

# 苍穹外卖 面试问答

---

Day 02

Day 03

## Day 02

Q: DTO和员工类属性名相同还要一个个去set吗?

A: 不需要, 对象属性拷贝。注意 `employee员工类` 多出来的属性, 需要自己手动set。

K: `LocalDateTime.now()`

K: 记录自己的MySQL数据库账号root密码root端口3306

Q: IDEA配置从mapper.java到mapper.xml自动SQL

A: 黑马javaweb  [Day08-15. Mybatis-入门-配置SQL提示\\_哔哩哔哩\\_bilibili](#)

1. 设置IDEA内嵌提示SQL语句提示
2. `File | Settings | Languages & Frameworks | SQL Dialects` 两个方言设置成 Mysql
3. `IDEA右侧bar-Database` 输入账号密码连接sky\_take\_out数据库, test connection 连接成功即可

P: MyBatis注入可以一定程度上解决该问题

Q: 数据表id\_number可以被类idNumber (驼峰命名) 识别吗?

A: 可以, 因为在application.yml有相关配置: `map-underscore-to-camel-case: true`

▼ application.yml

Plain Text

```
mybatis:
  #mapper配置文件
  mapper-locations: classpath:mapper/*.xml
  type-aliases-package: com.sky.entity
  configuration:
    #开启驼峰命名
    map-underscore-to-camel-case: true
```

Q: 新增员工调试报错 500

A: 极有可能是mapper里的sql语句写错了, 注意驼峰

```
{#{idNumber}, #{createTime}, #{updateTime}, #{createUser}, #{updateUser}}
```

Q: 如何获取当前用户id

A: 在拦截器代码 `JwtTokenAdminInterceptor.java` 中已经通过 `jwt反向解析` 获得了 `empId`

▼ JwtTokenAdminInterceptor.java

Java

```
1 log.info("jwt校验:{}", token);
2 Claims claims = JwtUtil.parseJWT(jwtProperties.getAdminSecretKey(), token);
3 Long empId = Long.valueOf(claims.get(JwtClaimsConstant.EMP_ID).toString());
4 log.info("当前员工id: ", empId);
```

我们使用 `ThreadLocal` 设置线程局部变量

▼ BaseContext

Java

```
1 package com.sky.context;
2 public class BaseContext {
3
4     public static ThreadLocal<Long> threadLocal = new ThreadLocal<>();
5
6     public static void setCurrentId(Long id) {
7         threadLocal.set(id);
8     }
9
10    public static Long getCurrentId() {
11        return threadLocal.get();
12    }
13
14    public static void removeCurrentId() {
15        threadLocal.remove();
16    }
17
18 }
19
```

▼ JwtTokenAdminInterceptor.java

Java

```
1 BaseContext.setCurrentId(empId);
```

▼ EmployeeServiceImpl.java

Java

```
1 employee.setCreateUser(BaseContext.getCurrentId());
```

### Q: PageHelper底层实现

A: PageHelper 是 MyBatis 中一个常用的分页插件。其底层实现主要依赖于 ThreadLocal 和 SQL 的拦截机制。具体步骤如下：

#### 1. ThreadLocal 存储分页信息，设置分页参数

在执行分页查询之前，PageHelper 会通过 ThreadLocal 来存储分页参数，如页码和每页记录数。ThreadLocal 保证了在同一个线程中的任何地方都可以方便地获取到这些分页参数。

```
PageHelper.startPage(pageNum, pageSize);
```

#### 2. SQL 拦截，拼接分页 SQL

PageHelper 使用 MyBatis 提供的拦截器机制来拦截执行的 SQL 语句。通过实现 Interceptor 接口，PageHelper 可以在 SQL 执行前后进行处理。在拦截到 SQL 语句之后，Page

`Helper` 会从 `ThreadLocal` 中获取之前设置的 `分页参数` 动态拼接适合该数据库的分页 SQL 语句。例如，对于 `MySQL` 会在后面添加 `LIMIT` 子句：

### 3. 执行分页查询后清除 `ThreadLocal`

好的，以下是对 `createTime` 是数组对象而不是日期格式，以及什么是消息转换器的完整回答：

Q：返回的`createTime`是个数组对象，而不是日期格式怎么办

A：可以扩展 `SpringMVC` 的消息转换器解决

Q：什么是消息转换器

A：**消息转换器（Message Converter）**是 `SpringMVC` 框架中用于在 HTTP 请求和响应之间**转换数据**的组件。它们负责将 `请求数据` 转换为 `Java 对象`，并将 `Java 对象` 转换为 `响应数据`。消息转换器在处理 `JSON`、`XML`、`表单数据`等格式时尤为重要。

Q：SpringMVC 中常用的消息转换器：

1. **MappingJackson2HttpMessageConverter**：用于将 `JSON` 数据转换为 `Java 对象`，或将 `Java 对象` 转换为 `JSON` 数据。它依赖于 `Jackson` 库。
2. **MappingJackson2XmlHttpMessageConverter**：用于将 `XML` 数据转换为 `Java 对象`，或将 `Java 对象` 转换为 `XML` 数据。它依赖于 `Jackson XML` 扩展。

## Day 03

Q：公共字段自动填充体现面向切面编程

A：sky-take-out/sky-server/src/main/java/com/sky/aspect/AutoFillAspect.java

```
▼ AutoFillAspect.java Java
1  @Pointcut("execution(* com.sky.mapper.*(..)) && @annotation(com.sky.annot
   ation.AutoFill)")
2  public void autoFillPointCut(){}

```

L：切面编程 通知（前置、环绕）

L: 黑马 阿里云对象存储服务 OSS

L: @Slf4j日志

AccessKey ID: LTAl5tJ8E1DSyyUZhgBvWdid

AccessKey Secret: OFEchPHD5qAkT0qPDhYTmPCSFnQjdo

<insert id="insert" useGeneratedKeys="true" keyProperty="id">

insert后产生的主键id值赋给id

什么是DTO? 什么是VO?

notes: git push时会检查dev.yml中的id和secret设置以确保安全