

JDK 1.8+

license Apache 2

Spring Cloud Greenwich.SR2

Spring Boot 2.1.6.RELEASE

HRS code challenge

作者：CJ

技术选型

- 服务注册与发现：Eureka
- 内部服务调用：Feign
- 网关：Spring Gateway
- 认证鉴权：JWT
- 程序监控：Spring Boot Admin / Spring Boot Actuator
- 数据库：MySQL 5.7
- 部署：Docker + docker-compose
- 构建工具：Maven
- 后台 API：swagger-bootstrap-ui （不推荐Swagger，代码入侵太强，推荐Yapi）
- 测试：Mockito 方便stub、mock

项目设计和流程

流程

流程：分解需求-> 需求、目标、性能、数据量、细节确认 -> 功能模块划分 -> 搭建框架和表设计 -> 每个模块功能实现和单元测试 -> 组装模块和服务聚合 -> 集成测试和功能测试 -> case测试覆盖 -> 编写文档整理Wiki -> Review代码 -> 提测 -> 测试通过 -> 修bug、测试、Review、备份、发版 -> 线上验证 -> 归档、修bug、回滚等

分析与设计

若系统是面向酒店平台，入离存取需经接单系统->MQ削峰->后续流量消费，考虑扩展和业务膨胀可提前分库分表

若系统是面向单个酒店，一个酒店一套，数据量不大，无需分库分表

查询类接口：宾客信息是否脱敏？对外接口要脱敏防爬，对内系统有登陆权限的人可以看

安全性：接口要防刷、调用要认证，内部服务之间调用安全放开，无限制

Parcel寄存、取件后若要打印流水单（唯一id生成器，选型leaf、snowflake、uuid），目前只做了uuid演示

调用链路是admin（酒店接待登陆操作，接口对外开放）-> 调聚合子服务feignclient -> feignclient调内部微服务 -> 单个微服务单模块项目可简单聚合相关业务service

feign client跨服务调用get请求传递问题，考虑get请求的参数太长会丢，内部服务最好支持post（项目采用的这种），若依旧get传递，解决方案参考

https://blog.csdn.net/qq_43648299/article/details/103223138，项目里已实现，但不推荐，只能适用get参数少的情况。

内部模块服务抛出的自定义异常向feign再向外逐层传递，需自己转化传递，项目已实现

查询接口公用单个、列表、模糊、指定条件都支持分页

模块划分：

授权：登陆、短信、扫码、第三方跳转登陆授权

酒店：酒店信息、入离登记、

用户：宾客信息、admin系统登陆者（即酒店接待、不排除有超管理员）

包裹：Parcel存、取信息、寄存柜（有、没有，系统现在实现了有，没有没兼容）、状态，关联来宾、admin、酒店入离时间等重要部分信息可以考虑冗余在包裹业务表，考虑数据量增大减少查询时间，其他不重要的可以从每个子模块服务的feignClient聚合向下传递、不重要的依赖模块查询可以异步聚合返回关于入离、酒店是否允许

关联酒店信息查询：酒店名称、地址、地域、酒店联系电话、酒店地址

Parcel寄存：宾客电话号码、宾客姓名、寄存单号、Parcel名称、Parcel数量、Parcel种类、Parcel明细、备注（特征）、操作员、寄存操作时间

Parcel查询

离店:Parcel查询、Hotel Room状态

Parcel领取+宾客签字+寄存单号核对+Parcel核对

寄存收费、押金退还合计

结账

出结账单

更新房态

Parcel遗留物品查询：

宾客信息查询、会员信息查询聚合 证件号码、证件类型、宾客姓名、电话号码、宾客地址、宾客会员类型、宾客会员卡号、剩余积分

宾客领回更新是否领回状态：

领回人姓名、领回人手机号码、领回时间、领回经办操作员、领回经办操作时间

添加遗留物品：

遗留物品发现时间、物品数量、物品明细、是否领回

添加Parcel遗失、赔付

耗时

需求分解、设计、环境本地、框架、项目结构搭建 2个晚上

开发、测试、写文档 1个晚上+半天+一天周末

项目介绍

code chanllenge 适用酒店入离包裹存储的分布式系统。系统前后端分离，后台采用Spring Cloud 为核心框架。系统目前已完成主要功能有包裹存、取、包裹记录查看（指定条件、状态、姓名模糊、手机号模糊查询等分页）、寄存柜查看（指定条件、状态查询分页）。该系统设计具有如下：

- 前后端分离，客户端和服务端纯Token交互；
- 客户端请求资源只能通过微服务网关获取；
- 集成 SpringBootAdmin ，多维度监控微服务， Skywaking 、 Prometheus 未接入；
- 集成 Swagger 接口文档为 swagger-bootstrap-ui ；
- 代码结构、模块、对内、对外的服务分明、方便后续扩展；
- 接入 Apollo 或 Nacos 配置中心，增强业务兼容无寄存柜存取包裹、丢失登记、清理寄存柜、赔偿、收费、极端情况等拓展（不断完善中）；
- 集成 Mockito ，支持丰富Mock测试覆盖CASE

项目组织结构

code-chanllenge

- └─ hrs-admin 前后台对接的管理系统后台服务，由于user服务和auth服务没开发呢，jwt先配置permitUrl允许通过从网关跳转过来的请求，实际上是要验证登陆授权后将角色注入SimpleGrantedAuthority，对外开放的接口
@PreAuthorize("hasAnyRole('ROLE')")验证角色
- └─ hrs-auth 登陆、授权、验证 没做
- └─ hrs-common 工具类、自定义异常、API返回、分页、实体等各种可以抽取公共的（服务层的数据库Dao、Service也可以抽取成公共，项目没做）
- └─ hrs-eureka-server 注册中心
- └─ hrs-gateway 网关、路由、限流（目前没做）
- └─ hrs-hotel 入离登记、酒店信息维护 没开发
 - └─ hrs-hotel-feign-client
 - └─ hrs-hotel-service
- └─ hrs-jwt-security 安全和token，启动类加了自定义注解
@EnableJwtSecurity, permitUrl不放开，是不能直接调Controller
- └─ hrs-parcel 包裹、寄存柜、存、取、查
 - └─ hrs-parcel-feign-client
 - └─ hrs-parcel-service
- └─ hrs-user 来宾、酒店接待、系统管理员维护，没开发
 - └─ hrs-user-feign-client
 - └─ hrs-user-service
- └─ zip
 - └─ jar 项目之间依赖的jar和可执行的服务jar
 - └─ pdf 文档说明
 - └─ sql-script 脚本SQL PMO号-任务号-优先级-日期-库名-

XX-feign-client是服务之间调用，通过feign-client来调内部的具体子服务

运行方式

- Mian 启动
- Jar 参考 https://blog.csdn.net/qg_35860138/article/details/82701919

```
nohup java -jar XXX.jar > log.txt &
```

运行步骤

- 请确认环境： MySQL5.7+ 、 JDK1.8;
- 创建hrs数据库，数据库脚本在项 zip/sql-script文件夹下， 直接导入即可;
- 安装Idea和Git并导入源码，Idea需安装 lombok 插件。源码仓库： git clone <https://github.com/hongchengjian/code-challenge.git>
- 导入源码之后，拉取Maven依赖进行build; 打包子服务命令是： `mvn install -Dmaven.test.skip=true`
`mvn clean package -Ptest -Dmaven.test.skip=true`
- 项目启动先后顺序
 - 启动 hrs-eureka-server 直接运行 com.hrs.eureka.EurekaApplication 的 main ;
 - 启动 hrs-admin 直接运行 com.hrs.admin.AdminServiceApplication 的 main ;
 - 启动 hrs-parcel-service 直接运行 com.hrs.parcel.ParcelServiceApplication 的 main ;
 - 启动 hrs-gateway 直接运行 com.hrs.gateway.GatewayApplication 的 main ;
- Eureka注册状态： `http://{ip}:9000/`
- Swagger文档： `http://{ip}:{port}/swagger-ui.html#` 只写了1个接口， 没继续写各种注解侵入性的代码。Swagger使用参考<https://www.jianshu.com/p/349e130e40d5>

配置整理

后续整到Apollo， 只举例， 不一一复制粘贴， 公用的配置可提取到一起

hrs-parcel-service

```
spring.application.name=hrs-parcel-service
server.port=8087
```

```
eureka.client.register-with-eureka=true
eureka.client.fetchRegistry=true
eureka.client.service-url.defaultZone=http://localhost:9000/eureka
```

```
spring.datasource.username=${DB_USERNAME:root}
spring.datasource.password=${DB_PASSWORD:root}
spring.datasource.url=jdbc:mysql://${DB_HOST:127.0.0.1}:3306/${DATABASE_NAME:hrs}?useSSL=false&characterEncoding=utf-8&serverTimezone=GMT%2B8&useTimezone=true&allowMultiQueries=true
```

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jackson.generator.WRITE_BIGDECIMAL_AS_PLAIN=true
spring.jackson.serialization.write-dates-as-timestamps=true

logging.path=${LOG_PATH:/opt/hrs/${spring.application.name}/}

mybatis-plus.configuration.log-
impl=org.apache.ibatis.logging.stdout.StdoutImpl

jwt.header=${JWT_HEADER:Authorization}
jwt.secret=${JWT_SECRET:test}
jwt.expiration=${JWT_EXPIRATION:60}
jwt.tokenHead=${JWT_TOKEN_HEADER:Bearer }
# 启动类加了自定义注解@EnableJwtSecurity, permitUrl不放开, 是不能直接调Controller
#security.permitUrl=${SECURITY_PERMITURL:/actuator/**,/internal/**}

security.permitUrl=${SECURITY_PERMITURL:/actuator/**,/internal/**}
```

省略...