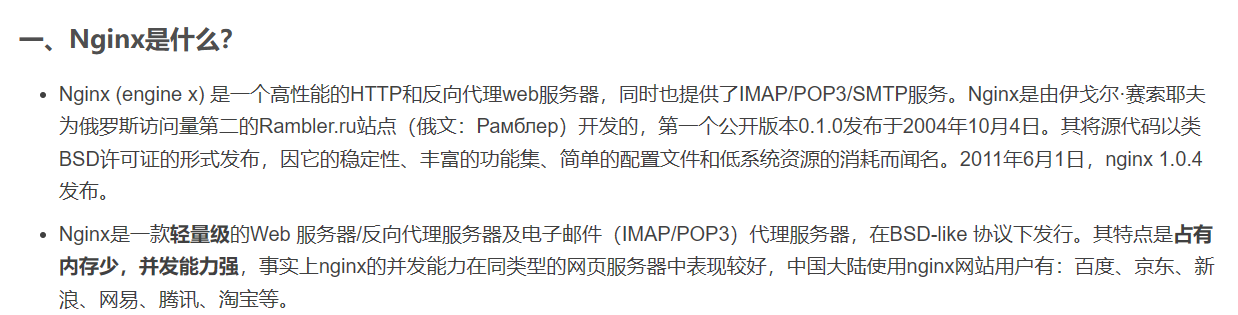
软件环境：开发环境，测试环境，生产环境

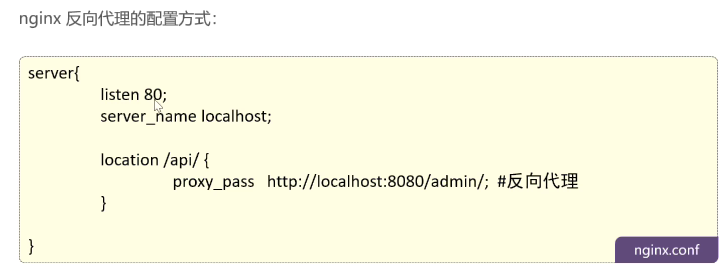
管理端：员工管理，分类管理，菜品管理，套餐管理，订单管理，工作台，数据统计，菜单提醒

用户端：微信登录（小程序），商品浏览，购物车，用户下单，微信支付，历史订单，地址管理，用户催单

# Nginx

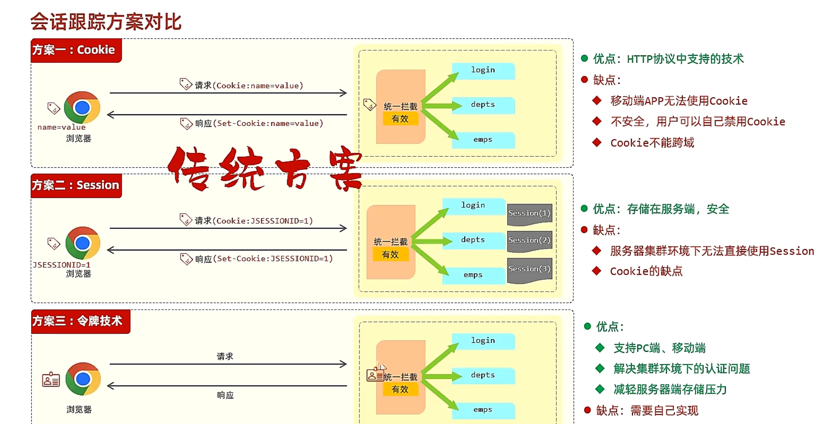


Nginx反向代理：提高访问速度，进行负载均衡（通过nginx来分发大量请求），保证后端服务安全（隐藏真正的后端域名）





# JWT





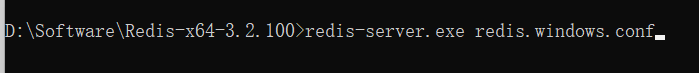


# Redis

基于**内存**的key-value结构数据库，mysql是存在磁盘中的，存在内存中的好处是读写性能高。

适合存储热点数据（适用于热点数据，某时间大量用户访问）

启动服务，默认端口：6379



停止服务：Ctrl+c

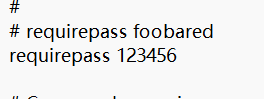
连接redis服务



也可通过这个命令连接服务



设置密码：

在中可以设置密码 

输入密码：



Redis中5种常用数据类型

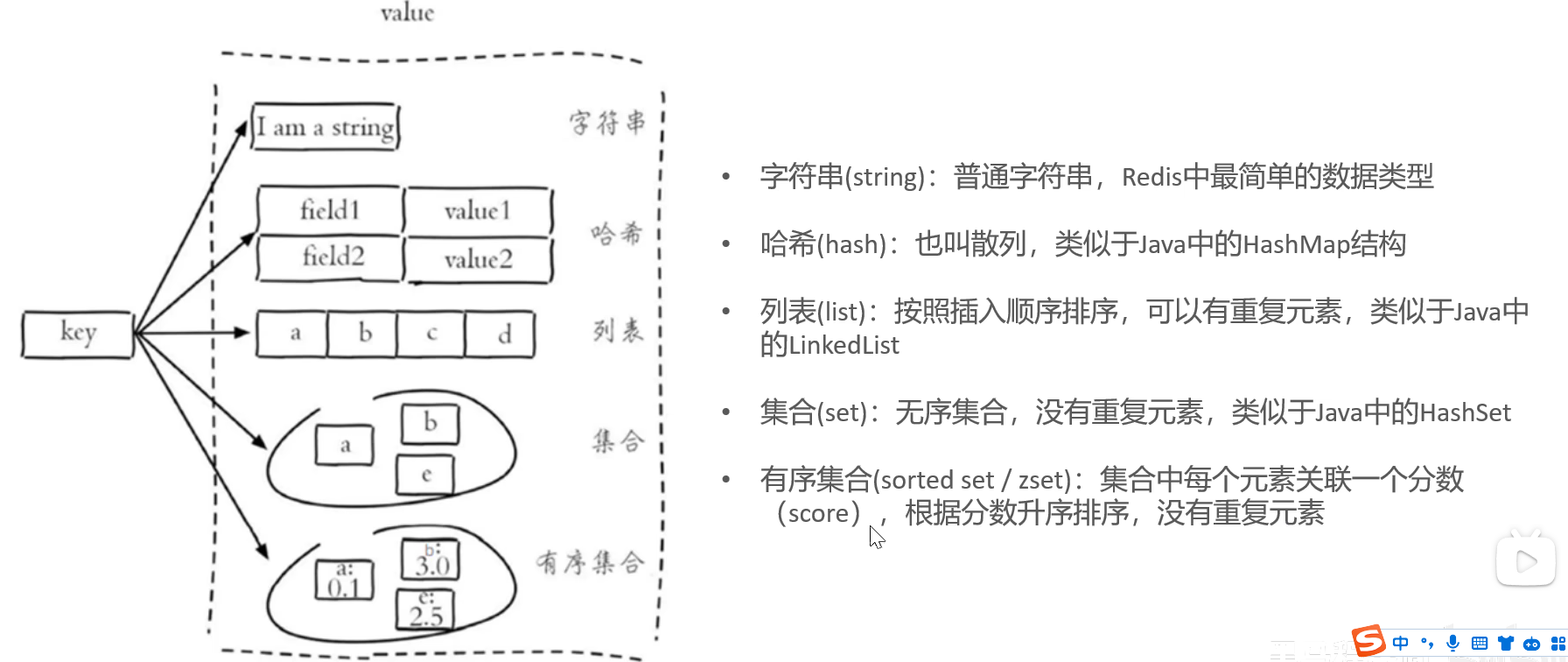
Redis是存储的是键值对结构的数据

Key：字符串类型

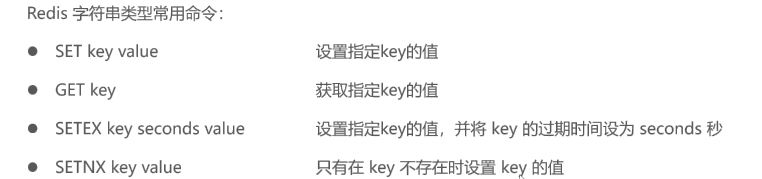
Value有五种常用的数据类型：

**字符串 String**；**哈希 hash**； **列表 list**； **集合 set**；

**有序集合 sorted set/zset**

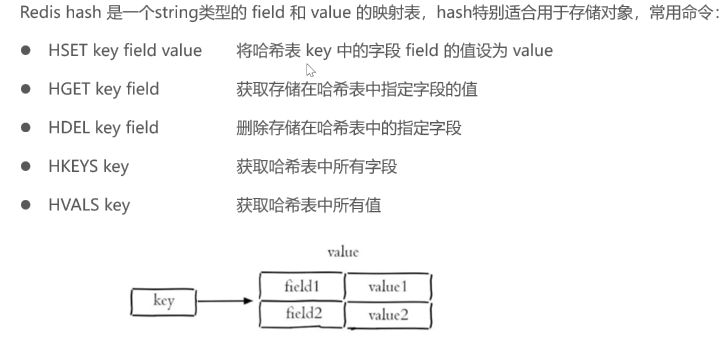


字符串操作命令

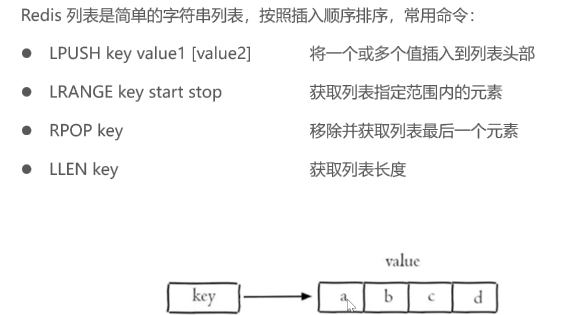


第三条命令适合存储验证码（5分钟后过期）

哈希操作命令



列表操作命令



左侧为头部，右侧为尾部

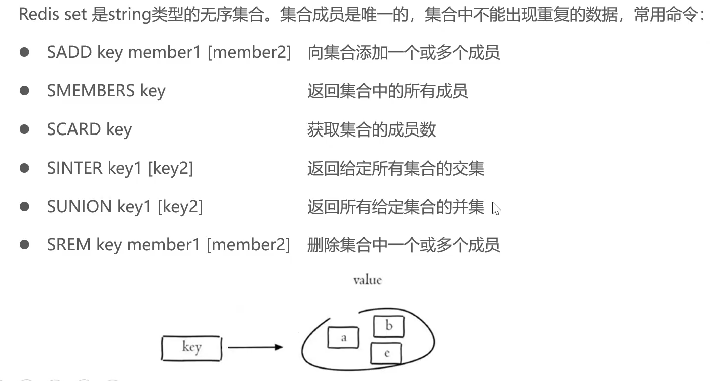
插入命令是1 2 3 ；；；；3 2 1（3是头部，1是尾部）

查询中0表示头部，-1表示尾部

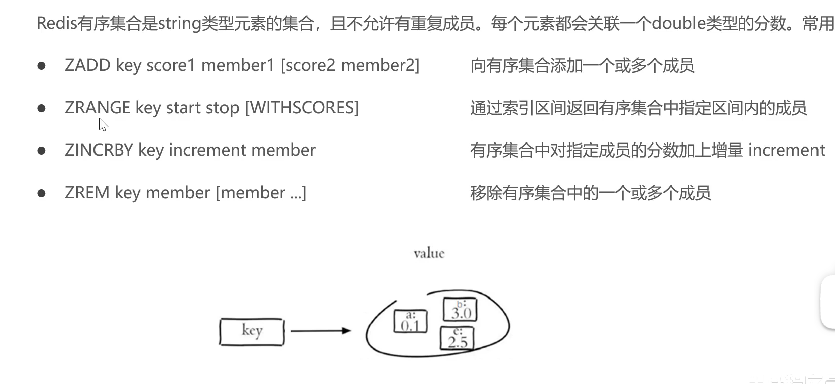
Lpop从左侧开始删除 rpop从右侧开始删除

Lpush从左侧开始添加 rpush从右开始添加

集合操作命令

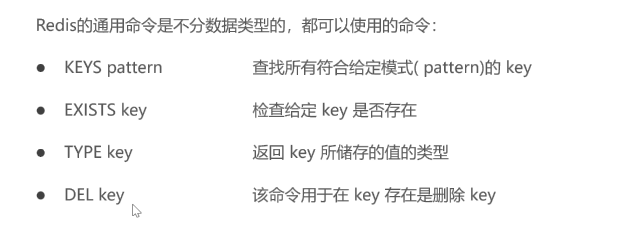


有序集合操作命令



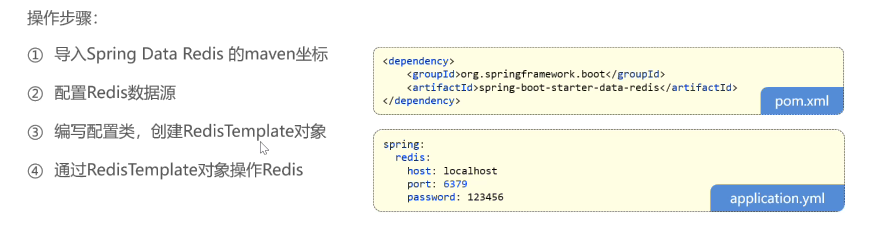
实现有序是大根堆

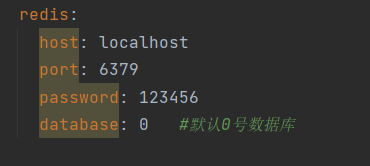
通用命令

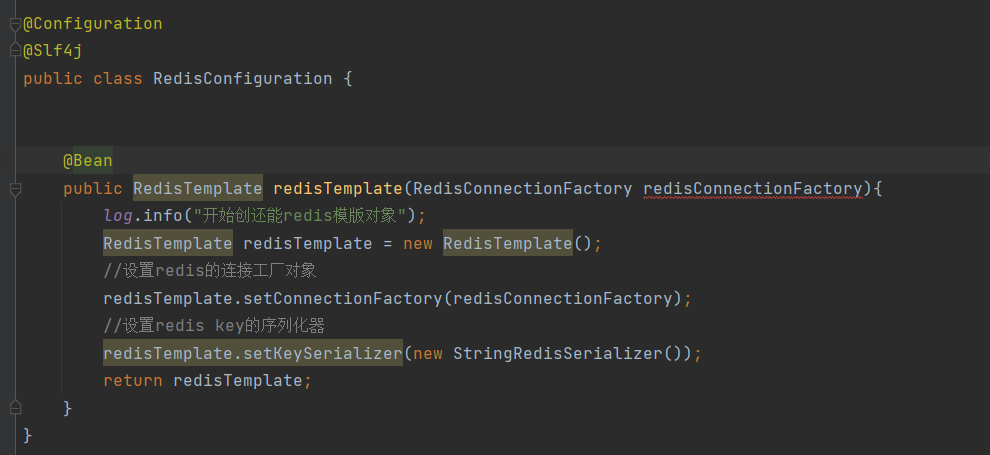


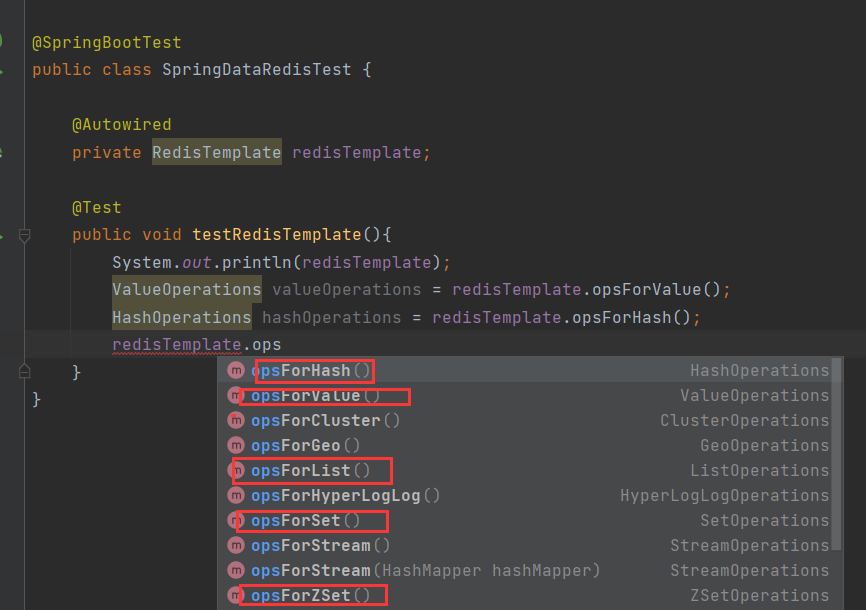
在java中操作redis

Redis的java客户端：jedis，lettuce，**spring data redis**







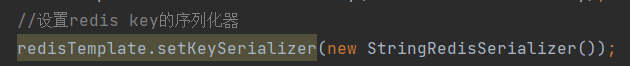


调用这些接口可以实现对不同数据类型进行操作

字符串类型



这个拿来设置超时



设置序列化器可以防止key值乱码



哈希类型



几种常用数据类型大同小异，记住方法就好

# VO和DT

# httpClient

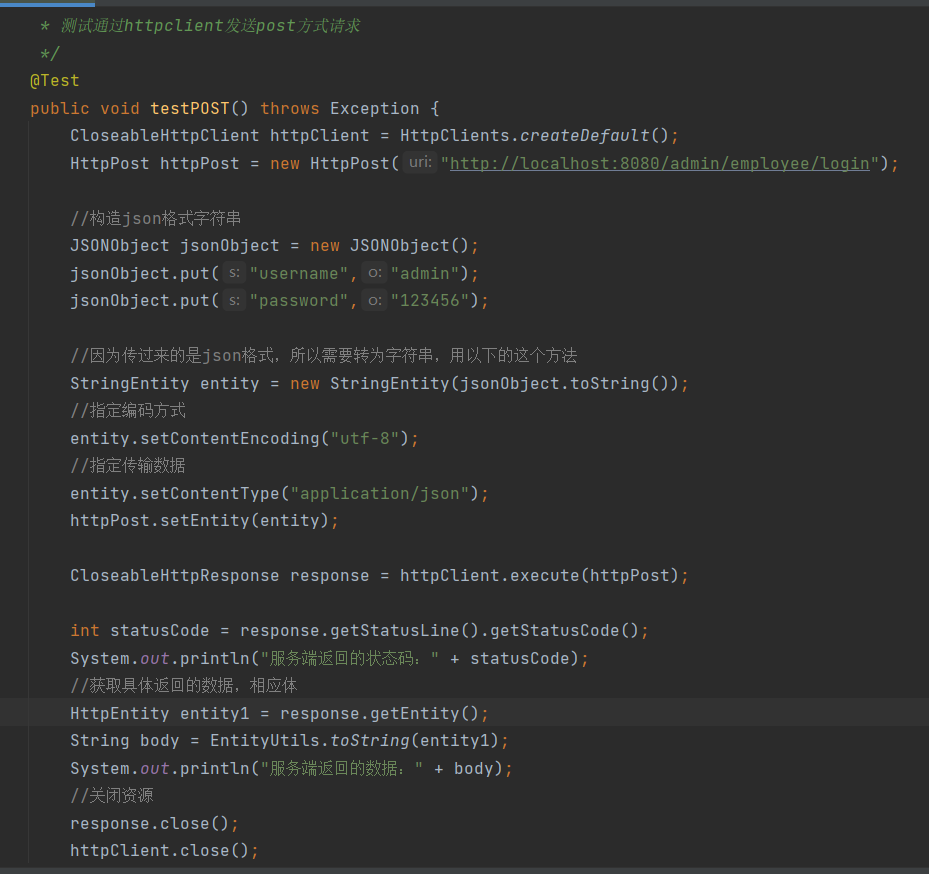


Aliyunoss已经包含了httpclient

不用重新导入maven了，以下是测试get请求



测试post请求



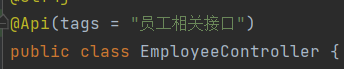
# 微信小程序

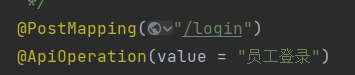


wxf647266814baa6c8 f16dc5a00614a521e2fb60fd613efde2

# Swagger







请求体以json的格式提交，需要加上注解@RequestBody

## 新增员工

账号（账号必须唯一）

手机号（合法的11位数字）

身份证号码（合法18位）

密码默认123456

### 接口设计

请求方式为post，提交内容表单项（json格式），后端返回格式为json，封装为result

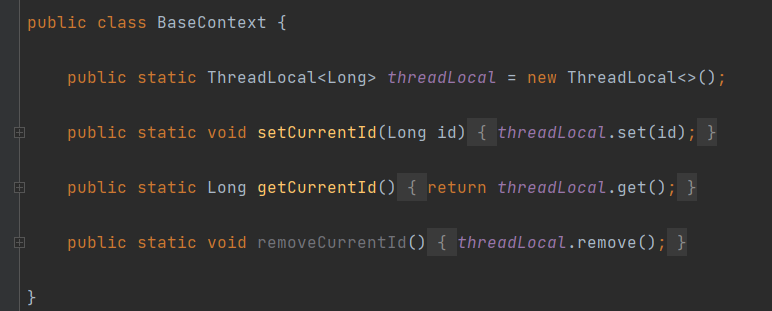


数据库表

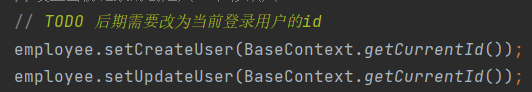


当前端提交的数据与实体类中对应的属性差别比较大的时候，建议用DTO来封装参数

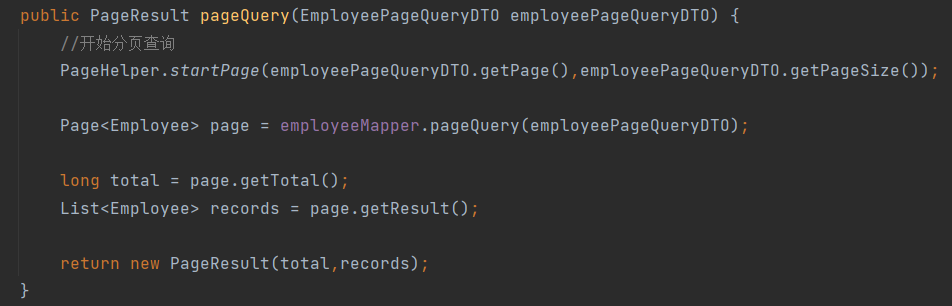
用线程来获得当前用户的登录id

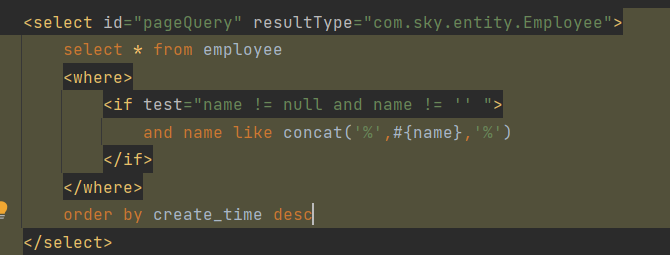






用pageHelper插件来进行分页



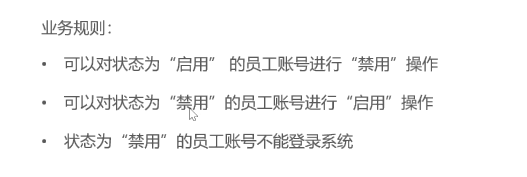


根据创造时间来排序

时间格式有问题

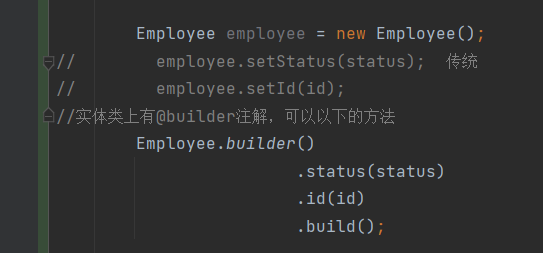
解决：建议第二种





接口设为post，提交状态和id

状态：1为启用，0为禁用

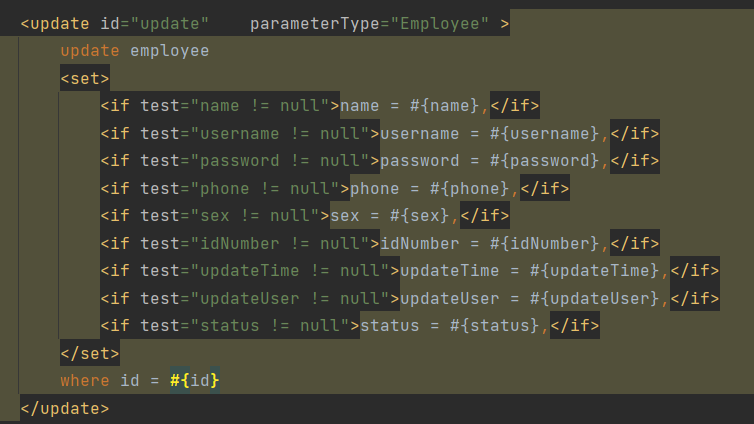


编辑员工

根据员工id查询员工信息 用get

辑员工信息 用put

因为用的是动态修改员工表，编辑员工和启用禁用员工用同一个mapper



## 菜品

菜品的分类管理

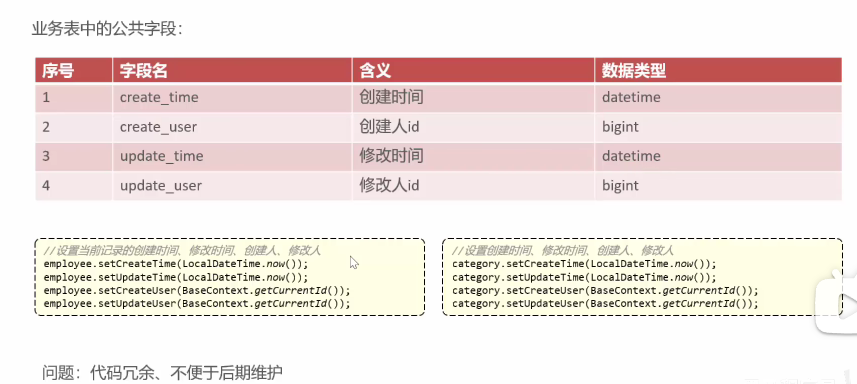
分类名称必须是唯一的

分类按照类型可以分为菜品分类和套餐分类

新添加的分类状态默认为“禁用”

业务表中员工表和分类表等许多表，有4个一样的字段

在程序中存在很多冗余的代码



采用aop来解决





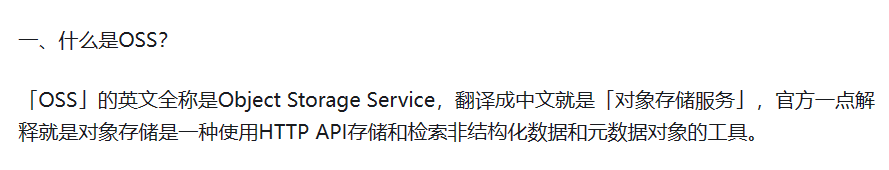
新增菜品

1、根据类型查询分类

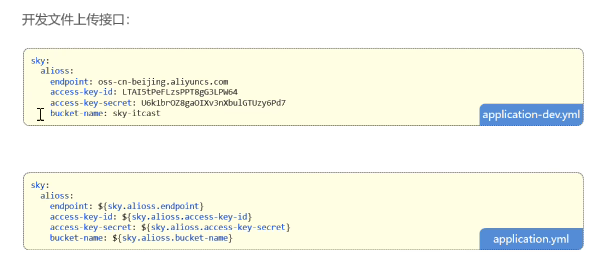
2、文件上传

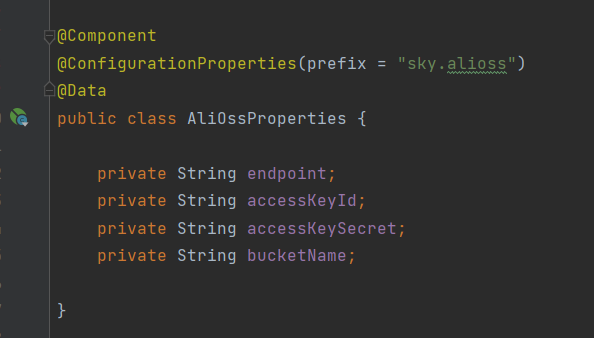
3、新增菜品

阿里云oss（用来存储上传的图片）



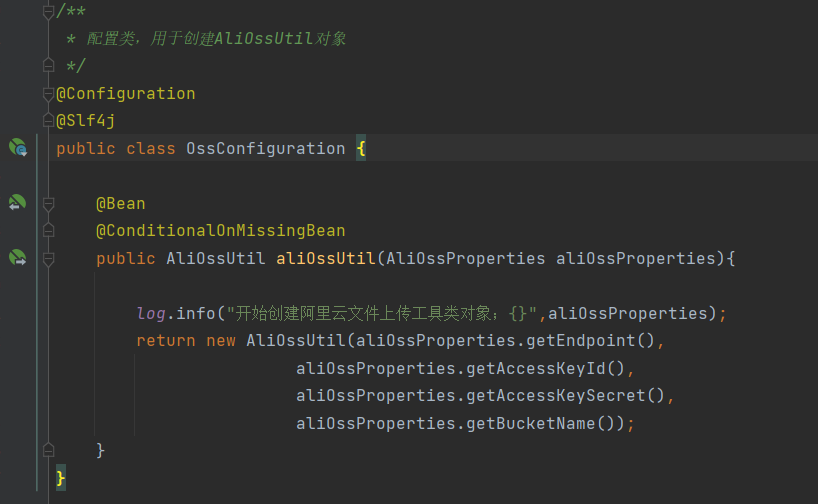
使用oss



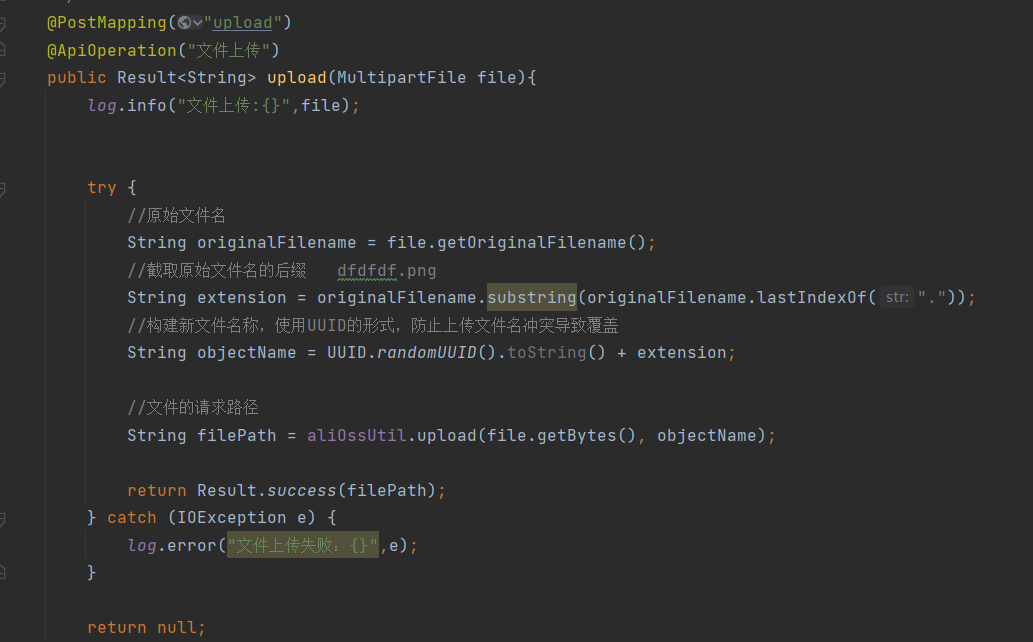


得到一个alioss的实体类

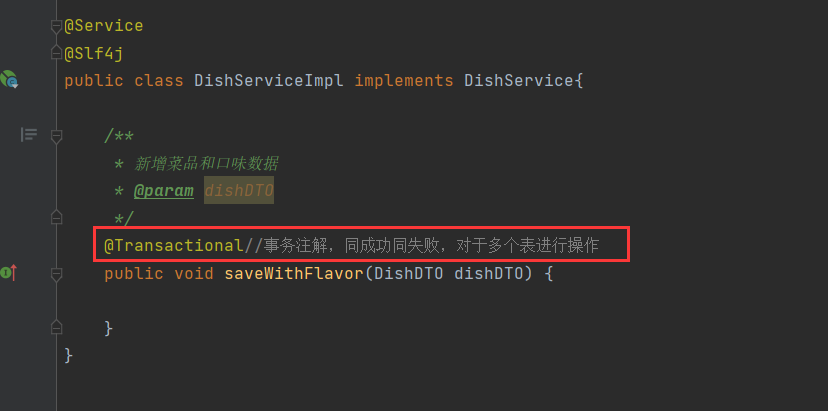
使用config来配置aliossutil对象



为了防止上传文件名字冲突，使用uuid

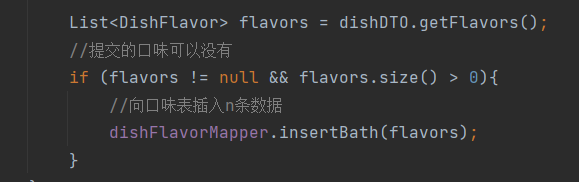


事务

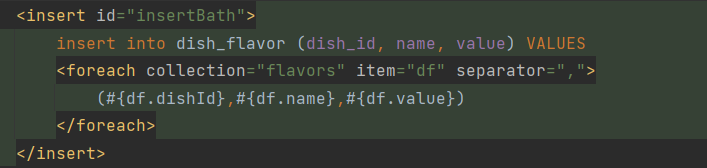


前提条件



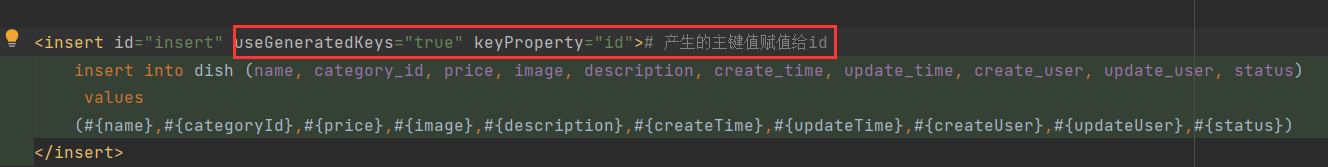
传过来的flavor是list，需要遍历

动态sql，通过《foreach》来遍历



从数据库取得自动生成的id（主键值），前端不需要传这个值

需要



菜品分页查询

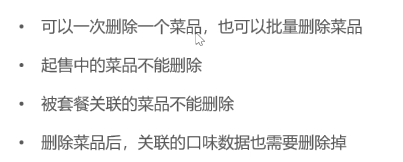
通过地址栏传参数，需要使用对应的DTO，这里还需要用VO，则不用@requestbody

如果传过来的是json格式，则不需要用DTO，但需要加参数@requestbody



和VO里面的字段相对应

删除菜品

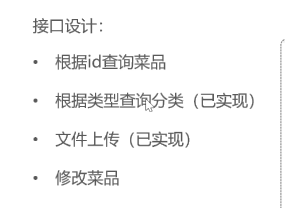


@RequestParam

这个注解可以让前端传过来的字符串，按逗号隔开进行遍历

**DTO是接收数据，VO是返回数据**

修改菜品

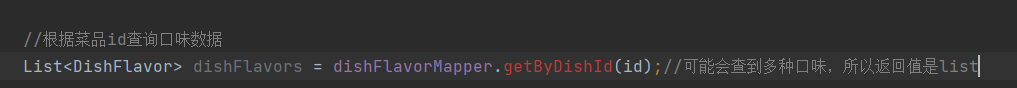


查询菜品时还需要查口味

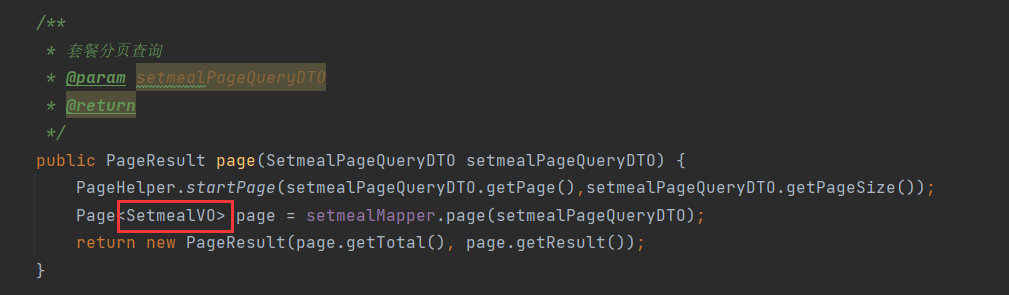
从前端传过来的是路径参数的话

需要

还需要



套餐分页查询

需要用

VO来返回

店铺营业状态

接口设计

设置营业状态

管理端查询营业状态 /admin前缀

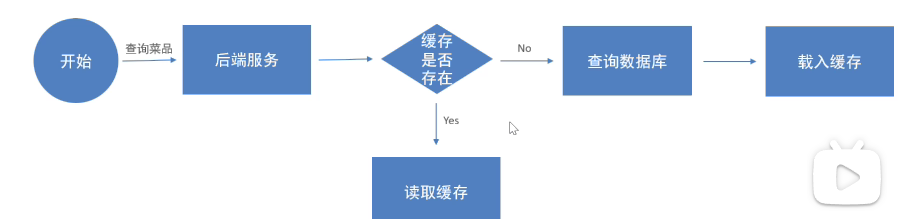
用户端查询营业状态 /user前缀

营业状态的存储方式：基于redis的字符串来进行存储（因为只有一个字段，没必要在mysql中单独搞一张表）

## 缓存菜品

问题：某时间段，访问用户过多，数据库查询会变慢

解决：redis来缓存菜品数据，减少数据库查询操作



缓存逻辑分析：

每个分类下的菜品保存一份缓存数据

数据库中菜品有变更时清理缓存数据