

一、取消预约

1、需求描述

取消订单分两种情况：

- (1) 未支付取消订单，直接通知医院更新取消预约状态
- (2) 已支付取消订单，先退款给用户，然后通知医院更新取消预约状态

2、开发取消预约接口

参考文档：https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9_4

该接口需要使用证书，详情参考文档并下载证书

(1) 配置证书

请下载证书放在service-order模块/resources/cert文件夹下

在application.properties文件配置证书路径

```
1 weixin.cert=  
2 C:\\yygh_parent\\service\\service_order\\src\\main\\resources\\cert\\apiclient_cert.p
```

(2) 添加获取支付记录方法

操作PaymentService接口和实现类

```
1 /**  
2     * 获取支付记录  
3     * @param orderId  
4     * @param paymentType  
5     * @return  
6     */  
7 PaymentInfo getPaymentInfo(Long orderId, Integer paymentType);  
8  
9 //实现获取支付记录方法  
10 @Override
```

```

11 public PaymentInfo getPaymentInfo(Long orderId, Integer paymentType) {
12     QueryWrapper<PaymentInfo> queryWrapper = new QueryWrapper<>();
13     queryWrapper.eq("order_id", orderId);
14     queryWrapper.eq("payment_type", paymentType);
15     return baseMapper.selectOne(queryWrapper);
16 }

```

(3) 添加退款记录

添加RefundInfoMapper

```

1 public interface RefundInfoMapper extends BaseMapper<RefundInfo> {
2 }

```

添加RefundInfoService方法和实现

```

1 //RefundInfoService
2 public interface RefundInfoService extends IService<RefundInfo> {
3     /**
4      * 保存退款记录
5      * @param paymentInfo
6      */
7     RefundInfo saveRefundInfo(PaymentInfo paymentInfo);
8 }
9
10 //RefundInfoServiceImpl
11 @Service
12 public class RefundInfoServiceImpl extends ServiceImpl<RefundInfoMapper, RefundInfo>
13
14     @Override
15     public RefundInfo saveRefundInfo(PaymentInfo paymentInfo) {
16         QueryWrapper<RefundInfo> queryWrapper = new QueryWrapper<>();
17         queryWrapper.eq("order_id", paymentInfo.getOrderId());
18         queryWrapper.eq("payment_type", paymentInfo.getPaymentType());
19         RefundInfo refundInfo = baseMapper.selectOne(queryWrapper);
20         if(null != refundInfo) return refundInfo;
21         // 保存交易记录
22         refundInfo = new RefundInfo();

```

```

23     refundInfo.setCreateTime(new Date());
24     refundInfo.setOrderId(paymentInfo.getOrderId());
25     refundInfo.setPaymentType(paymentInfo.getPaymentType());
26     refundInfo.setOutTradeNo(paymentInfo.getOutTradeNo());
27     refundInfo.setRefundStatus(RefundStatusEnum.UNREFUND.getStatus());
28     refundInfo.setSubject(paymentInfo.getSubject());
29     //paymentInfo.setSubject("test");
30     refundInfo.setTotalAmount(paymentInfo.getTotalAmount());
31     baseMapper.insert(refundInfo);
32     return refundInfo;
33 }
34 }

```

(4) 添加微信退款方法

在WeixinService添加方法

```

1  /**
2   * 退款
3   * @param orderId
4   * @return
5   */
6  Boolean refund(Long orderId);

```

在WeixinServiceImpl添加实现

```

1  @Override
2  public Boolean refund(Long orderId) {
3      try {
4          PaymentInfo paymentInfoQuery = paymentService.getPaymentInfo(orderId, Payment
5
6          RefundInfo refundInfo = refundInfoService.saveRefundInfo(paymentInfoQuery);
7          if(refundInfo.getRefundStatus().intValue() == RefundStatusEnum.REFUND.getStat
8              return true;
9      }
10     Map<String,String> paramMap = new HashMap<>(8);
11     paramMap.put("appid",ConstantPropertiesUtils.APPID);           //公众账号ID
12     paramMap.put("mch_id",ConstantPropertiesUtils.PARTNER);       //商户编号
13     paramMap.put("nonce_str",WXPayUtil.generateNonceStr());

```

```

14     paramMap.put("transaction_id", paymentInfoQuery.getTradeNo()); //微信订单号
15     paramMap.put("out_trade_no", paymentInfoQuery.getOutTradeNo()); //商户订单编号
16     paramMap.put("out_refund_no", "tk"+paymentInfoQuery.getOutTradeNo()); //商户退款单号
17     //     paramMap.put("total_fee", paymentInfoQuery.getTotalAmount().multiply(100));
18     //     paramMap.put("refund_fee", paymentInfoQuery.getTotalAmount().multiply(100));
19     paramMap.put("total_fee", "1");
20     paramMap.put("refund_fee", "1");
21     String paramXml = WXPAYUtil.generateSignedXml(paramMap, ConstantPropertiesUtil.getProperties());
22     HttpClient client = new HttpClient("https://api.mch.weixin.qq.com/secapi/pay/");
23     client.setXmlParam(paramXml);
24     client.setHttps(true);
25     client.setCert(true);
26     client.setCertPassword(ConstantPropertiesUtil.PARTNER_PASSWORD);
27     client.post();
28     //3、返回第三方的数据
29     String xml = client.getContent();
30     Map<String, String> resultMap = WXPAYUtil.xmlToMap(xml);
31     if (null != resultMap && WXPAYConstants.SUCCESS.equalsIgnoreCase(resultMap.get("return_code"))) {
32         refundInfo.setCallbackTime(new Date());
33         refundInfo.setTradeNo(resultMap.get("refund_id"));
34         refundInfo.setRefundStatus(RefundStatusEnum.REFUND.getStatus());
35         refundInfo.setCallbackContent(JSONObject.toJSONString(resultMap));
36         refundInfoService.updateById(refundInfo);
37         return true;
38     }
39     return false;
40 } catch (Exception e) {
41     e.printStackTrace();
42 }
43 return false;
44 }

```

(5) 完成取消预约方法

在OrderService添加和实现

```

1 /**
2  * 取消订单
3  * @param orderId
4  */
5 Boolean cancelOrder(Long orderId);

```

```

6
7 //实现方法
8 @Override
9 public Boolean cancelOrder(Long orderId) {
10     OrderInfo orderInfo = this.getById(orderId);
11     //当前时间大约退号时间，不能取消预约
12     DateTime quitTime = new DateTime(orderInfo.getQuitTime());
13     if(quitTime.isBeforeNow()) {
14         throw new YyghException();
15     }
16
17     Map<String, Object> reqMap = new HashMap<>();
18     reqMap.put("hoscode", orderInfo.getHoscode());
19     reqMap.put("hosRecordId", orderInfo.getHosRecordId());
20     reqMap.put("timestamp", HttpRequestHelper.getTimestamp());
21     reqMap.put("sign", "");
22
23     JSONObject result = HttpRequestHelper.sendRequest(reqMap, "http://localhost:9998/
24
25     if(result.getInteger("code") != 200) {
26         throw new YyghException(ResultCodeEnum.FAIL.getCode(), result.getString("messa
27     } else {
28         //是否支付 退款
29         if(orderInfo.getOrderStatus().intValue() == OrderStatusEnum.PAID.getStatus()).
30             //已支付 退款
31             boolean isRefund = weixinService.refund(orderId);
32             if(!isRefund) {
33                 throw new YyghException();
34             }
35         }
36         //更改订单状态
37         orderInfo.setOrderStatus(OrderStatusEnum.CANCLE.getStatus());
38         this.updateById(orderInfo);
39         //发送mq信息更新预约数 我们与下单成功更新预约数使用相同的mq信息，不设置可预约数与剩
40         OrderMqVo orderMqVo = new OrderMqVo();
41         orderMqVo.setScheduleId(orderInfo.getScheduleId());
42         //短信提示
43         MsmVo msmVo = new MsmVo();
44         msmVo.setPhone(orderInfo.getPatientPhone());
45         orderMqVo.setMsmVo(msmVo);
46         rabbitService.sendMessage(MqConst.EXCHANGE_DIRECT_ORDER, MqConst.ROUTING_ORDE
47     }
48     return true;

```

```
49 }
```

(6) 在WeixinService添加方法

在OrderApiController添加方法

```
1 @ApiOperation(value = "取消预约")
2 @GetMapping("auth/cancelOrder/{orderId}")
3 public R cancelOrder(
4     @ApiParam(name = "orderId", value = "订单id", required = true)
5     @PathVariable("orderId") Long orderId) {
6
7     Boolean flag = orderService.cancelOrder(orderId);
8     return R.ok().data("flag", flag);
9 }
```

(7) 修改监听器方法

修改HospitalReceiver 类

```
1 @Component
2 public class HospitalReceiver {
3
4     @Autowired
5     private ScheduleService scheduleService;
6
7     @Autowired
8     private RabbitService rabbitService;
9
10    @RabbitListener(bindings = @QueueBinding(
11        value = @Queue(value = MqConst.QUEUE_ORDER, durable = "true"),
12        exchange = @Exchange(value = MqConst.EXCHANGE_DIRECT_ORDER),
13        key = {MqConst.ROUTING_ORDER}
14    ))
15    public void receiver(OrderMqVo orderMqVo, Message message, Channel channel) {
16        if(null != orderMqVo.getAvailableNumber()) {
17            //更新排班
18            Schedule schedule = scheduleService.getById(orderMqVo.getScheduleId());
19            schedule.setReservedNumber(orderMqVo.getReservedNumber());
20        }
21    }
22 }
```

```

20         schedule.setAvailableNumber(orderMqVo.getAvailableNumber());
21         scheduleService.update(schedule);
22     } else {
23         //取消预约更新预约数
24         Schedule schedule = scheduleService.getById(orderMqVo.getScheduleId());
25         int availableNumber = schedule.getAvailableNumber().intValue() + 1;
26         schedule.setAvailableNumber(availableNumber);
27         scheduleService.update(schedule);
28     }
29     //发送短信
30     MsmVo msmVo = orderMqVo.getMsmVo();
31     if(msmVo != null) {
32         rabbitService.sendMessage(MqConst.EXCHANGE_DIRECT_MSM, MqConst.ROUTING_MS
33     }
34 }
35 }

```

3、开发微信退款前端

(1) 添加wx.js添加方法

```

1 cancelOrder(orderId) {
2     return request({
3         url: `/api/order/orderInfo/auth/cancelOrder/${orderId}`,
4         method: 'get'
5     })
6 },

```

(2) 修改show.vue组件

```

1 //取消预约方法
2 cancelOrder() {
3     this.$confirm('确定取消预约吗?', '提示', {
4         confirmButtonText: '确定',
5         cancelButtonText: '取消',
6         type: 'warning'

```

```
7   }).then(() => { // promise
8       // 点击确定，远程调用
9       return weixinApi.cancelOrder(this.orderId)
10  }).then((response) => {
11      this.$message.success('取消成功')
12      this.init()
13  })
14 }
```