分页使用restful-framework-persist-2.0.0.jar中的PAGINATION.xml

LIMIT #{offset},#{pageSize}

其中#{offset} 就是调用QueryConditionBase.getOffset()

### ## 多条件联合查询例子

ScheduleTestDOCondition cond = **new** ScheduleTestDOCondition();

Date currentDate= **new** Date();

Date prevDayDate = DateUtil.*addDays*(currentDate, -1);

cond.createCriteria().andLoanProjNoLikeInsensitive("l123%") // upper(loan\_proj\_no) like 'L123%'

.andCashGreaterThanOrEqualTo(**new** Money(2,0) ) // Cash >= 2.0

.andGmtModifiedGreaterThan(prevDayDate) // 最近24小时修改

;

List<ScheduleTestDO> list = scheduleTestDAO.selectByCondition(cond);