预约挂号及下单

## 一、预约挂号

### 1、需求分析

**1、接口分析**

（1）根据预约周期，展示可预约日期数据，按分页展示

（2）选择日期展示当天可预约列表（该接口后台已经实现过）

**2、页面展示分析**

（1）分页展示可预约日期，根据有号、无号、约满等状态展示不同颜色，以示区分

（2）可预约最后一个日期为即将放号日期，根据放号时间页面展示倒计时

### 2、在HospitalApiController类添加方法

### 3、添加service接口实现

### 4、前端实现

## 二、预约确认

1、在HospitalApiController类添加方法

2、添加service接口实现

3、前端实现

## 三、预约下单

### 1、分析

下单参数：就诊人id与排班id

1. 下单我们要获取就诊人信息
2. 获取排班下单信息与规则信息
3. 获取医院签名信息，然后通过接口去医院预约下单
4. 下单成功更新排班信息与~~发送短信~~

### 2、搭建service-order模块

#### 1、添加配置文件application.properties

1. # 服务端口
2. server.port=8206
3. # 服务名
4. spring.application.name=service-order
5. # 环境设置：dev、test、prod
6. spring.profiles.active=dev
8. # mysql数据库连接
9. spring.datasource.driver-class-name=com.mysql.jdbc.Driver
10. spring.datasource.url=jdbc:mysql://localhost:3306/yygh\_order?characterEncoding=utf-8&useSSL=false
11. spring.datasource.username=root
12. spring.datasource.password=root
14. #返回json的全局时间格式
15. spring.jackson.date-format=yyyy-MM-dd HH:mm:ss
16. spring.jackson.time-zone=GMT+8
18. spring.data.mongodb.uri=mongodb://192.\*\*\*:27017/yygh\_hosp
20. # nacos服务地址
21. spring.cloud.nacos.discovery.server-addr=localhost:8848
23. #rabbitmq地址
24. spring.rabbitmq.host=192.168.2\*\*
25. spring.rabbitmq.port=5672
26. spring.rabbitmq.username=guest
27. spring.rabbitmq.password=guest

#### 2、启动类

1. **package** com.xlh.yygh.order;
3. **import** org.springframework.boot.SpringApplication;
4. **import** org.springframework.boot.autoconfigure.SpringBootApplication;
5. **import** org.springframework.cloud.client.discovery.EnableDiscoveryClient;
6. **import** org.springframework.cloud.openfeign.EnableFeignClients;
7. **import** org.springframework.context.annotation.ComponentScan;
9. @SpringBootApplication
10. @ComponentScan(basePackages = {"com.xlh"})
11. @EnableDiscoveryClient
12. @EnableFeignClients(basePackages = {"com.xlh"})
13. **public** **class** ServiceOrderApplication {
14. **public** **static** **void** main(String[] args) {
15. SpringApplication.run(ServiceOrderApplication.**class**, args);
16. }
17. }

#### 3、配置网关

1. #设置路由id
2. spring.cloud.gateway.routes[4].id=service-order
3. #设置路由的uri
4. spring.cloud.gateway.routes[4].uri=lb://service-order
5. #设置路由断言,代理servicerId为auth-service的/auth/路径
6. spring.cloud.gateway.routes[4].predicates= Path=/\*/order/\*\*

#### 4、添加Mapper、service接口及实现类、controller

### 3、封装Feign调用获取就诊人接口

**1、获取就诊人信息api接口：**

**操作模块：**service-user

在PatientApiController类添加方法

1. @ApiOperation(value = "获取就诊人")
2. @GetMapping("inner/get/{id}")
3. **public** Patient getPatientOrder(
4. @ApiParam(name = "id", value = "就诊人id", required = **true**)
5. @PathVariable("id") Long id) {
6. **return** patientService.getById(id);
7. }

**2、搭建service-user-client模块**

1. **package** com.xlh.yygh.user.client;
3. **import** com.xlh.yygh.model.user.Patient;
4. **import** org.springframework.cloud.openfeign.FeignClient;
5. **import** org.springframework.stereotype.Repository;
6. **import** org.springframework.web.bind.annotation.GetMapping;
7. **import** org.springframework.web.bind.annotation.PathVariable;
9. @FeignClient(value = "service-user")
10. @Repository
11. **public** **interface** PatientFeignClient {
12. //获取就诊人
13. @GetMapping("/api/user/patient/inner/get/{id}")
14. Patient getPatient(@PathVariable("id") Long id);
15. }

### 4、封装Feign调用获取排班下单信息接口

操作模块：service-hosp

1、添加controller方法

2、添加service接口与实现

3、获取下单引用签名信息接口

操作模块：service-hosp

1）在HospitalSetService类添加接口getSignInfoVo()

2) 在HospitalSetServiceImpl类添加实现

3) 在HospitalApiController类添加方法

4、搭建service-hosp-client模块

添加Feign接口类：

1. **package** com.xlh.yygh.hosp.client;
3. **import** com.xlh.yygh.vo.hosp.ScheduleOrderVo;
4. **import** com.xlh.yygh.vo.order.SignInfoVo;
5. **import** org.springframework.cloud.openfeign.FeignClient;
6. **import** org.springframework.stereotype.Repository;
7. **import** org.springframework.web.bind.annotation.GetMapping;
8. **import** org.springframework.web.bind.annotation.PathVariable;
10. @FeignClient(value = "service-hosp")
11. @Repository
12. **public** **interface** HospitalFeignClient {
13. /\*\*
14. \* 根据排班id获取预约下单数据
15. \*/
16. @GetMapping("/api/hosp/hospital/inner/getScheduleOrderVo/{scheduleId}")
17. ScheduleOrderVo getScheduleOrderVo(@PathVariable("scheduleId") String scheduleId);
18. /\*\*
19. \* 获取医院签名信息
20. \*/
21. @GetMapping("/api/hosp/hospital/inner/getSignInfoVo/{hoscode}")
22. SignInfoVo getSignInfoVo(@PathVariable("hoscode") String hoscode);
23. }

### 5、实现下单接口

操作模块：service-order

1、实现下单接口

2、预约成功后处理逻辑

预约成功后我们要更新预约数，为了提高下单的并发性，这部分逻辑我们就交给mq为我们完成。详情见代码。