# Day48 Mybatis

## Mybatis简介

### 1.历史由来

Mybatis的前身叫ibatis,是Apache的一个开源项目,2010年从Apache转移到了GoogleCode开源组织中了,从此改名为Mybatis.

### 2.Mybatis概念

Mybatis是一种支持普通的SQL查询,存储过程和高级映射等功能的持久层框架,和Hibernate功能类似.

### 3.Mybatis的优点

Mybatis几乎消除了所有的JDBC代码和参数的手工设置,以及结果集的手工检索.

### 4.Mybatis自身的特色

具有比较强大的动态语句功能.Mybatis框架更侧重于JavaBean与表之间的映射关系.

### 5.Mybatis框架的核心API

#### ①.SqlSessionFactoryBuilder:

SqlSessionFactoryBuilder是Mybatis框架执行的入口,我们利用SqlSessionFactoryBuilder类build()方法来创建出SqlSessionFactory对象.我们通常情况下,会在xml配置文件中配置SqlSessionFactoryBuilder的核心对象Configuration,对我们的数据源进行各种配置.

#### ②.SqlSessionFactory:

由SqlSessionFactoryBuilder创建出来的,该类的主要功能是创建SqlSession对象.这个类对象没有必要每次使用都创建一遍,也就是说该类对象只需要创建一个就可以了.

#### ③.SqlSession:

SqlSession对象主要是进行数据库的各种操作,比如进行数据库的访问和结果集的映射.

注意:

SqlSession不是线程安全的,因此不能被类中多个方法所共享,也就是说该类的对象作用域会在方法内部,在方法内部创建该对象,并且在使用完之后及时的close().

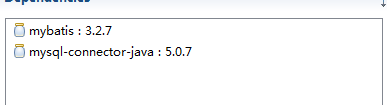
### 6.Mybatis和Hibernate对比

Hibernate属于比较重量级的框架,使用起来比较麻烦,入门的门槛比较高.但是Hibernate功能很多,方法很全.

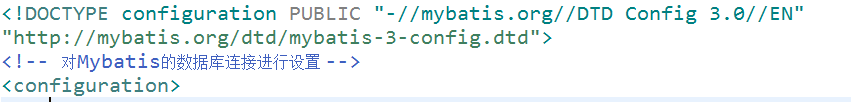
Mybatis属于比较轻量级的框架,使用起来比较简单,入门门槛较低.但是Mybatis功能相对来说单薄点,功能较少.

## Mybatis的具体开发

### 1.添加依赖



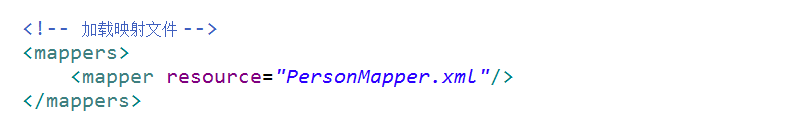
### 2.创建Mybatis.xml配置文件





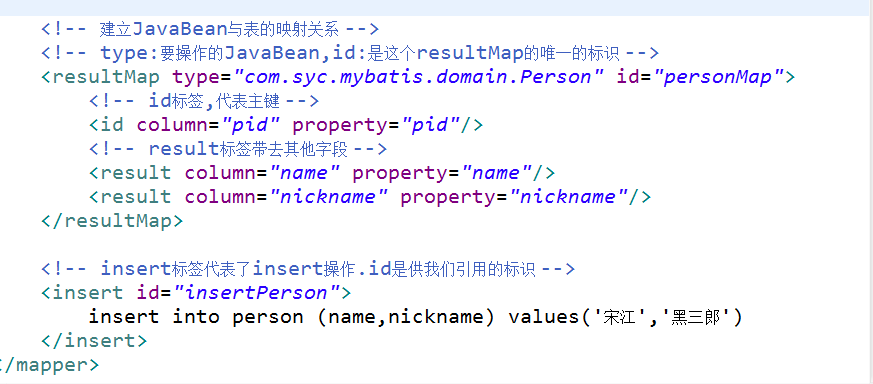


加载映射文件:

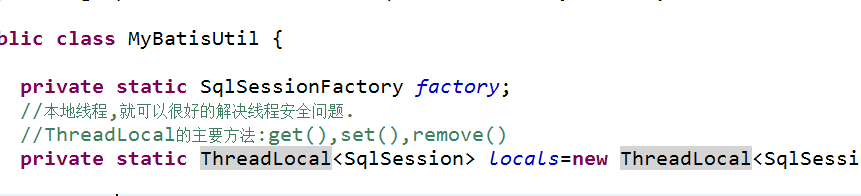


### 3.创建并配置PersonMapper.xml映射文件





### 4.创建MybatisUtil工具类,主要是用来创建SqlSessionFactory对象,并提供获取SqlSession和关闭SqlSession的方法.









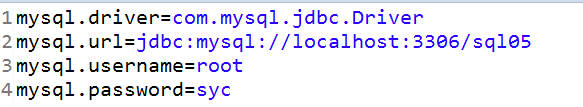
### 5.封装PersonDao类



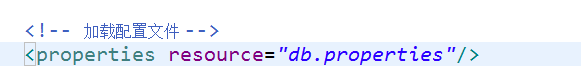
## 三.简化Mybatis配置和操作

### 1.通过db.properties配置文件简化数据源的配置

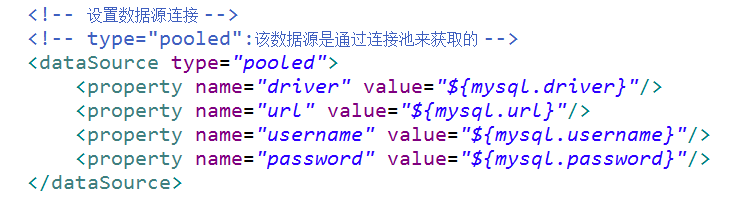
#### db.properties配置文件:



#### 引入配置文件:



#### 简化配置:

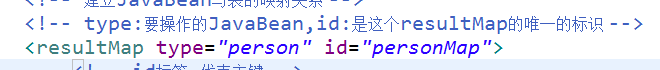


### 2.通过类型别名来简化类型的书写

#### 定义类型别名:



#### 进行person别名的引用:

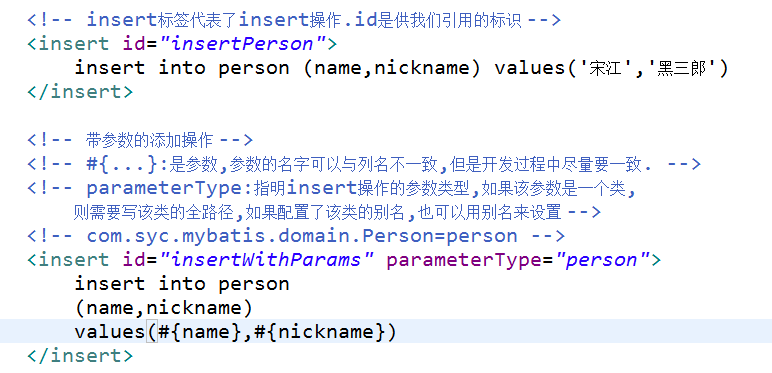


### 3.Java代码中的简化

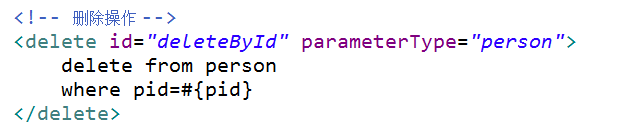


## 四.实现Mybatis的CRUD操作

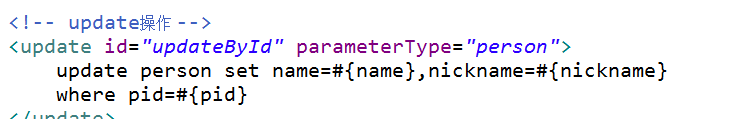
### 1.insert操作:



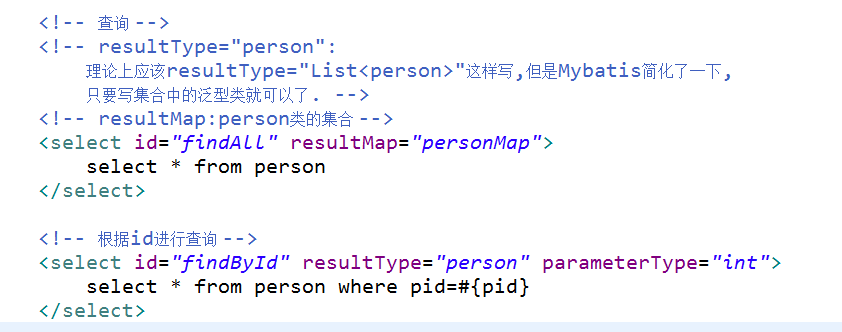
### 2.delete操作



### 3.update操作

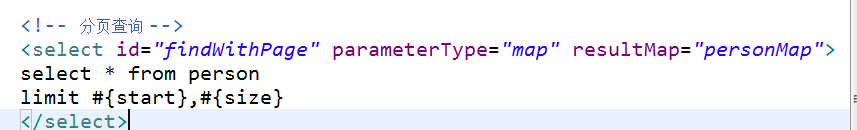


### 4.select操作

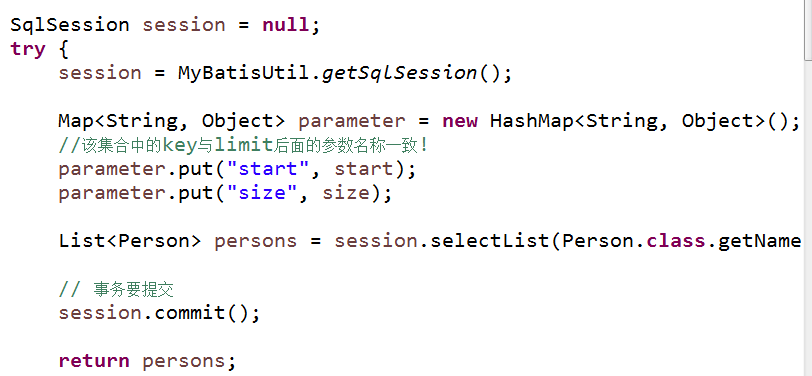


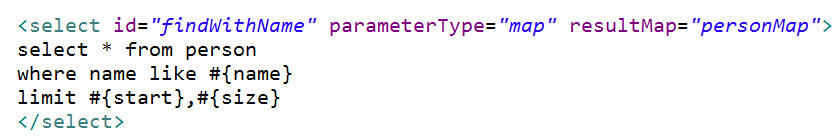
## 五.Mybatis实现分页查询

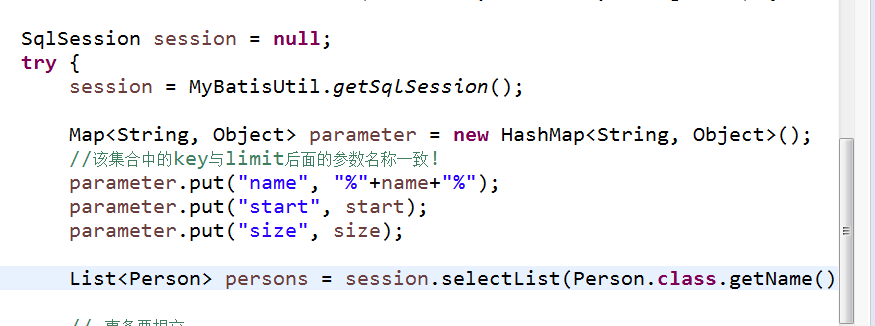
#### xml中的配置:



#### dao方法:

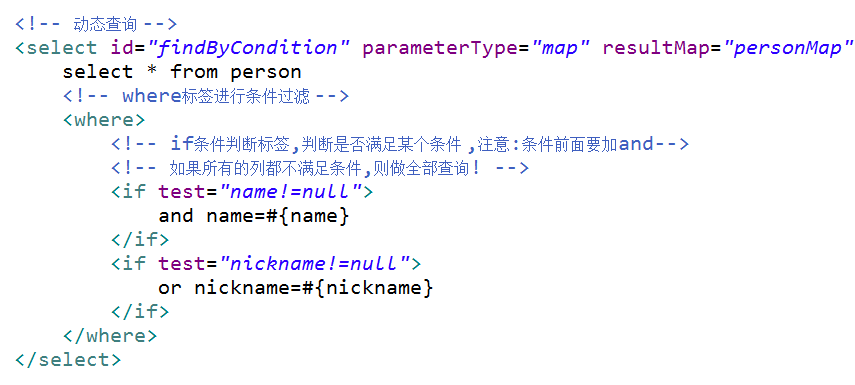


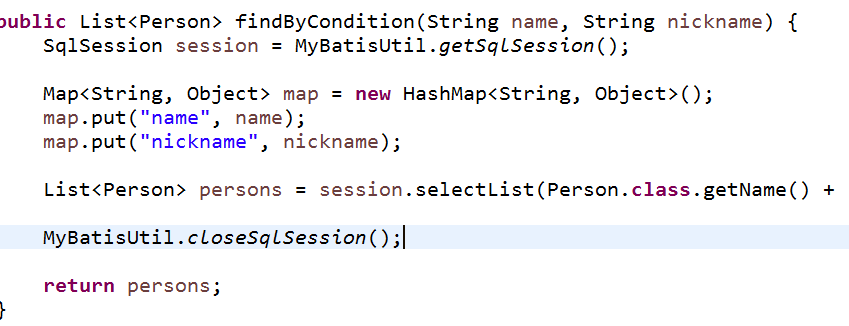




## 六.Mybatis的动态语句

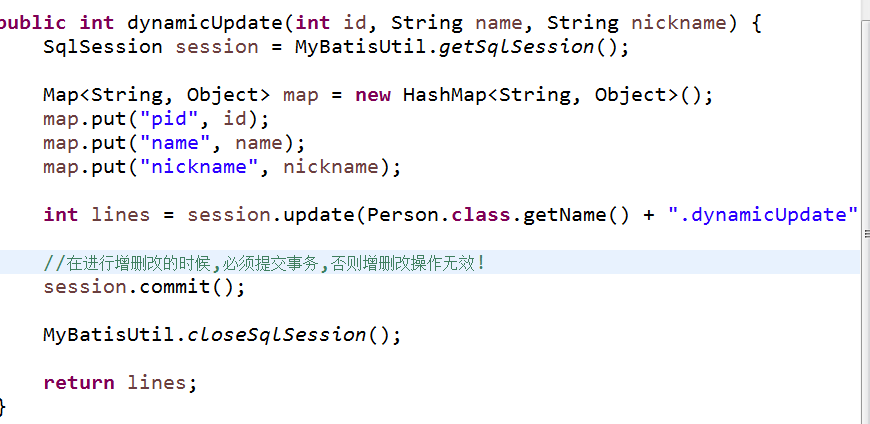
### 1.动态查询



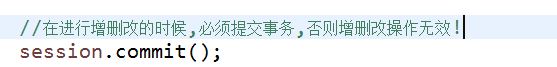


### 2.动态更新



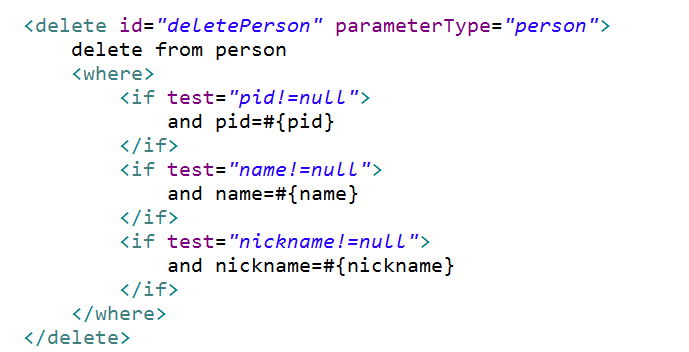


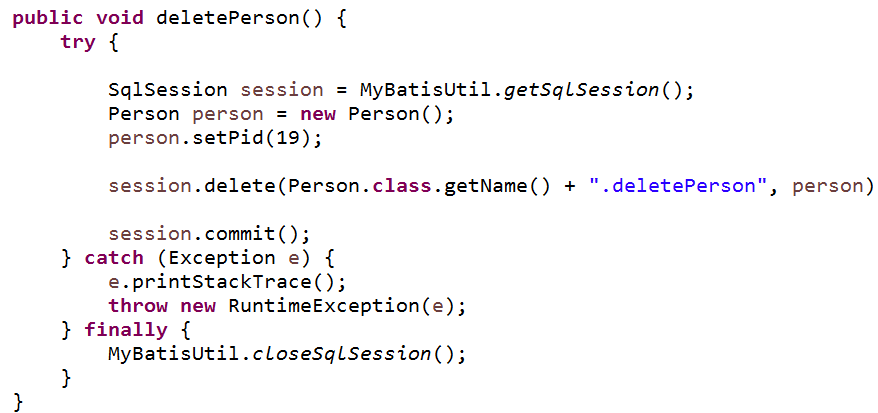
注意:



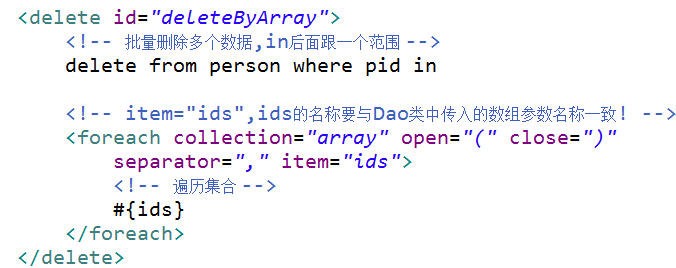
### 3.动态删除

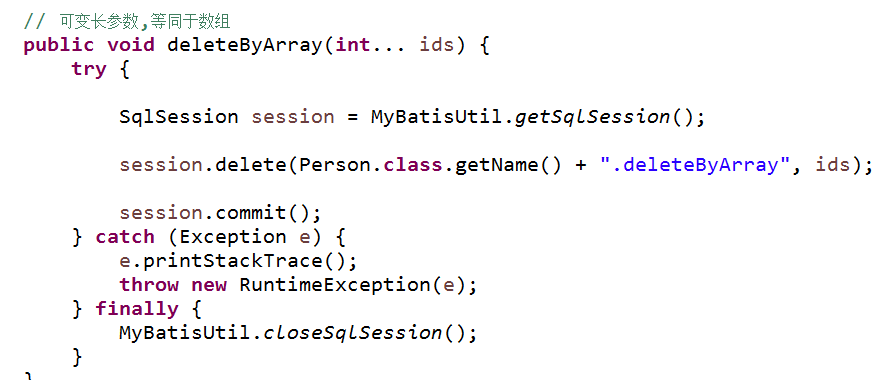
普通的删除:



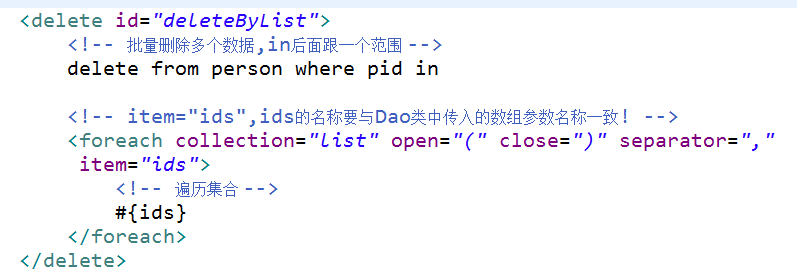


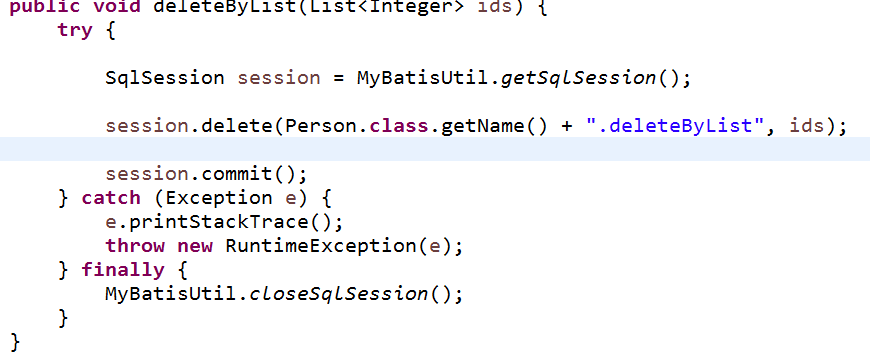
批量删除-根据数组删除:





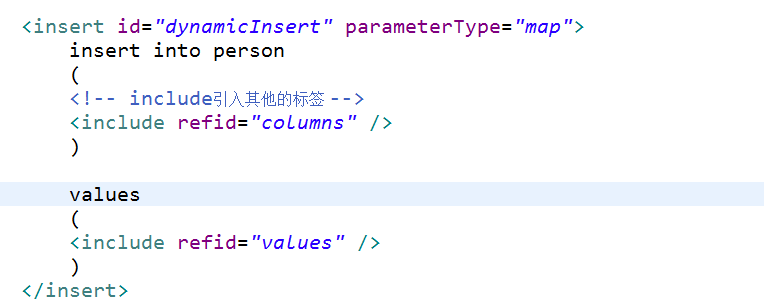
批量删除-根据List集合删除:





### 4.动态添加





dao类代码:

