

# 面试题总结

---

## 一.mybatis的面试题总结

- 1.三层架构及其作用
- 2.mybatis多参数处理应用场景及其处理方式
- 3.谈谈你对mybatis面向接口编程的理解
- 4.#{ }和\${ }的区别
5. 常见的动态sql标签及其作用
- 6.mybatis执行流程

## 一.mybatis的面试题总结

### 1.三层架构及其作用

三层架构：表示层，业务逻辑层，会话层

表示层：说的是我们的界面层UI

业务层：我们专注的是业务逻辑的一个处理，不用管哪写数据怎么查，怎么跳页面的

会话层：也叫数据访问层，就是用户专门负责和数据库做交互的为我们的业务逻辑，提供数据的

三层的作用：为了更好的解耦，这样各层各司其职，各层之间使用接口去访问  
当我们的某一个模块有错误的时候，尽可能少影响到其他层的代码，做到高内聚，低耦合。

### 2.mybatis多参数处理应用场景及其处理方式

mybatis多参数处理方式：大致分为五种方式

- 2.1. 使用 mybatis这个半orm框架底层的默认方式，arg0,arg1的方式
- 2.2. 使用 mybatis 的方式 params1.params2这样的方式进行数据映射
- 2.3. 使用注解映射 @Param("userid")uid 这样的方式一般是在参数比较少的使用,及其推荐
- 2.4. 使用对象作为数据载体进行处理，一般数据比较多，可以使用属性名进行sql传参
- 2.5. 使用map传参，这个一般是一些杂乱的数据，当然可以封装实体类，不过map也可以

### 3.谈谈你对mybatis面向接口编程的理解

我们先想一想，如果没有mybatis面向接口编程，举个例子：比如我们mapper层有一个userMapper,service层有一个userService,-----我们在调用的时候需要这么办 在userService类中 UserMapper userMapper=new UserMapperImpl,如果是这样的话，mapper层代码一旦改动，我们编译的时候需要对UserMapperImpl进行重新编译，而且我们的userService也要编译，如果使用了mybatis的面向接口编程，将我们的实体类直接用xml文件替代了，都知道xml文件是不需要编译的，就达到了一种解耦的效果，今后只改xml，不用管其他的

#### 4. #{}和\${}的区别

#{} ----->相当于占位符，先预编译

\${}-----> 字符串拼接符，先拼接，在编译，这样就有sql注入的风险

所以建议使用#{ }的方式

#### 5. 常见的动态sql标签及其作用

<sql></sql> 动态sql片段,可以在一处定义,在sql中处处使用

<if></if> 表示对字段进行判断，经常用于空判和“”里面写and,or等

<set></set> 表示更新的时候set关键字，和删除字段后面的逗号

<where></where> 表示查询的时候where关键字，和删除字段前面的and/or

<trim></trim> 很灵活有四个属性，前后添加什么，前后删除什么，可以替换set和where

<foreach></foreach> 表示循环，可以遍历 map,list,array,对象（注意类型写对象属性）

#### 6.mybatis执行流程

mybatis呢，最关键的两个组件就是sqlSessionFactory和SqlSession,sqlSessionFactory用于构建全局配置文件，mybatis-config.xml,获取数据源，解析主配置文件中的mapper标签获取sql语句,然后通过sqlSession的-----openSession()获取执行器（简单，复用批量）去执行sql语句。