1. 简介
2. Mybatis概念
3. ORM工具的基本思想
4. 开发环境搭建
5. 所需jar包
6. 设置mybatis配置文件，configuration.xml
7. 新建与数据库表对应的实体类
8. 新建实体类的映射文件 .xml
9. 加载配置文件获取SqlSessionFactory来实例化SqlSession执行SQL语句
10. 接口编程 -- 用接口代替sqlSession查询方法

二、单表增删改查的实例demo

1. 查询语句 -- select
2. 增加/插入语句 - insert
3. 更新数据语句 - update
4. 删除数据语句 - delete

三、关联数据的查询

四、Mybatis动态SQL语句基础

1. if语句 -- 用来拼接SQL语句
2. choose语句 -- 用来拼接SQL语句
3. trim语句 -- 修改SQL语句
4. where语句 -- 简化SQL语句
5. set语句 -- 简化SQL语句
6. foreach语句 -- 批量插入，一般用于in语句中

五、向SQL语句中传入多个参数

1. 利用HashMap传入多个参数
2. 利用Mybatis自带传递多个参数的方式

六、Mybatis代码自动生成工具

七、实现Mybatis分页效果 - 网页显示

九、Mybatis和Spring集成

十、Mybatis和Spring MVC 集成

# 简介

## Mybatis概念

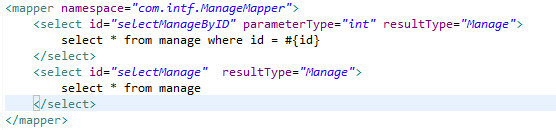
MyBatis 是支持普通 SQL 查询，存储过程和高级映射的优秀持久层框架。MyBatis 消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML 或注解用于配置和原始映射，将接口和 Java 的 POJOs（Plan Old Java Objects，普通的 Java 对象）映射成数据库中的记录。

## ORM工具的基本思想

1. 从配置文件(通常是 XML 配置文件中)得到 sessionfactory。
2. 由 sessionfactory 产生 session。
3. 在 session 中完成对数据的增删改查和事务提交等。
4. 在用完之后关闭 session 。
5. 在 Java 对象和 数据库之间有做 mapping 的配置文件，也通常是 xml 文件。

二、增删改查的实例demo

1. 查询语句 -- <select>标签



selectMange返回的是List数组，但是resultType表示的是List中的对象类型

1. resultType 和 resultMap的区别
2. resultType：当使用resultType做SQL语句返回结果类型处理时，对于SQL语句查询出的字段（使用别名用来对应pojo中的变量名）在相应的pojo中必须有和它相同的字段对应，而resultType中的内容就是pojo在本项目中的位置。

--- resultType的类中属性要和SQL语句中的返回值一一对应

1. resultMap：当使用resultMap做SQL语句返回结果类型处理时，通常需要在mapper.xml中定义resultMap进行pojo和相应表字段的对应。

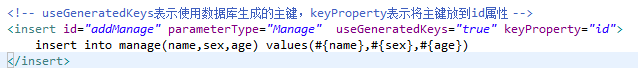
--- resultMap可以将返回值一一对应到类的属性名



* resultMap中 将类的属性名和SQL返回值的属性名进行一一手动匹配，并由id名表示。

resultType中 类的属性名 和 SQL返回值的属性名 一一对应，自动匹配。

1. 增加/插入语句 - insert



1. 在接口类中增加方法：public void addManage（Manage manage）
2. 在.xml中增加SQL插入语句
3. 测试类中调用，必须sqlSession.commit(); - 除了查询语句都要加入提交语句。
4. 更新数据语句 - update

C:\Users\HP\AppData\Local\Temp\1511851573(1).png

1. sqlSession.commit()要加入提交语句。
2. 删除数据语句 - delete

C:\Users\HP\AppData\Local\Temp\1511851939(1).png

1. sqlSession.commit()要加入提交语句。

三、关联数据的查询



1. resultMap中<association>可以用来引用别的实体类，匹配引用的实体类属性

如：public class Article{

private Manage manage;

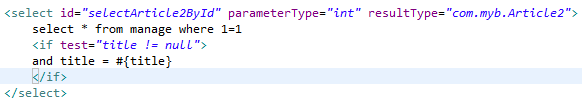
}

property属性指的是Article中manage属性，javaType指的是该属性的实体类。

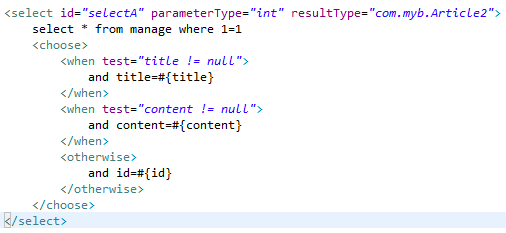
1. SQL语句返回的属性值/属性名 与 配置中的属性名 一一对应！！！！！！！
2. SQL语句返回的属性名不得重复，否则自动匹配会失败。
3. Mybatis中.xml文件的SQL语句返回值 和 指定的实体类 是映射的是自动匹配的，匹配的原则就是根据属性名。

四、Mybatis动态SQL语句基础

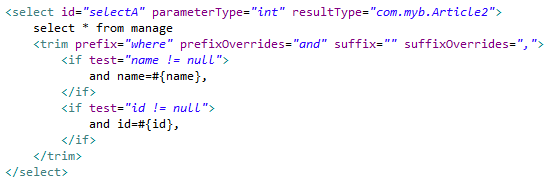
1. if语句 -- 用来拼接SQL语句



1. choose语句 -- 用来拼接SQL语句



1. 当满足when时，就跳出choose，如果都不满足，就执行otherwise。和switch类似。
2. trim语句 -- 修改SQL语句



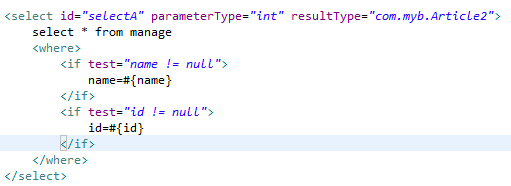
prefix：添加前缀，不关心prefixOverrides判断结果。

suffix：添加后缀，不关心suffixOverrides判断结果。

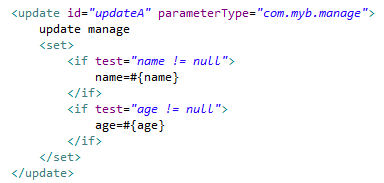
prefixOverrides：判断前缀内容，为真就用prefix覆盖。

suffixOverrides：判断后缀内容，为真就用suffix覆盖。

1. where语句 -- 简化SQL语句



1. where语句会自动为条件添加and或者or，会自动忽略掉第一个and或者or
2. set语句 -- 简化SQL语句



1. 和where语句功能类似，主要用于更新语句。
2. foreach语句 -- 批量插入，一般用于in语句中
3. 六个参数

collection：要循环的集合，List，Array

index：循环索引，比如当前循环到第一个，index值为0，第二个，index值为1，是自增值。

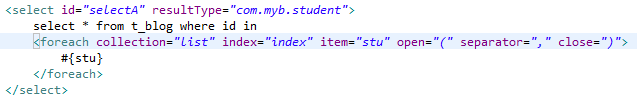
item：集合中的一个元素，循环时候用的别名。

open：以什么开始

close：以什么结束

separator：循环内容之间以什么分隔

1. 传入参数为list时 -- collection值为list



1. 传入参数为数组时， -- collection值为array
2. 传入参数为Map时， - collection值为Map的值

当Map中只有一个key，且value值为list或者array时，collection值为这个key值。

五、向SQL语句中传入多个参数

1. 利用HashMap传入多个参数

C:\Users\HP\AppData\Local\Temp\1511863056(1).png

1. 接口加入方法。

public Manage selectManageByMap(Map<String,Object> param);

1. 测试时新建一个HashMap，传入参数。
2. 利用Mybatis自带传递多个参数的方式 -- 不重要，省略

六、Mybatis代码自动生成工具 - 在Maven中使用

1. 配置pom.xml文件，导入相关jar包以及plugin。

详细配置见文件 ：mybatis-generator-pom.xml

1. 配置generatorConfig文件，放在src/main/java目录下面

详细配置见文件 ： generatorConfig.xml

1. 在Maven--build中输入命令：mybatis-generator:generate
2. 使用自动生成Example

