一、订单支付后处理

1、前端页面修改

添加定时器方法，每隔3秒去查询一次支付状态

（1）封装api方法

```
queryPayStatus(orderId) {
    return request({
        url: `/api/order/weixin/queryPayStatus/${orderId}`,
        method: 'get'
    })
},
```

（2）修改页面调用

```
<script>
import '~/assets/css/hospital_personal.css'
import '~/assets/css/hospital.css'
import orderInfoApi from '@/api/orderInfo'
import weixinApi from '@/api/wx'
export default {
  data() {
    return {
      orderId: null,
      orderInfo: {
        param: {}
      },
      dialogPayVisible: false,
      payObj: {},
      timer: null  // 定时器名称
    }
  },
  created() {
    this.orderId = this.$route.query.orderId
    this.init()
  },
  methods: {

    ........

    pay() {
      this.dialogPayVisible = true
      weixinApi.createNative(this.orderId).then(response => {
        this.payObj = response.data
        if(this.payObj.codeUrl == '') {
          this.dialogPayVisible = false
          this.$message.error("支付错误")
        } else {
          this.timer = setInterval(() => {
            this.queryPayStatus(this.orderId)
          }, 3000);
        }
      })
    },
    queryPayStatus(orderId) {
      weixinApi.queryPayStatus(orderId).then(response => {
        if (response.message == '支付中') {
          return
        }
        clearInterval(this.timer);
        window.location.reload()
      })
    },
    closeDialog() {
      if(this.timer) {
        clearInterval(this.timer);
      }
    }
  }
}
</script>
```

2、创建查询支付状态接口

(1）添加WeixinService方法和实现

```
/**
 * 根据订单号去微信第三方查询支付状态
 */
Map queryPayStatus(Long orderId, String paymentType);

//实现方法
@Override
public Map queryPayStatus(Long orderId, String paymentType) {
    try {
        OrderInfo orderInfo = orderService.getById(orderId);
        //1、封装参数
        Map paramMap = new HashMap<>();
        paramMap.put("appid", ConstantPropertiesUtils.APPID);
        paramMap.put("mch_id", ConstantPropertiesUtils.PARTNER);
        paramMap.put("out_trade_no", orderInfo.getOutTradeNo());
        paramMap.put("nonce_str", WXPayUtil.generateNonceStr());
        //2、设置请求
        HttpClient client = new HttpClient("https://api.mch.weixin.qq.com/pay
        client.setXmlParam(WXPayUtil.generateSignedXml(paramMap, ConstantProp
        client.setHttps(true);
        client.post();
        //3、返回第三方的数据，转成Map
        String xml = client.getContent();
        Map<String, String> resultMap = WXPayUtil.xmlToMap(xml);
        //4、返回
        return resultMap;
    } catch (Exception e) {
        return null;
    }
}
```

(2) 添加PaymentService方法和实现

```
//更新支付状态
//outTradeNo  交易号
//paymentType  支付类型 微信 支付宝
//paramMap 调用微信查询支付状态接口返回map集合
void paySuccess(String outTradeNo, Integer paymentType, Map<String, String> p

//实现方法
//更新订单状态和更新支付记录状态
@Override
public void paySuccess(String out_trade_no, Integer weixtype, Map<String, Str
    //1 更新订单状态
    QueryWrapper<OrderInfo> wrapperOrder = new QueryWrapper<>();
    wrapperOrder.eq("out_trade_no",out_trade_no);
    OrderInfo orderInfo = orderInfoService.getOne(wrapperOrder);
    //状态已经支付
    orderInfo.setOrderStatus(OrderStatusEnum.PAID.getStatus());
    orderInfoService.updateById(orderInfo);

    //2 更新支付记录状态
    QueryWrapper<PaymentInfo> wrapperPayment = new QueryWrapper<>();
    wrapperPayment.eq("out_trade_no",out_trade_no);
    PaymentInfo paymentInfo = baseMapper.selectOne(wrapperPayment);
    //设置状态
    paymentInfo.setPaymentStatus(PaymentStatusEnum.PAID.getStatus());
    paymentInfo.setTradeNo(resultMap.get("transaction_id"));
    paymentInfo.setCallbackTime(new Date());
    paymentInfo.setCallbackContent(resultMap.toString());
    baseMapper.updateById(paymentInfo);
}
```

（3）添加Controller方法

```
@Autowired
private PaymentService paymentService;

@ApiOperation(value = "查询支付状态")
@GetMapping("/queryPayStatus/{orderId}")
public R queryPayStatus(
        @ApiParam(name = "orderId", value = "订单id", required = true)
        @PathVariable("orderId") Long orderId) {
    //调用查询接口
    Map<String, String> resultMap = weixinPayService.queryPayStatus(orderId,
    if (resultMap == null) {//出错
        return R.error().message("支付出错");
    }
    if ("SUCCESS".equals(resultMap.get("trade_state"))) {//如果成功
        //更改订单状态，处理支付结果
        String out_trade_no = resultMap.get("out_trade_no");
        paymentService.paySuccess(out_trade_no, PaymentTypeEnum.WEIXIN.getSta
        return R.ok().message("支付成功");
    }
    return R.ok().message("支付中");
}
```