## 一、通用Mapper插件：Mybatis Generator

**MybatisGenerator详解**：

<http://git.oschina.net/free/Mybatis_Utils/tree/master/MybatisGeneator/MybatisGeneator.md>

插件代码在包：tk.mybatis.mapper.generator，一共包含两个类：

MapperCommentGenerator：该类用于生成数据库备注字段的注释，以及实体类字段的注解。

MapperPlugin：插件的实现类，该类默认使用上面这个注释生成器，插件屏蔽了一般的CRUD方法（保留了Example），插件可以生成实体的@Table注解。

两种方式自动生成代码：编码方式+maven插件运行方式

1. 使用Maven执行Mybatis Generator

1）配置pom.xml文件

2）配置MBG配置文件：generatorConfig.xml

3）配置数据库连接等：config.properties

4）在pom.xml所在目录的命令行执行下面命令：

mvn mybatis-generator:generate

5）或者在eclipse，右键pom.xml->run as ->maven build

在goals中输入：mybatis-generator:generate

2、pom.xml配置

参见文件：[pom.xml](mapper-generator/pom.xml)

3、generatorConfig.xml配置

参见文件：[generatorConfig.xml](mapper-generator/generatorConfig.xml)

4、config.properties配置

参见文件：[config.properties](mapper-generator/config.properties)

## 二、Mybatis分页插件

资料：

<http://git.oschina.net/free/Mybatis_PageHelper/blob/master/wikis/HowToUse.markdown>

1、引入分页插件

通过maven引入分页插件，具体的依赖如下：

<dependency>

<groupId>com.github.pagehelper</groupId>

<artifactId>pagehelper</artifactId>

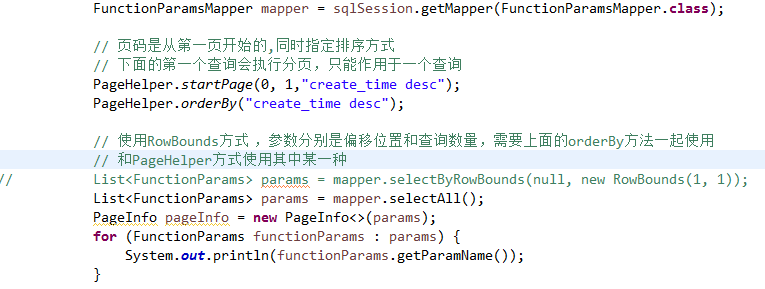
<version>4.1.6</version>

</dependency>

2、在mybatis-config.xml配置分页拦截器插件



3、调用



## 三、Mybatis返回值

Mybatis提供多种元素来匹配不同类型的语句，如：select、insert、update、delete.每种类型的元素需要提供一个id属性，id的值与mapper接口的对应方法名称一致。

**3.1 Insert语句**

## 四、XML配置SQL映射器

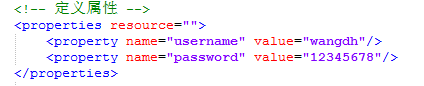
## 五、Mybatis配置文件

学习资料：<http://www.mybatis.org/mybatis-3/zh/configuration.html>

模板参见：[mybatis-config-template.xml](Mybatis/mybatis-config-template.xml)

**5.1 properties属性**

该元素用来配置一下可替换的数据，如数据库配置等。在该元素中定义的配置，可以在整个配置文件中进行使用。配置的格式如下：



可以使用resource或url属性配置外部配置文件。mybatis将按照下面的顺序来加载配置数据(当读取到相应配置后不会再继续往下)：

1. properties元素体内指定的属性
2. 通过resource或url指定的外部配置
3. 通过方法参数传递的属性

**5.2 settings元素**

该配置将会改变mybatis的运行行为，具体的配置项如下：

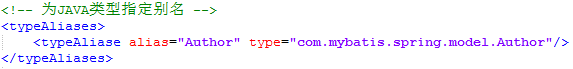
|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **说明** | **有效值** | **默认值** |
| cacheEnabled | 所有映射器中配置的缓存的全局开关 | true/false | true |
| lazyLoadingEnabled | 延迟加载关联对象的全局开关 | true/false | false |
| aggressiveLazyLoading | 当启用时，对任意延迟属性的调用会使带有延迟加载属性的对象完整加载；反之，每种属性将会按需加载 | true/false | true |
| multipleResultSetsEnabled | 是否允许单一语句返回多结果集（需要兼容驱动） | true/false | true |
| useColumnLabel | 使用列标签代替列名 | true/false | true |
| useGeneratedKeys | 允许 JDBC 支持自动生成主 | true/false | false |
| autoMappingBehavior | 指定 MyBatis 应如何自动映射列到字段或属性。 NONE 表示取消自动映射；PARTIAL 只会自动映射没有定义嵌套结果集映射的结果集。 FULL 会自动映射任意复杂的结果集 | NONE/PARTIAL/ FULL | PARTIAL |
| autoMappingUnknownColumnBehavior |  | NONE/WARNING/ FAILING | NONE |
| defaultExecutorType | 配置默认的执行器。SIMPLE 就是普通的执行器；REUSE 执行器会重用预处理语句（prepared statements）； BATCH 执行器将重用语句并执行批量更新。 | SIMPLE/REUSE/ BATCH | SIMPLE |
| defaultStatementTimeout | 设置超时时间，它决定驱动等待数据库响应的秒数 | 任何正整数 | null |
| defaultFetchSize | 返回结果的数量，会被查询语句覆盖 | 任何正整数 | null |
| safeRowBoundsEnabled | 允许在嵌套语句中使用分页（RowBounds） | true/false | false |
| safeResultHandlerEnabled | 允许在嵌套语句中使用分页（ResultHandler） | true/false | false |
| mapUnderscoreToCamelCase | 否开启自动驼峰命名规则（camel case）映射，即从数据库列名 A\_COLUMN到 Java 属性名 aColumn 的类似映射。 | true/false | false |
| localCacheScope | MyBatis 利用本地缓存机制（Local Cache）防止循环引用（circular references）和加速重复嵌套查询。 默认值为 SESSION，这种情况下会缓存一个会话中执行的所有查询。 若设置值为 STATEMENT，本地会话仅用在语句执行上，对相同 SqlSession 的不同调用将不会共享数据。 | SESSION  STATEMENT | SESSION |
| logPrefix | 指定 MyBatis 增加到日志名称的前缀。 | 任何字符串 | null |

配置实例：



5.3 typeAliases元素

该元素用于为java类型设置一个较短的名字，以消除完全限定名的冗余，配置示例：



Author可以用在任何使用com.mybatis.spring.model.Author的地方

也可以指定一个包名，mybatis会自动扫描需要的Java bean，这种情况下，当该bean没有注解时，会使用首字母小写的非限定名作为别名。



5.4 typeHandlers元素

## 六、Mybatis集成通用mapper(非spring)

6.1 获取通用mapper和mybatis及对应的数据库驱动jar包依赖

<!-- SqlServer 驱动 -->

<dependency>

<groupId>com.microsoft.sqlserver</groupId>

<artifactId>sqljdbc4</artifactId>

<version>${sqlserver.version}</version>

</dependency>

<!-- mybatis -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.3.0</version>

</dependency>

<!-- 通用mapper -->

<dependency>

<groupId>org.mybatis.generator</groupId>

<artifactId>mybatis-generator-core</artifactId>

<version>1.3.2</version>

<scope>compile</scope>

<optional>true</optional>

</dependency>

6.2 在代码中使用下面语句继承mapper

// 需要使用这段代码来集成通用mapper

MapperHelper mapperHelper = new MapperHelper();

mapperHelper.registerMapper(FunctionParamsMapper.class);;

mapperHelper.processConfiguration(sqlSession.getConfiguration());

6.3 和使用mybatis一样正常使用了

## 7、Mybatis打印SQL语句

新建log4j.properties文件，添加下面内容：

log4j.rootLogger=debug,stdout

#

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%d %p [%c] -%m%n

#

#

log4j.logger.com.ibatis=debug

#print package soc.dao sql

log4j.logger.soc.dao=debug

log4j.logger.com.ibatis.common.jdbc.SimpleDataSource=debug

log4j.logger.com.ibatis.common.jdbc.ScriptRunner=debug

log4j.logger.com.ibatis.sqlmap.engine.impl.SqlMapClientDelegate=debug

log4j.logger.java.sql.Connection=debug

log4j.logger.java.sql.Statement=debug

log4j.logger.java.sql.PreparedStatement=debug,stdout